Massivizing Computer Systems = Making Computer Systems Scalable, Reliable, Performant, etc., Yet Able to Form an Efficient Ecosystem

VU VRIJE UNIVERSITEIT AMSTERDAM

TUDelft

Prof. dr. ir. Alexandru Iosup

@AIosup

Co-sponsored by:

NWO Netherlands Organisation for Scientific Research

STW connecting innovators

COMMIT/

Massivizing Computer Systems
A Proposal for Collaboration, with Topics

60’

~2’ — About the Massivizing Computer Systems Group

5’ — The Golden Age of Large-Scale Computer Systems

5’ — Yet We Are in Crisis

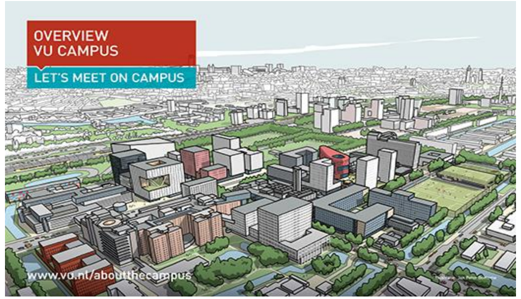
- The main challenges
- How we address them

~40’ — Our Vision and Topics

10’ — Take-Home Message

VU Amsterdam / TU Delft – Netherlands – Europe

founded 10th century
pop: 850,000



founded 1880
pop: 23,500

Amsterdam

Delft

founded 1842
pop: 19,500

founded 13th century
pop: 100,000

The Netherlands

pop: 16.5 M

EUROPE

Europe

Walldorf, Germany

http://atlarge-research.com

Our Mission

1. Improve the lives of millions through impactful research.

2. Educate the new generation of top-quality, socially responsible professionals.

3. Make innovation available to society and industry.

VU VRIJE UNIVERSITEIT AMSTERDAM

https://atlarge-research.com/about.html

TU Delft

The AtLarge Research team (Oct 2017)

Professor

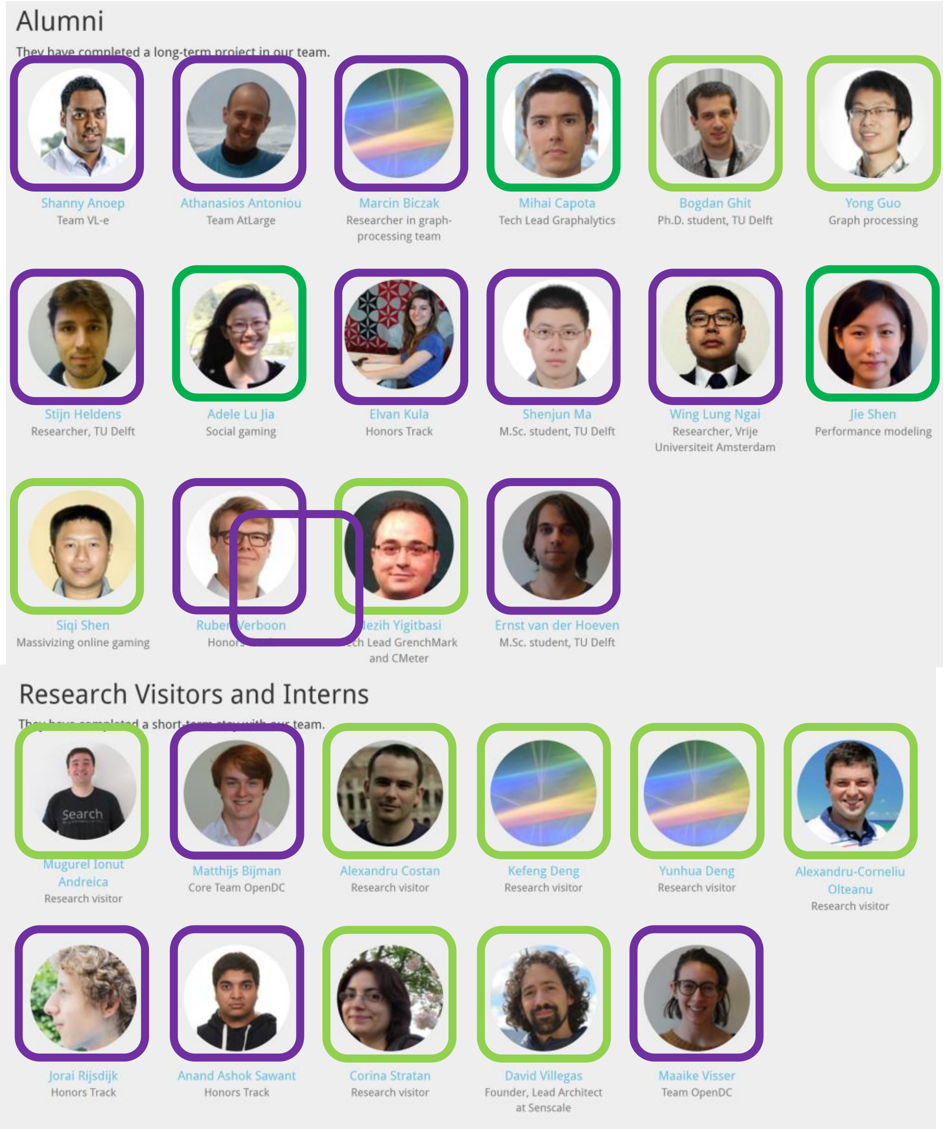
Assistant Prof.

Teacher

Post-doc

Ph.D. student

Scientist



https://atlarge-research.com/people.html

Massivizing Computer Systems
A Proposal for Collaboration, with Topics

60’

~2’ —— About the Massivizing Computer Systems Group

5’ —— The Golden Age of Large-Scale Computer Systems

5’ —— Yet We Are in Crisis

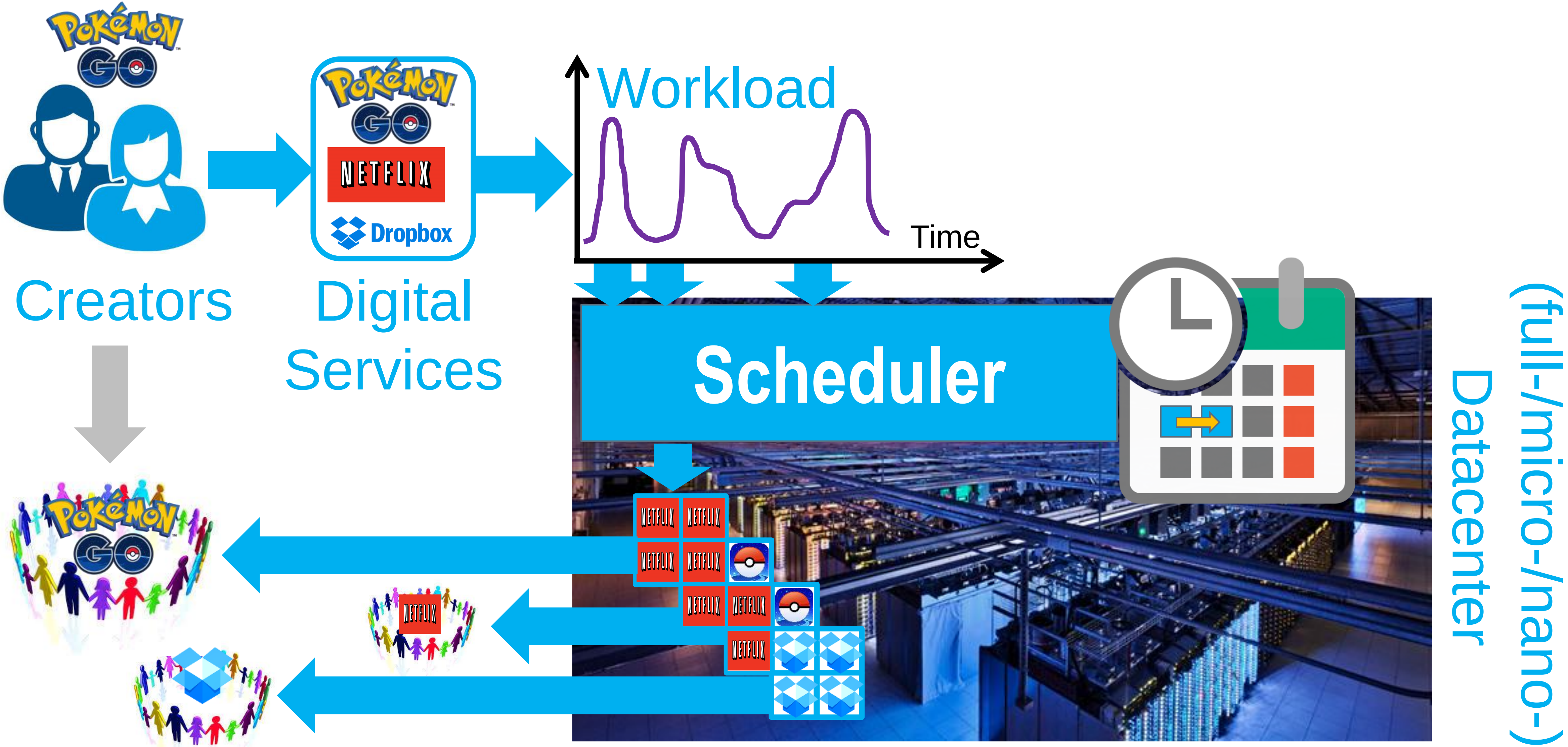
- The main challenges
- How we address them

~40’ —— Our Vision and Topics

10’ —— Take-Home Message

This Is the Golden Age of Large-Scale Systems

Current Technology: Scheduler? Datacenter? Etc.



Performance, Dependability, Efficiency

The Golden Age of Computer Systems
… Yet We Are in a Crisis

The Crisis: In the Digital Economy, Few Can Afford Being Successful!

Why does this happen?

Creator

What to do about it?

My Research

“ICT is vital for SMEs, SMEs are 60% GDP”
“15% ICT market is simple cloud services”
“Already 60+ bn.€/year”

Sources: Eurostat’15,EC Digital Agenda,IDC’14

The Scheduling Challenge

“30—70% scheduler decisions incorrect in datacenters”

Source: IEEE Computer’15

“current schedulers not efficient for many users, diverse services”

Source: Dutch industry, CCGRID’15

→ Need Smarter Schedulers

“new schedulers not used in datacenters, fear of failure”

Source: EuroPar’13,’14

→ Need to Select Schedulers

The Dependability* Challenge

* Availability, Reliability, etc.

The Register®
Biting the hand that feeds IT

Google goes dark for 2 minutes, kills 40% of world's net traffic www.theregister.co.uk/2013/08/17/google_outage/

Systemwide outage knocks every service offline

THE VERGE

TRENDING NOW
The new Nvidia Shield is the 'world's first 4K Android TV console' and launches this May for $199...

26 NEW ARTICLES

LOG IN | SIGN UP LONGFORM VIDEO REVIEWS TECH SCIENCE ENTERTAINMENT DESIGN BUSINESS US & WORLD FORUMS

APPS TECH

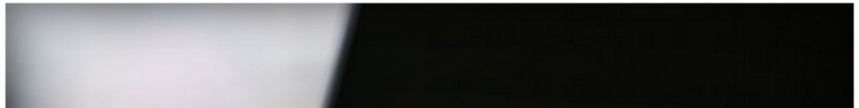
www.theverge.com/2014/2/23/5439398/whatsapp-founder-apologizes-for-our-longest-and-biggest-outage-in

82 COMMENTS

WhatsApp founder apologizes for 'our longest and biggest outage in years'

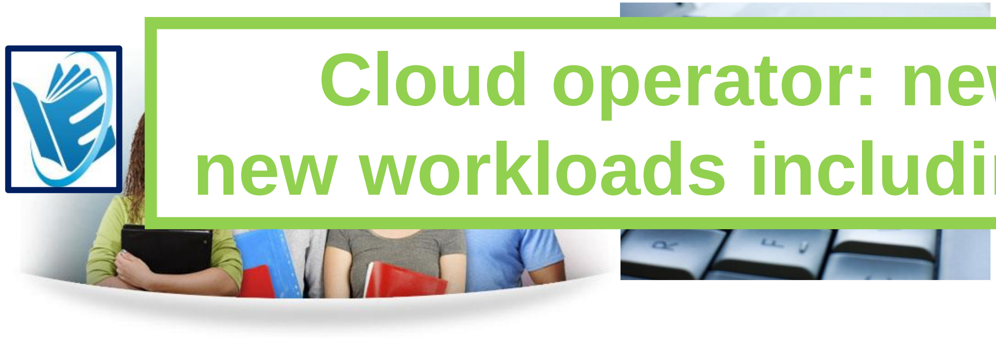
By Russell Brandom on February 23, 2014 12:25 pm Email @russellbrandom

DON'T MISS STORIES FOLLOW THE VERGE Like Follow Subscribe Follow

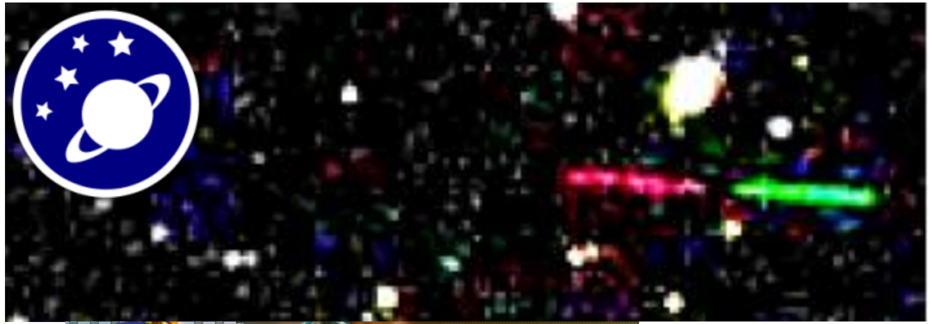


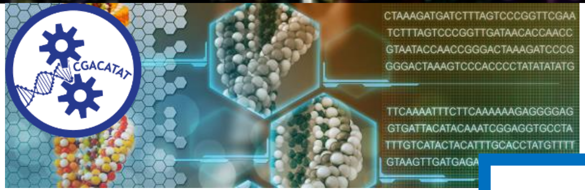
Need Dependable Systems

The New World Challenge



Cloud operator: new value-adding services, new workloads including FaaS, DevOps workloads





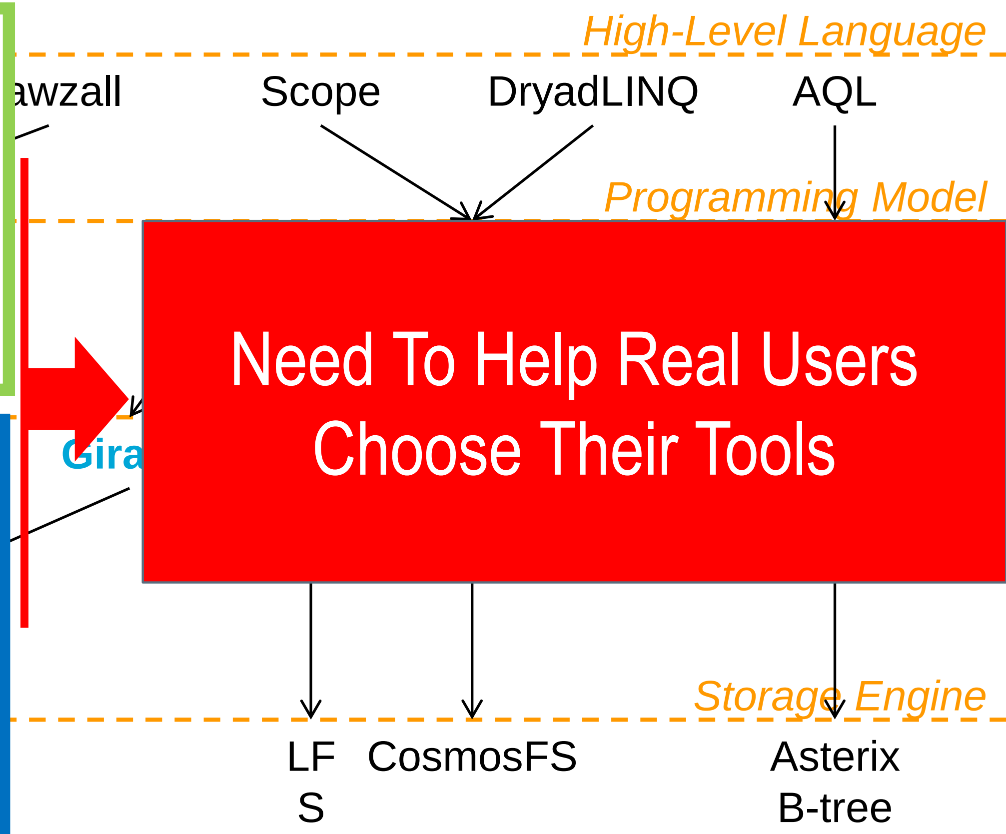
Need Operational Models

Cloud customer: new apps, new services, micro-services, customers can become operators (value-chain)

The Ecosystem Navigation Challenge

Cloud operator: how to prove capabilities? How to tune the tool? In which technology to invest? Which tech to DevOp in-house?

Cloud customer: how to choose the right tool?
For batch, workflows, stream, transactions, etc.
(No one size fits all!)



Need To Help Real Users Choose Their Tools

Batch data processing ecosystem in 2011. A later example will cover the status in 2017.

Jevons Effect: More Efficient, Yet Less Capable

Nov 2015: Over 500 YouTube videos have at least 100,000,000 viewers each.

Jun 2017: How many are there?

If you want to help kill the planet:
https://www.youtube.com/playlist?list=PLirAqAtl_h2r5g8xGajEwdXd3x1s

Need To Be Much More Efficient, But Also To Educate Our Customers

PSY Gangnam consumed ~500GWh

= more than entire countries* in a year (*41 countries),
= over 50MW of 24/7/365 diesel, 135M liters of oil,
= 100,000 cars running for a year, ...

Source: Ian Bitterlin and Jon Summers, UoL, UK, Jul 2013.
Note: Psy has >3 billion views (Nov 2015).

The New "Jevons Effect": The "Data Deluge" Challenge

Data Deluge =

Need To Address The "Data Deluge"

- Creating

To be capable of processing Big Data, need to address Volume, Velocity, Variety of Big Data*

* Other Vs possible: ours is "vicissitude"

Sources: IDC, EMC.

60' Massivizing Computer Systems
A Proposal for Collaboration, with Topics

- ~2' — About the Massivizing Computer Systems Group
- 5' — The Golden Age of Large-Scale Computer Systems
- 5' — Yet We Are in Crisis
  - The main challenges
  - How we address them
- ~40' — Our Vision and Topics
- 10' — Take-Home Message

This Is the Golden Age of Computer Systems and We Have Many Tools… Yet We Are in a Crisis

Need to Understand How to Use Our Tools

Need Smarter Schedulers

Need Dependable Systems

Need to Address "Data Deluge", "Ecosystem Navi", etc.

Need to Be Much More Efficient, But Also Ethical

… but the Current Laws and Theories Were Built For Isolated Computer Systems

Need to Understand Operational Laws when Massivizing Computer Systems

Need to Create Theories on how to Massivize Computer Systems while Ensuring Wanted Properties

Need to Build, to Massivize Computer Systems with Wanted Properties

This Is the Golden Age of Computer Systems

… Yet We Are in a Crisis

Massivizing Computer Systems Tackles All These Challenges…

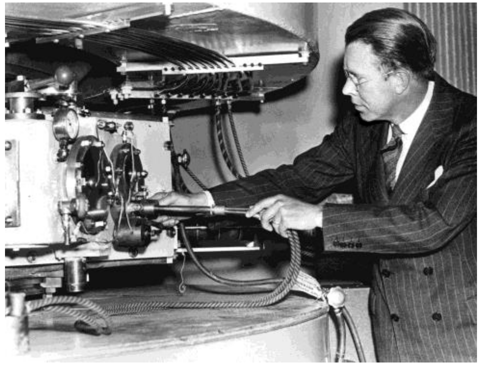
… and Is Relevant, Impactful, and Inspiring for Many Young Scientists

Massivizing Computer Systems

In Pasteur's Quadrant+:

- Fundamental research
- Inspired by real use
- Experimental in nature

~ Big Science as management, including int'l. collaborations



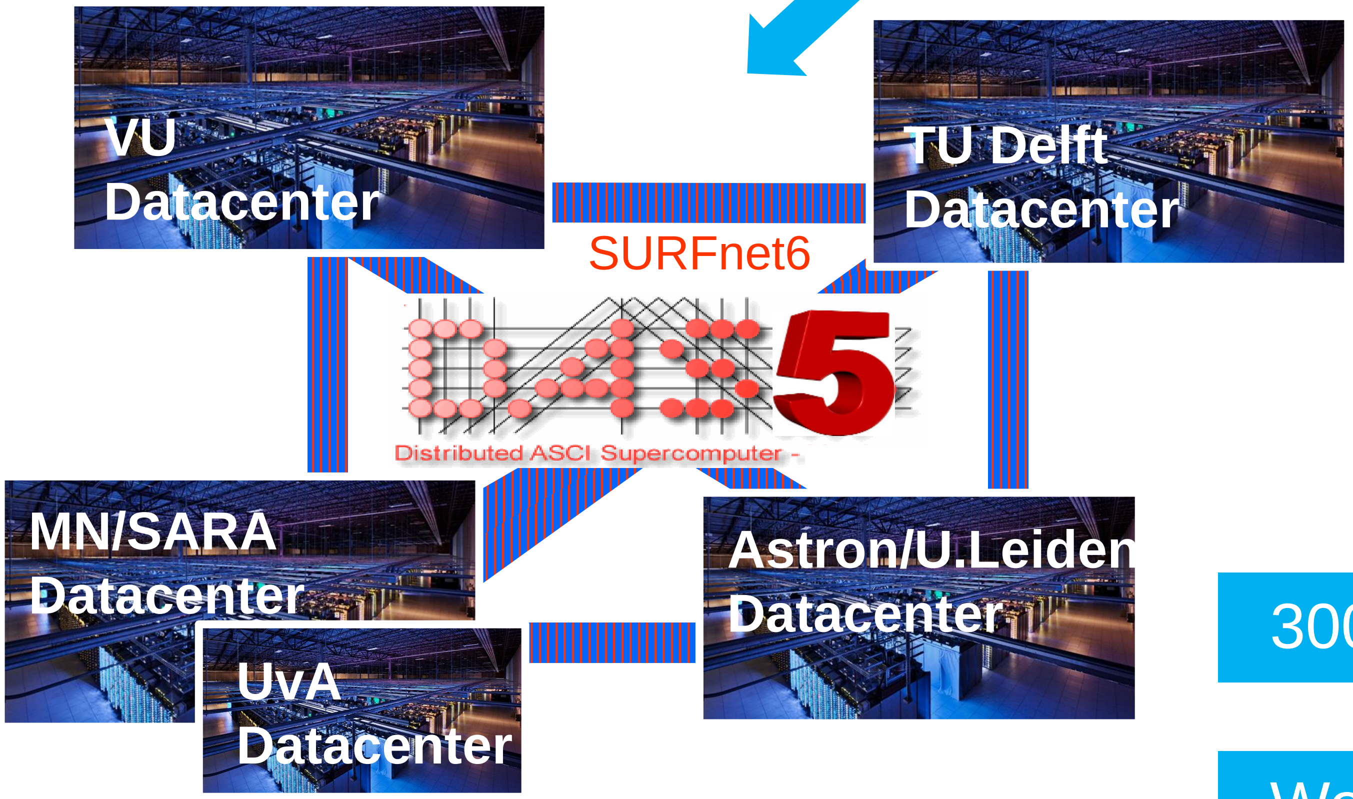
VRIJE UNIVERSITEIT AMSTERDAM

TU Delft

+ Please ask for an example

Experimental Research Methodology
Our Main Scientific Instrument: DAS-5

Our (& Your) Prototypes



300+ scientists as users

Won IEEE Scale Challenge 2014

Fundamental Research in Massivizing Comp. Sys.

Scheduling
Bags-Of-Tasks
Workflows
Portfolio

Dependability
Failure Analysis*
Space-/Time-Correlation
Availability-On-Demand

New World+
Workload Modeling
Business-Critical
Online Gaming

Ecosystem Navigator+
Performance Variability
Grid*, Cloud, Big Data
Benchmarking*
Longitudinal Studies

Scalability/Elasticity+
Delegated Matchmaking*
BTWorld*, POGGI*, AoS
Auto-Scalers
Heterogeneous Systems

Socially Aware+
Collaborative Downloads*
Groups in Online Gaming
Toxicity Detection*
Interaction Graphs

Education
Social Gamification*

Software Artifacts
Graphalytics, OpenDC

Data Artifacts
Distributed Systems Memex*

Fundamental Problems/Research Lines
My Contribution So Far
Personal grants

+ Please ask for a definition
* Award-level

60’ Massivizing Computer Systems
A Proposal for Collaboration, with Topics

~2’ — About the Massivizing Computer Systems Group

5’ — The Golden Age of Large-Scale Computer Systems

5’ — Yet We Are in Crisis

- The main challenges
- How we address them

~40’ — Our Vision and Topics

10’ — Take-Home Message

To Begin Our Discussion, Let's First Agree on Terminology

1. Let's focus on datacenter (DC) technology*, in general
2. In the following slides, you will see our view on DC technology

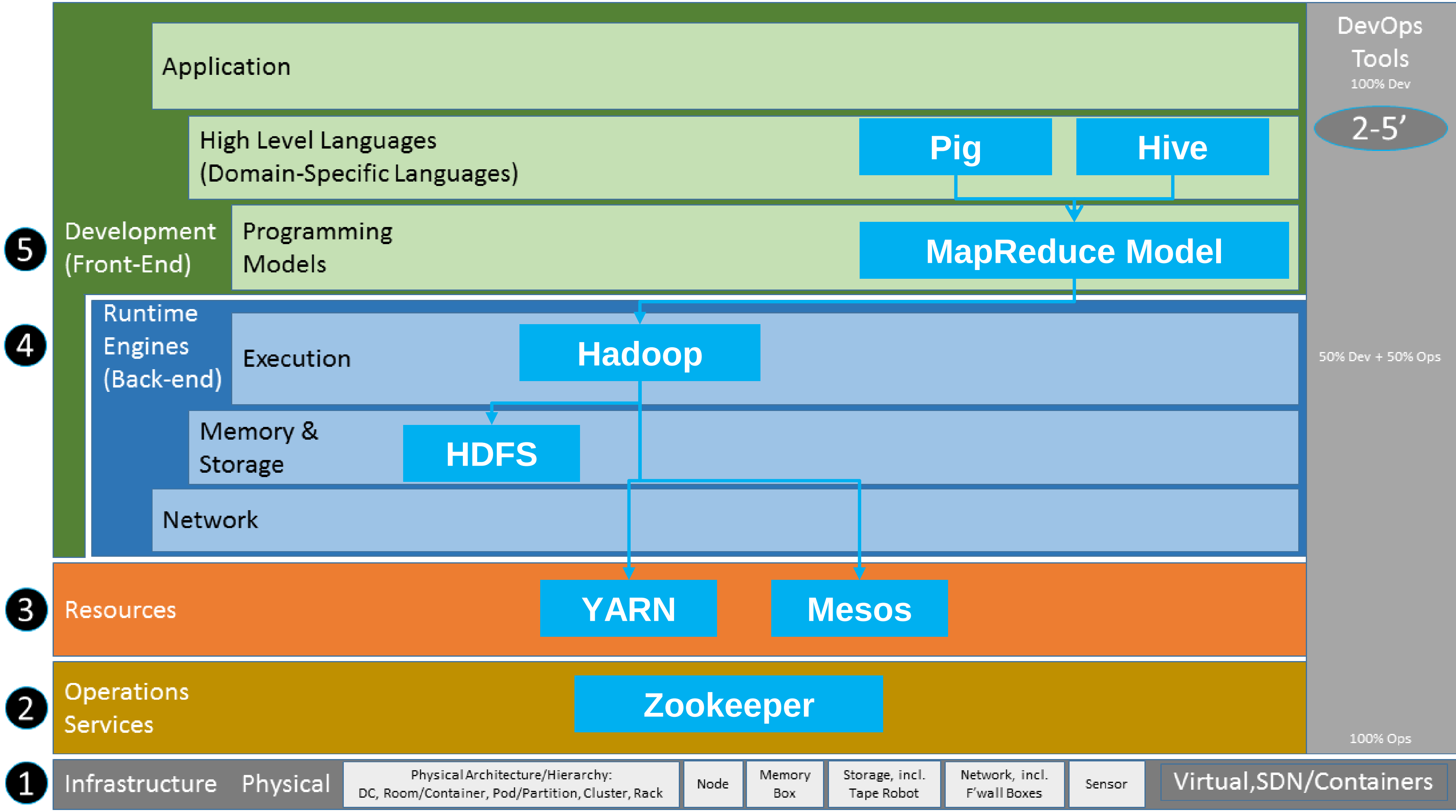
* it's everywhere

© 2017 Alexandru Iosup. All rights reserved.

A Reference Architecture for Massivizing Computer Systems

5 layers:

1. Infrastructure
2. Operations Services
3. Resources
4. Runtime Engines (Back-end)
5. Development (Front-end)

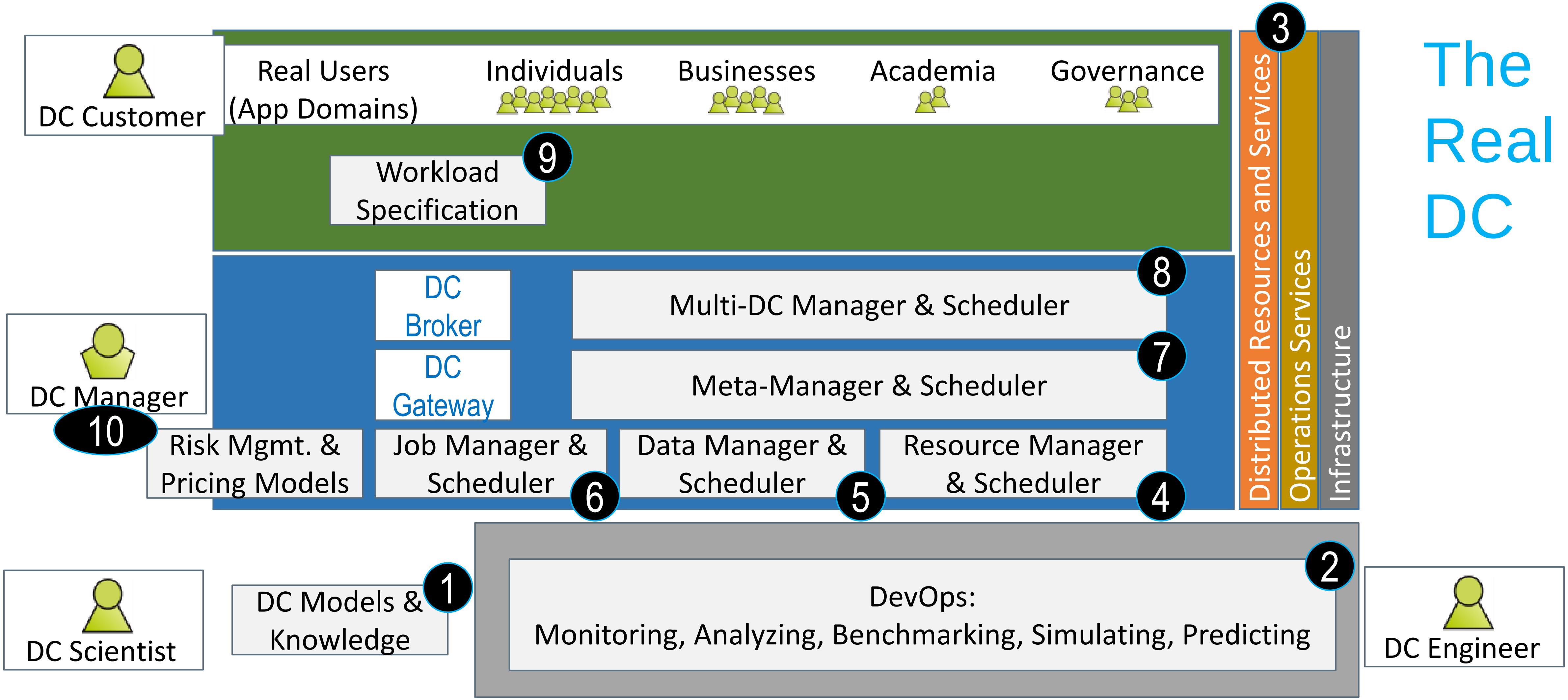


VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

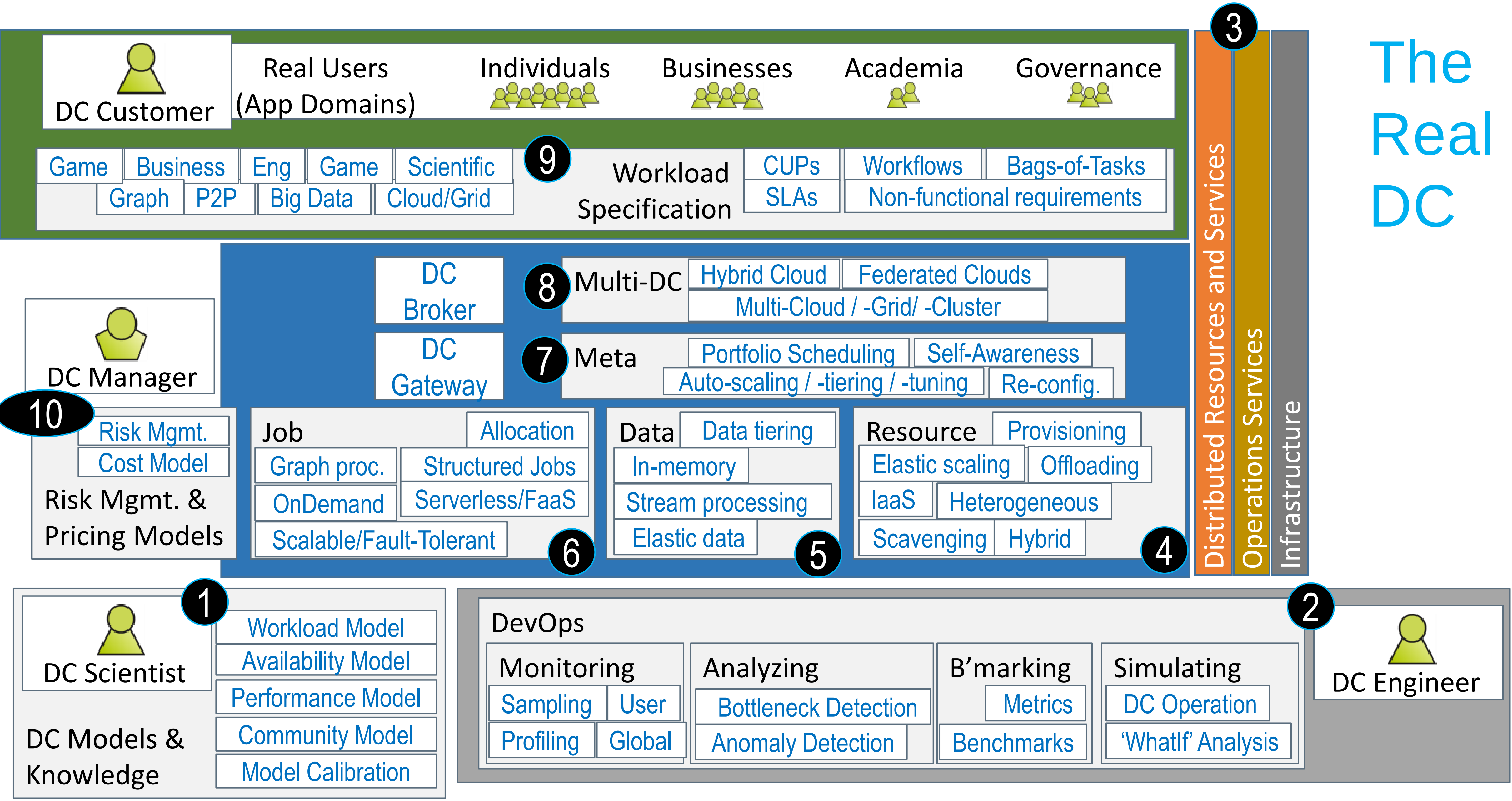
The Real DC



VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

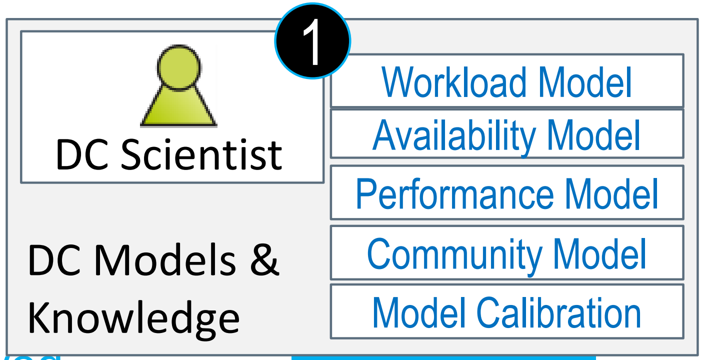

The Real DC
DC Customer
Real Users (App Domains)
Individuals
Businesses
Academia
Governance
Game
Business
Eng
Game
Scientific
Graph
P2P
Big Data
Cloud/Grid
9
Workload Specification
CUPs
SLAs
Workflows
Bags-of-Tasks
Non-functional requirements
3
Distributed Resources and Services
Operations Services
Infrastructure
DC Broker
8
Multi-DC
Hybrid Cloud
Federated Clouds
Multi-Cloud / -Grid/ -Cluster
DC Gateway
7
Meta
Portfolio Scheduling
Self-Awareness
Auto-scaling / -tiering / -tuning
Re-config.
DC Manager
10
Risk Mgmt.
Cost Model
Risk Mgmt. & Pricing Models
Job
Allocation
Graph proc.
Structured Jobs
OnDemand
Serverless/FaaS
Scalable/Fault-Tolerant
6
Data
Data tiering
In-memory
Stream processing
Elastic data
5
Resource
Provisioning
Elastic scaling
Offloading
IaaS
Heterogeneous
Scavenging
Hybrid
4
1
DC Scientist
Workload Model
Availability Model
Performance Model
Community Model
Model Calibration
DC Models & Knowledge
DevOps
Monitoring
Sampling
User
Profiling
Global
Analyzing
Bottleneck Detection
Anomaly Detection
B'marking
Metrics
Benchmarks
Simulating
DC Operation
'WhatIf' Analysis
2
DC Engineer

1. DC Models & Knowledge

Knowledge / Software tools / Data archives

- Various theories of how DCs operate
- Operational characterization and modeling
  - Largest study of global BitTorrent network (2005, 2010)
  - 1st comprehensive performance study of IaaS clouds (2008)
  - 1st performance variability (2011) & isolation (2011) studies
- Workload characterization and modeling
  - 1st characterization of scientific workflows (2008)
  - 1st model of grid computing workloads, bags-of-tasks (2008)
- Various characterization and modeling tools
- Various simulation tools: OpenDC (formerly DGSim)

- Data archives
  - Grid Workloads
  - Failure Traces
  - P2P Workloads
  - Game Traces
  - DC Traces (2015—ongoing)
  - Data collection & processing tools

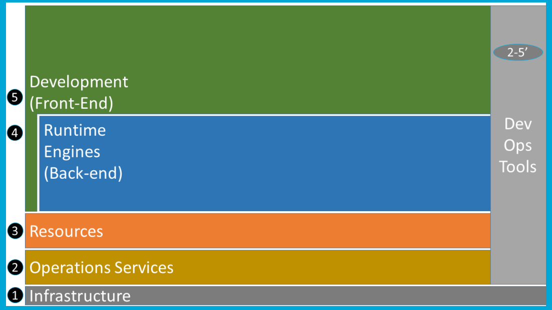


VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.



Alexandru Iosup   Vincent van Beek   Tim Hegeman



A Theory of Datacenter Stacks

How to Think About Datacenters?

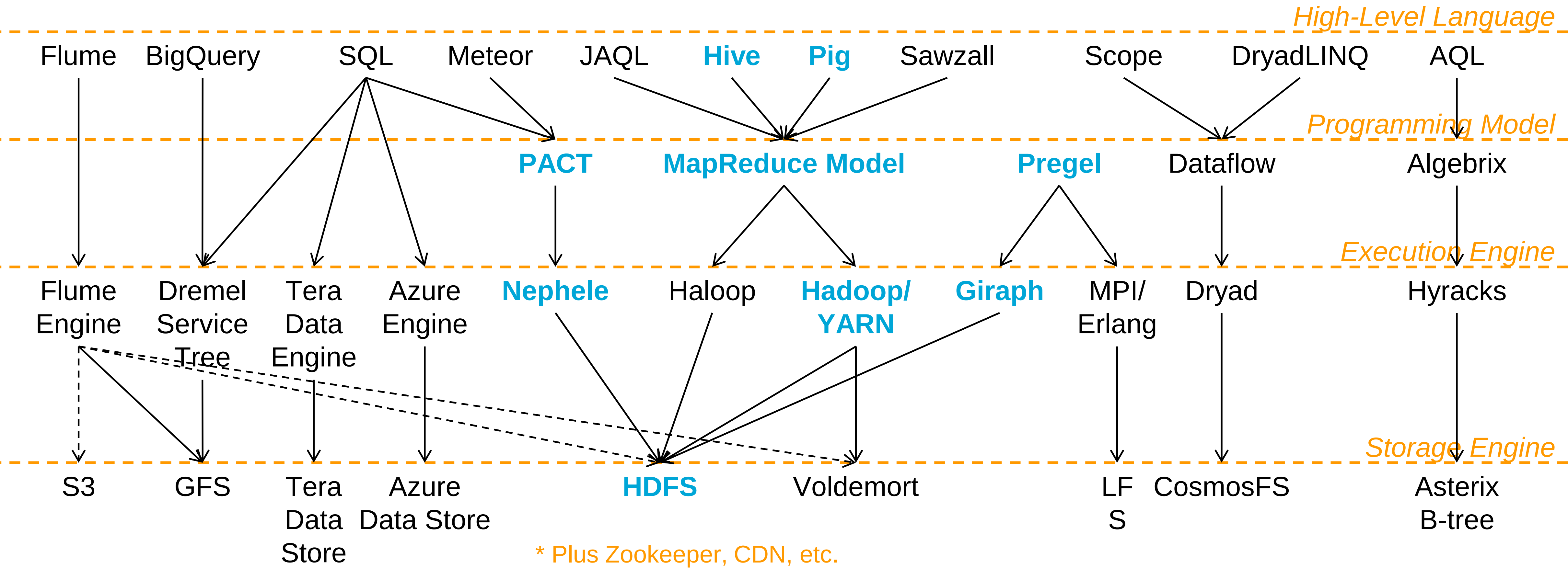
(Nov 2016)

Matt Turck's Big Data Landscape 2016
(zoom in on a part of the whole picture)

The Ecosystem Navigator Challenge:

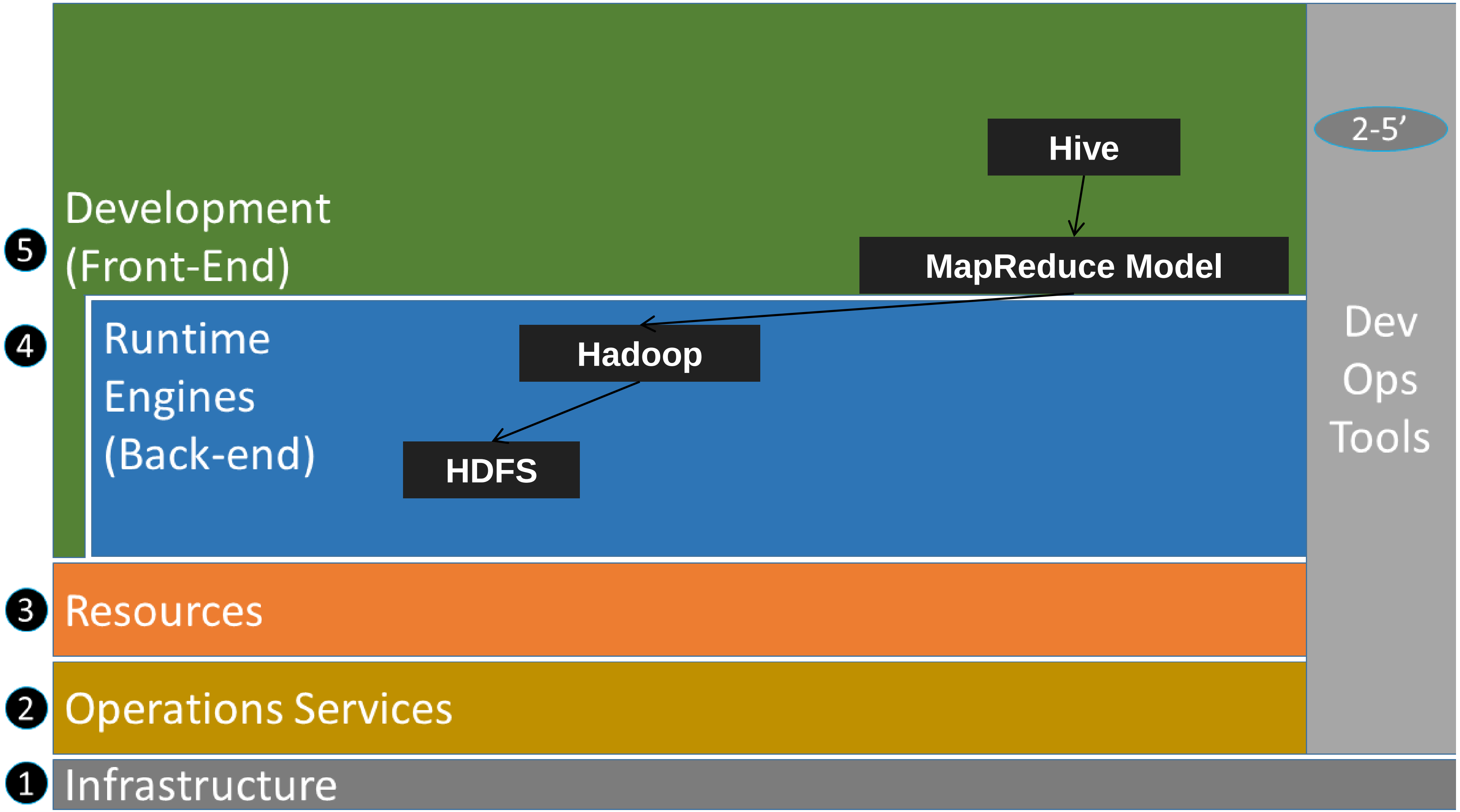
Can your team navigate this ocean of tools?

The Ecosystem Navigation Challenge

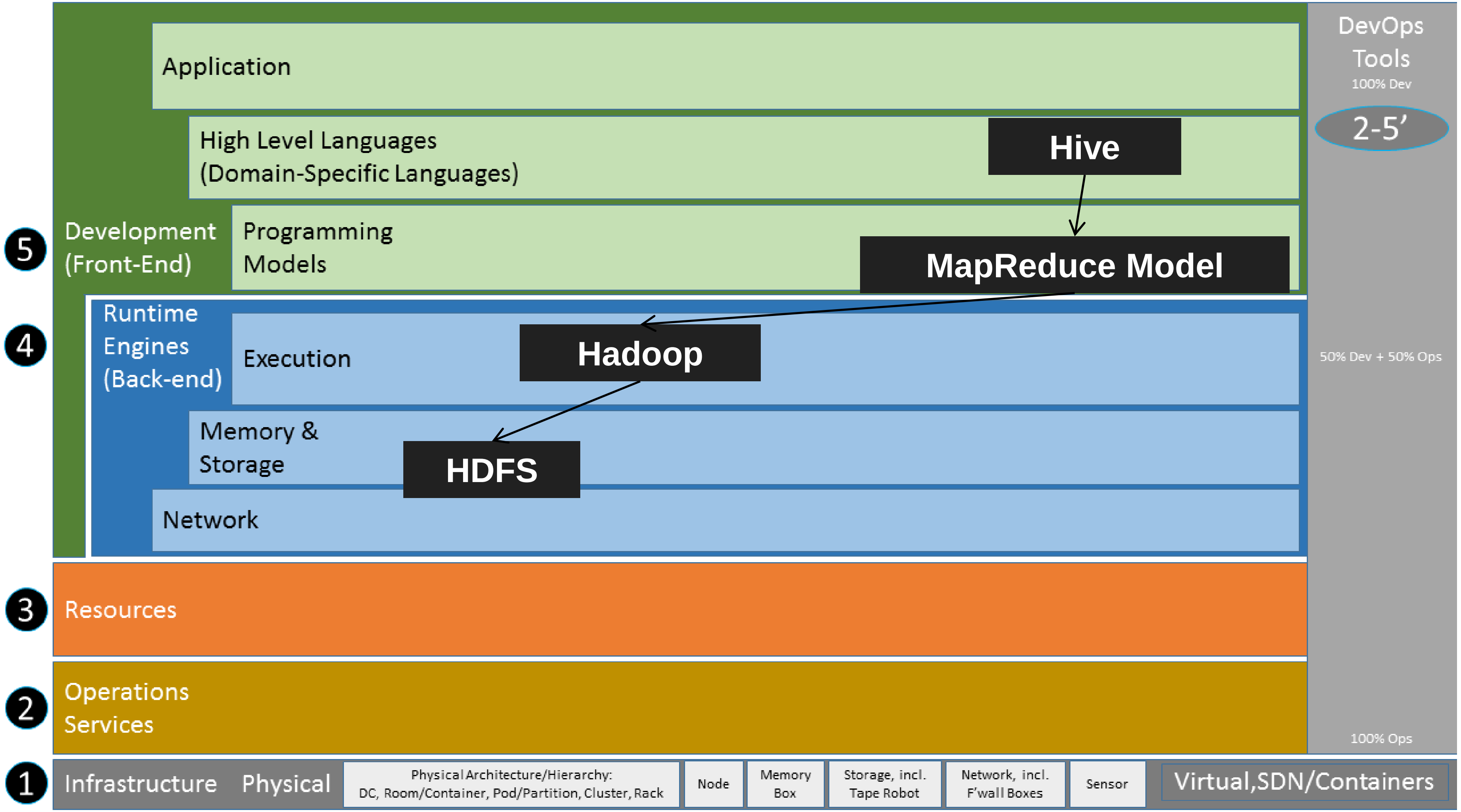


* Plus Zookeeper, CDN, etc.

A Reference Architecture for Massivizing Computer Systems



A Reference Architecture for Massivizing Computer Systems



Georgios Andreadis

Alexandru Iosup

A Theory of Datacenter Scheduling

How to Think About Datacenter Scheduling?

(Sep 2017)

(unpublished, so please do not record or share)

Alexandru Iosup, Tim Hegeman, Wing-Lung Ngai, Stijn Heldens, Ana Lucia Varbanescu, Yong Guo.

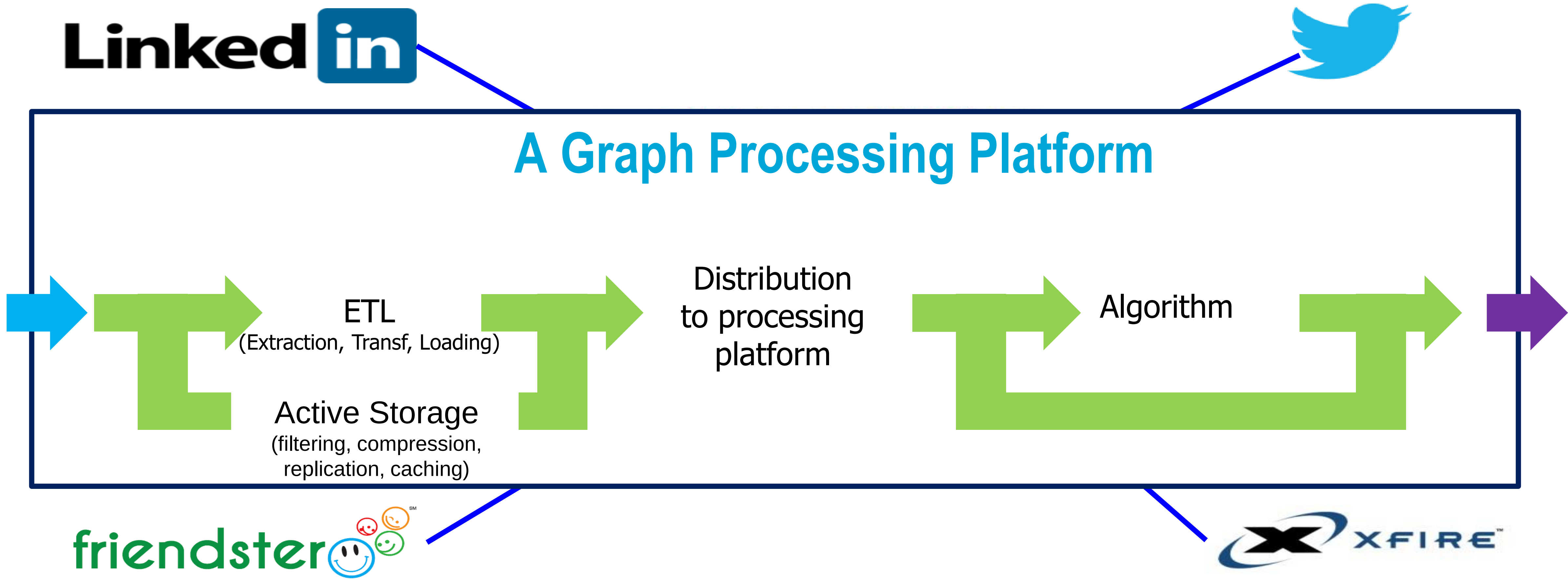
The performance of graph-processing systems is a non-trivial function of (Dataset, Algorithm, Platform)

Empirical laws of operation for modern data-processing systems

Guo, Biczak, Varbanescu, Iosup, Martella, Willke. How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis. IPDPS 2014: 395-404

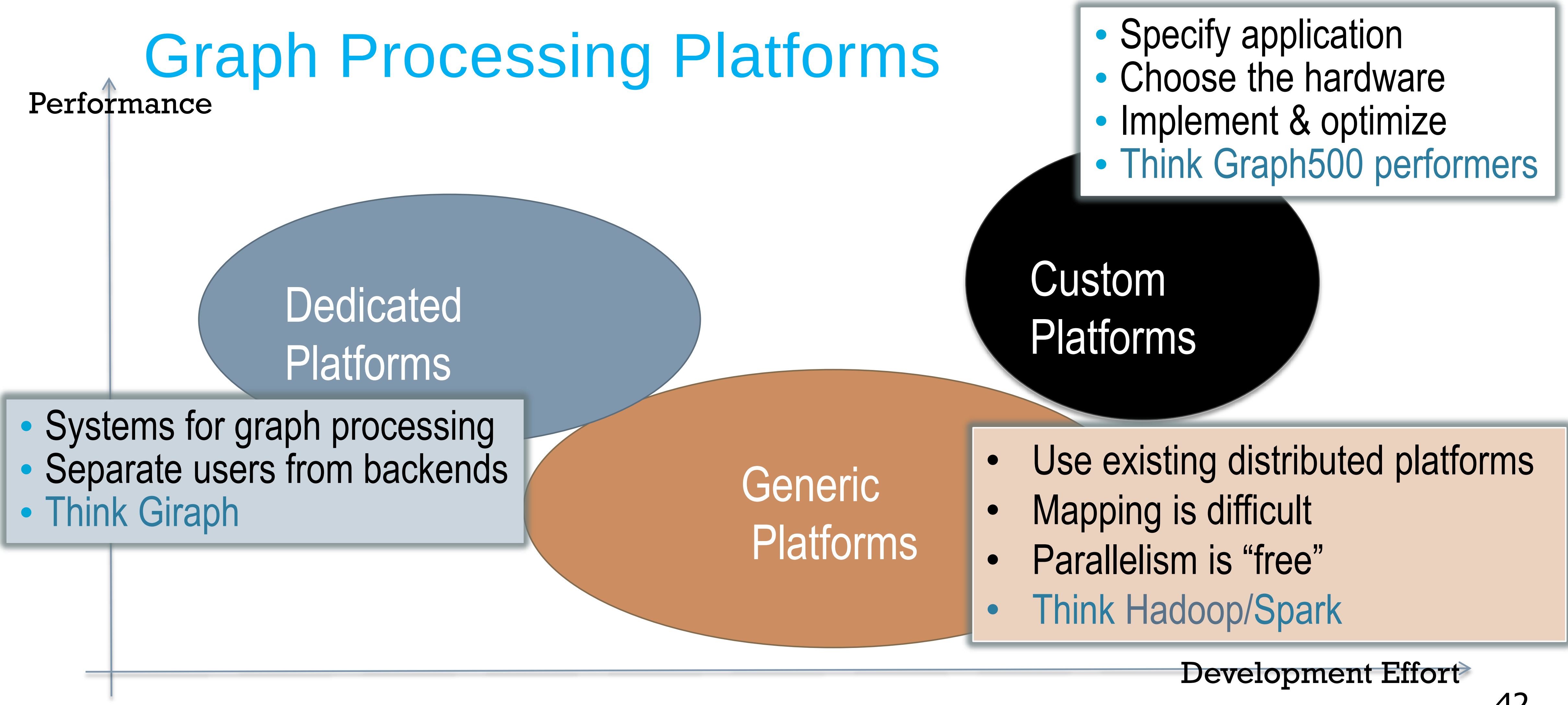
Guo, Varbanescu, Iosup, Epema: An Empirical Performance Evaluation of GPU-Enabled Graph-Processing Systems. CCGRID 2015: 423-432

How to do Graph Analysis? Graph Processing @large



Interactive processing not considered in this presentation.
Streaming not considered in this presentation.

Graph Processing Platforms



Results: Experimental Setup (1)

Graphalytics has been implemented for 3 community-driven platforms (Giraph, GraphX, PowerGraph) and 3 industry-driven platforms (PGX, GraphMat, OpenG).

PGX GraphMat OpenG

Iosup, Hegeman, Ngai, Heldens, Prat-Pérez, Manhardt, Chafi, Capota, Sundaram, Anderson, Tanase, Xia, Nai, Boncz. LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms. PVLDB 9(13): 1317-1328 (2016)

VU

Results: Experimental Setup (2)

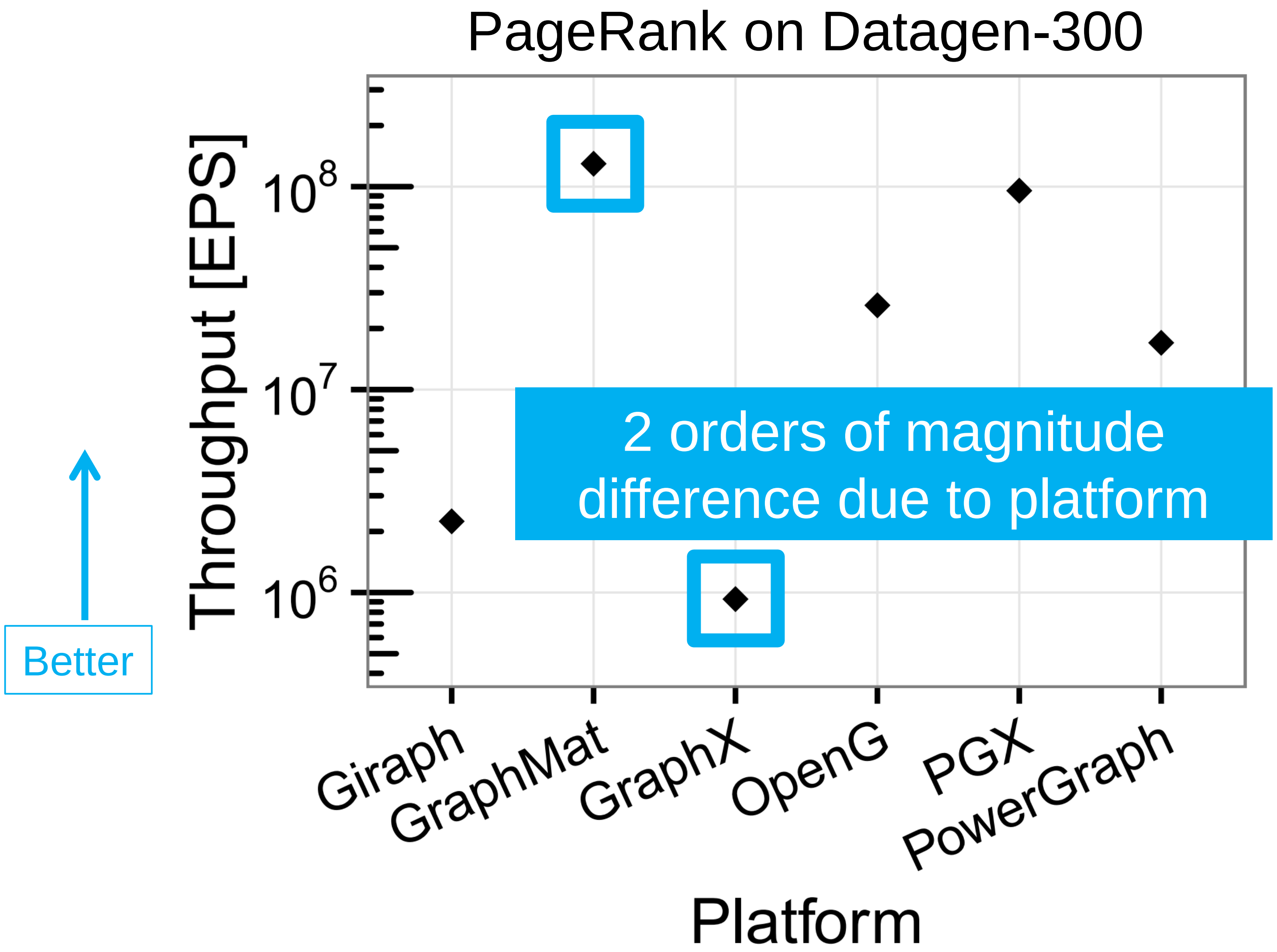
All experiments were performed by TU Delft on DAS-5 (Distributed ASCI Supercomputer, the Dutch national supercomputer for Computer Science research).

Environment: 1 machine (64GB, 2x8 cores)

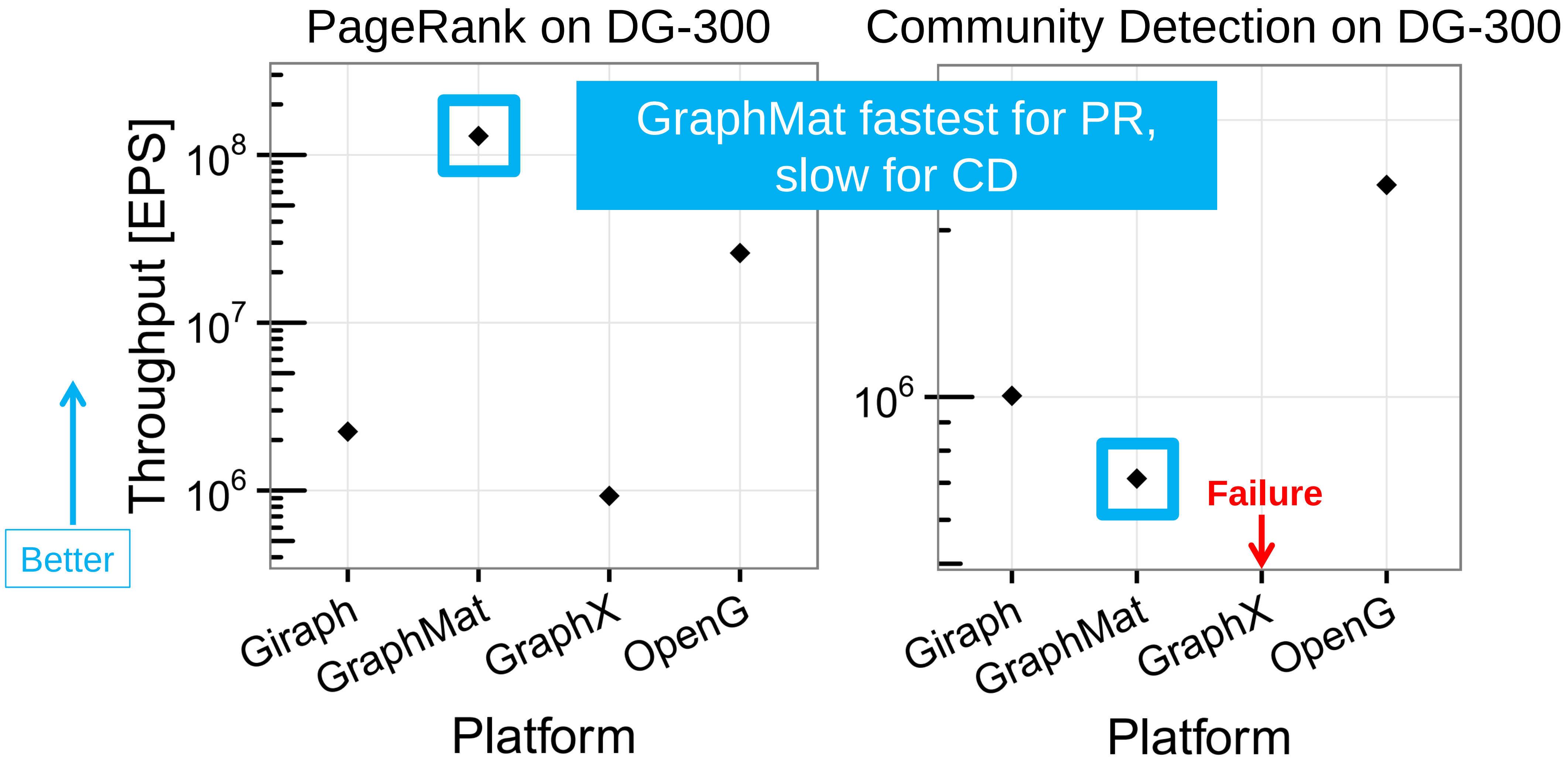
[experiments with up to 50 machines in VLDB article]

Capota, Hegeman, Iosup, Prat-Pérez, Erling, Boncz: Graphalytics: A Big Data Benchmark for Graph-Processing Platforms. GRADES@SIGMOD/PODS 2015: 7:1-7:6

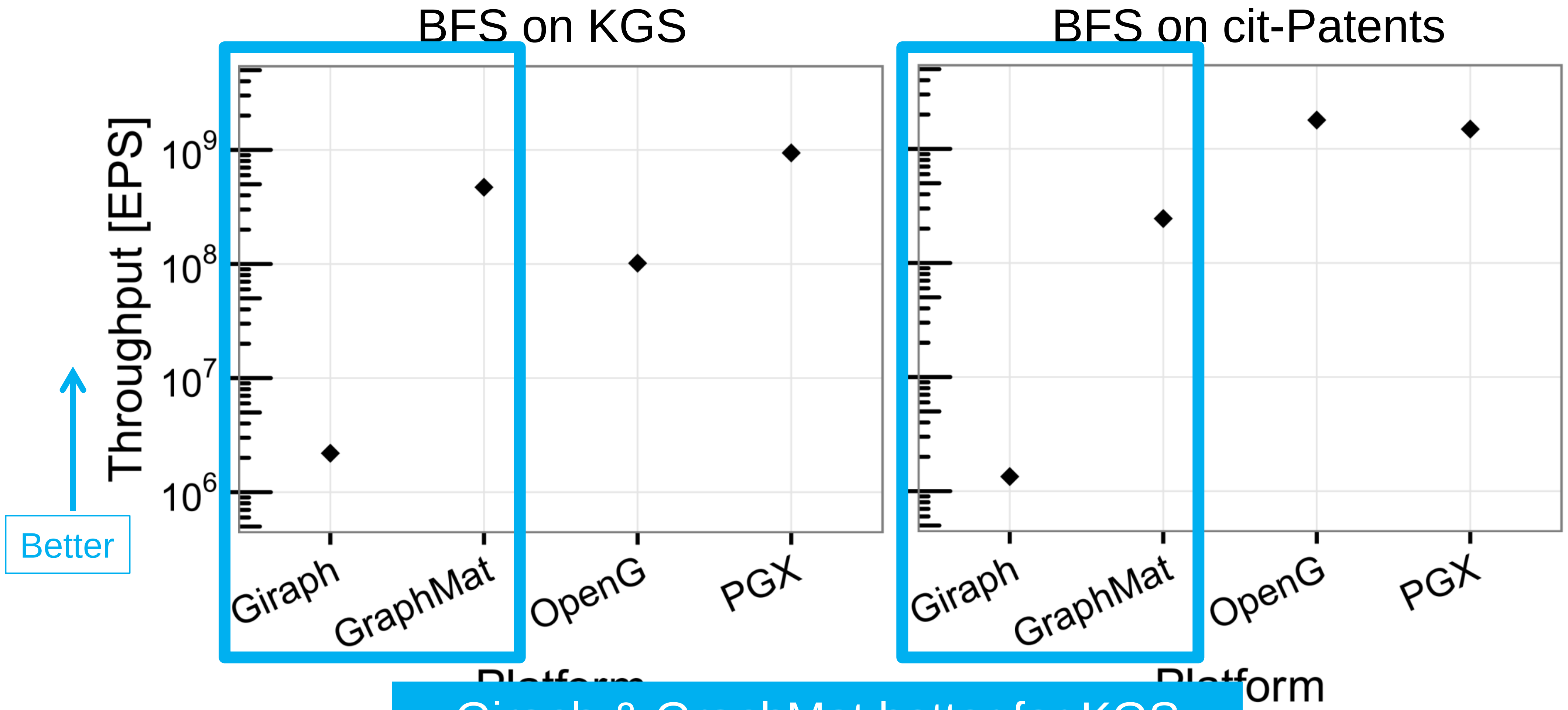
The Platform Has Large Impact



The Algorithm Has Large Impact

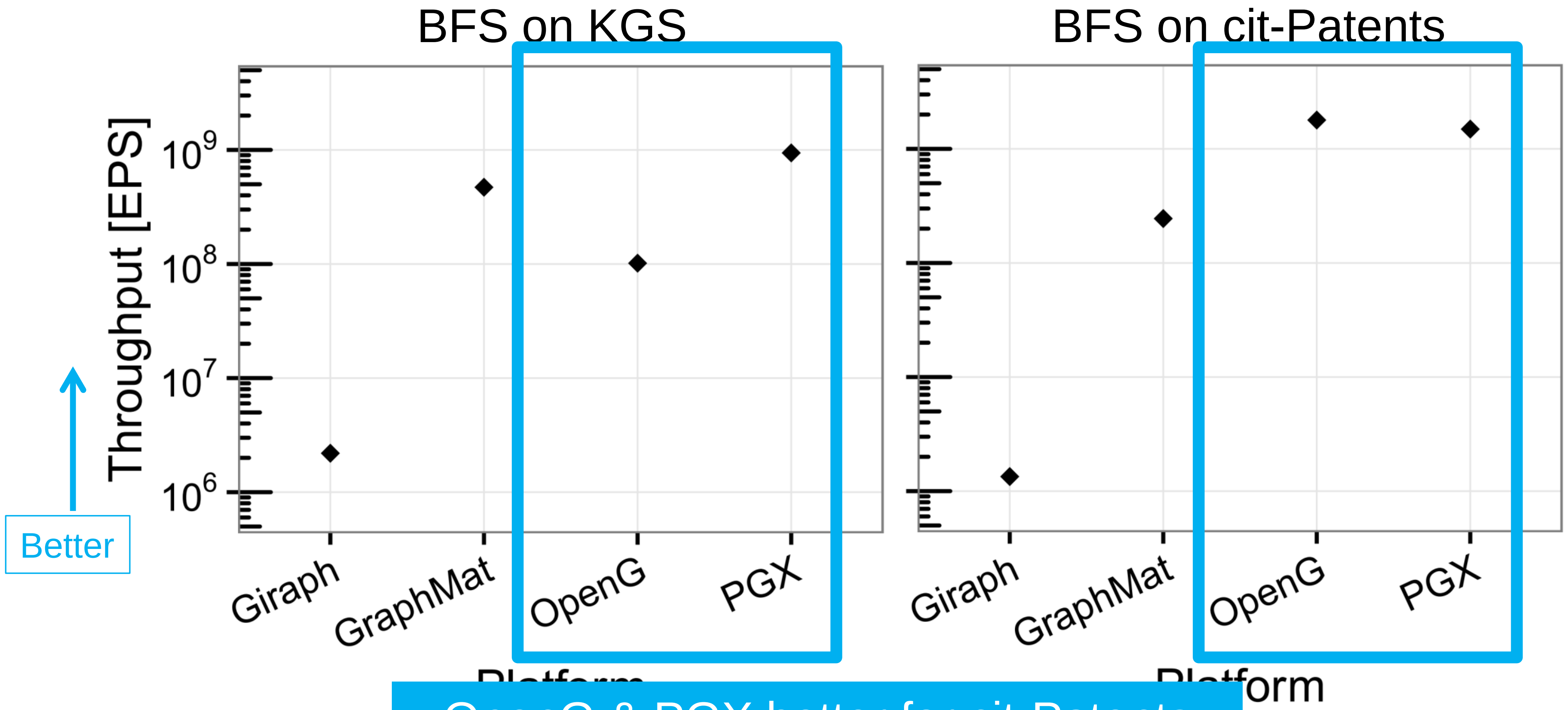


The Dataset Has Large Impact



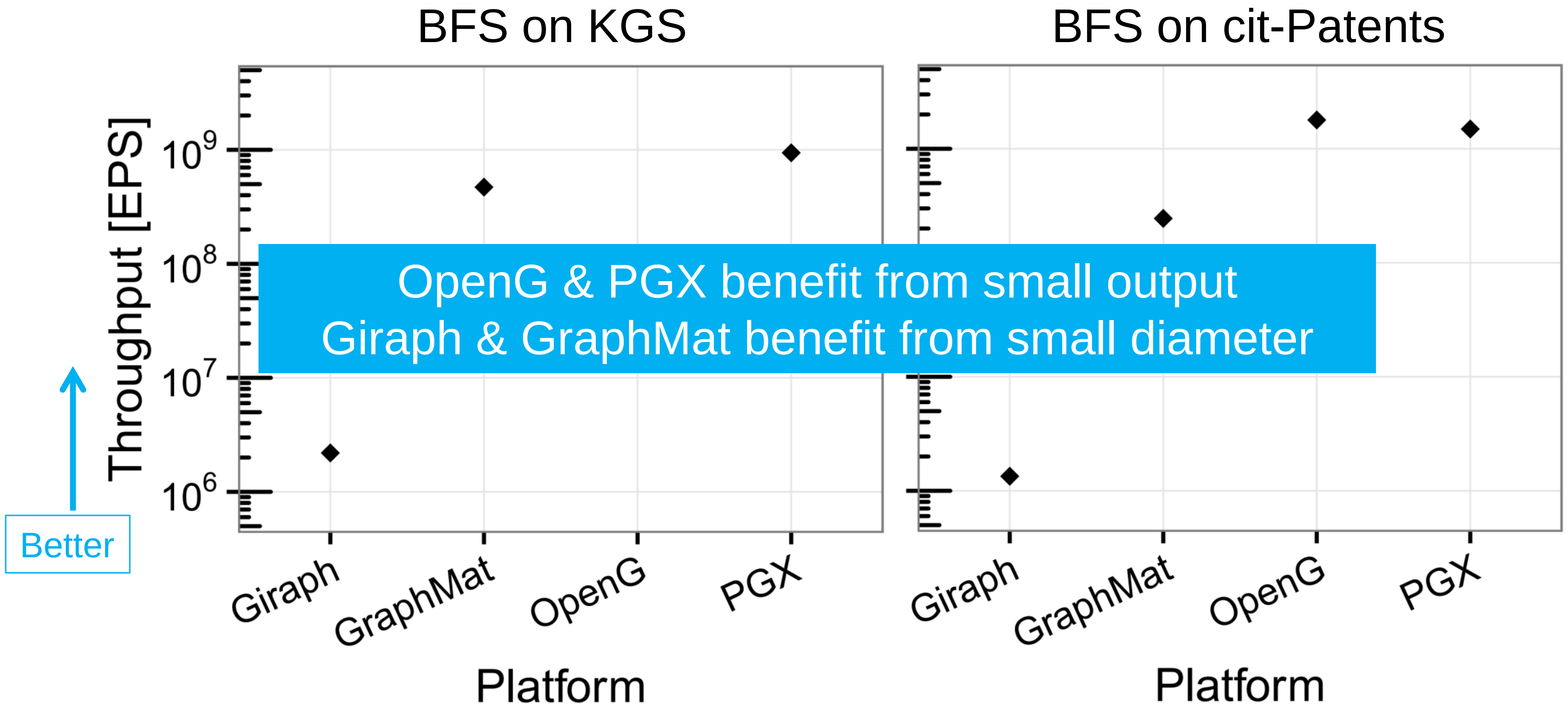
Giraph & GraphMat better for KGS

The Dataset Has Large Impact



OpenG & PGX better for cit-Patents

The Dataset Has Large Impact



OpenG & PGX benefit from small output
Giraph & GraphMat benefit from small diameter

For GPU-enabled systems

TOTEM

medusa-gpu
Medusa: Simplified Graph Processing on GPUs

mapgraph Beta
Massively Parallel Graph processing on GPU

General Challenges

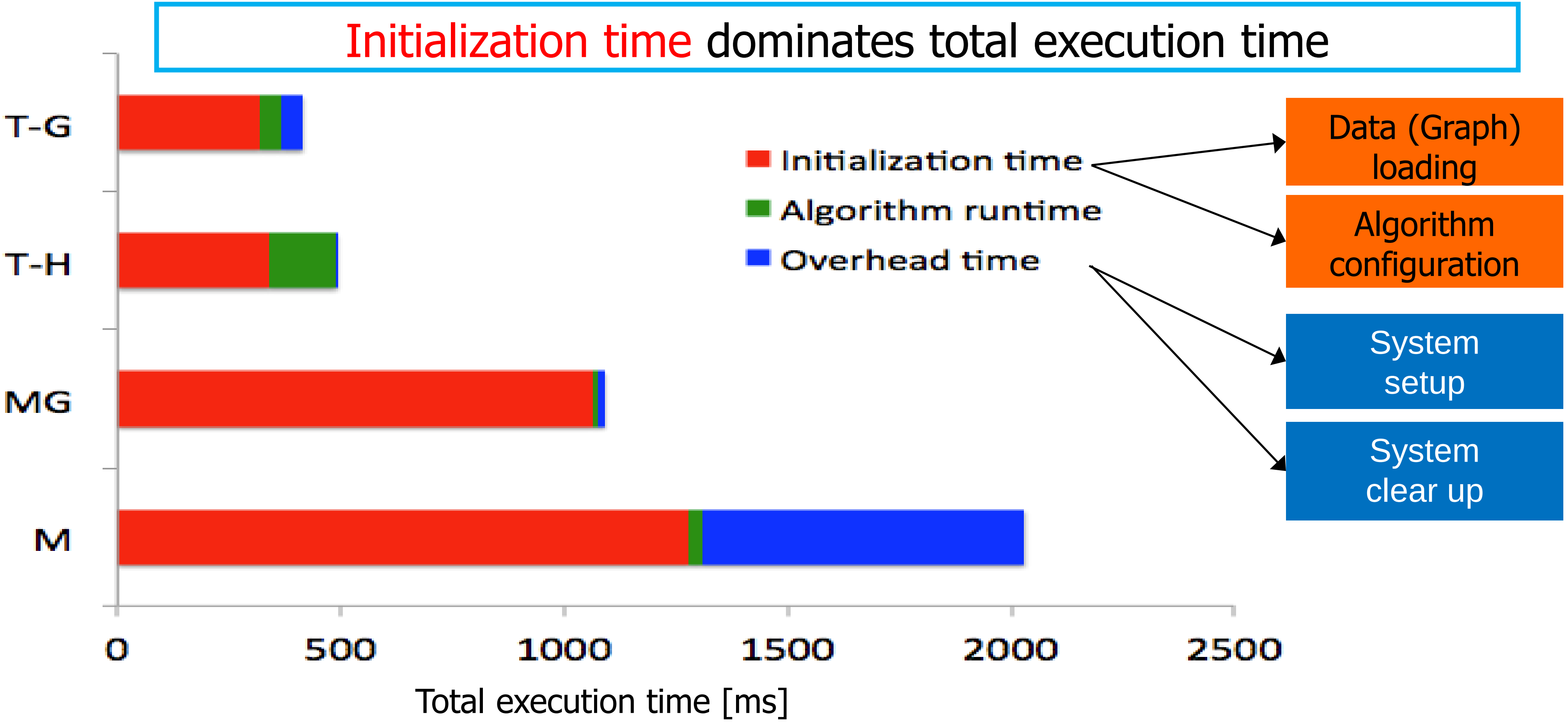
Performance Metrics + Graph Diversity + Algorithm Diversity

Challenges for evaluating GPU-enabled systems

In-memory graph formats + Optimization techniques + GPU generations

Y. Guo, A. L. Varbanescu, A. Iosup, and D. Epema, "An Empirical Performance Evaluation of GPU-Enabled Graph-Processing Systems," *CCGrid, 2015.*

Sample Result:
BFS Algo on Amazon Data for all systems



Y. Guo, A. L. Varbanescu, A. Iosup, and D. Epema, "An Empirical Performance Evaluation of GPU-Enabled Graph-Processing Systems," *CCGrid, 2015.*

Lessons learned

Performance of graph processing is a non-trivial function of
(Platform, Algorithm, Dataset, …), the P-A-D triangle

Understanding performance requires in-depth analysis
- We are building tools for manual/automated choke-point analysis

All current platforms can also have drawbacks
- Ease-of-use/programmability of a platform is very important
- Significant knowledge required to tune a system

The Datacenter Research Toolbox

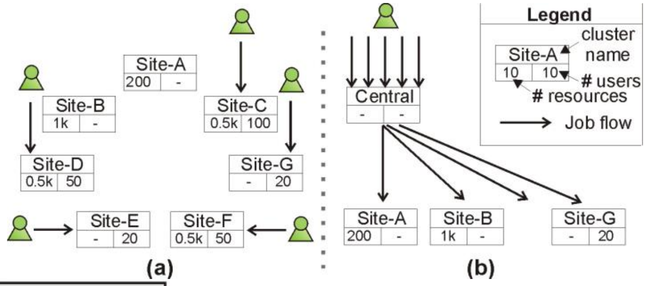
How to Explore Datacenter Technology? Open-Access Data Archives, Workload and Operational Models, plus many DevOps tools (monitoring, benchmarking, simulation)

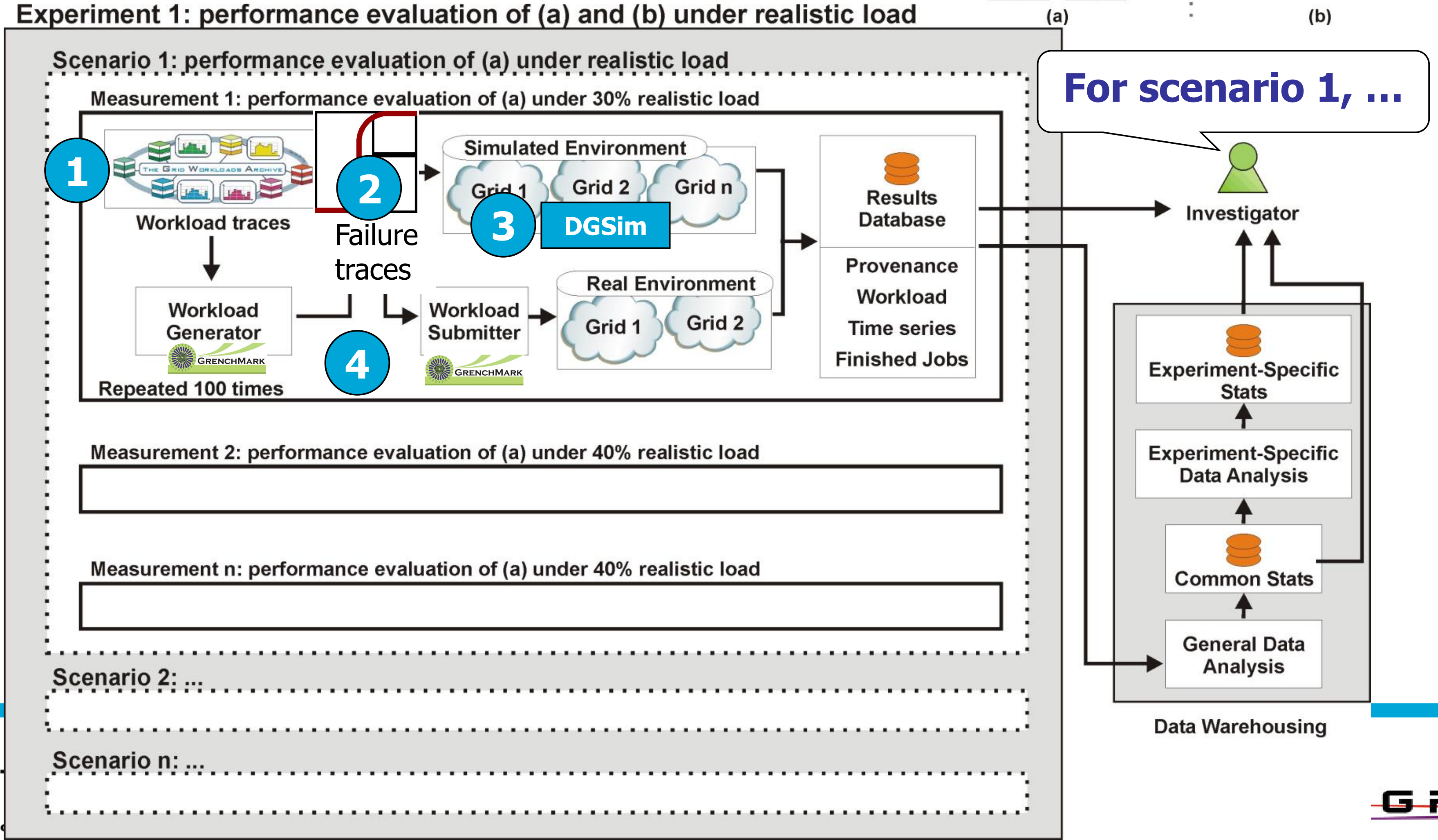
Key publications:

- Process for grids [JSSPP'06] and p2p systems [Sampling bias, EuroPar'10], and metrics for grids [JSSPP'07] and clouds [TOMPECS'17]
- Benchmarking software [Grenchmark, CCGrid'06] and [C-Meter] [CCGrid'09]
- Grid Workloads Archive [FGCS'08], workload models for Bags of Tasks [HPDC'08] and groups of jobs [EuroPar'07], and workload characterization for Bags of Tasks [Grid'06], workflows [EuroPar WS'08], and longitudinal study of grid workloads [IC'11]
- Failure Trace Archive [CCGrid'10] [JPDC'13], and models for resource availability [Grid'07] and correlated failures [Space-correlated failures, EuroPar'10] [Time-correlated failures, Grid'10]
- Game Trace Archive [NETGAMES'12], characterization of workload [SC|08] [HAVE'12], mobility [NOSSDAV'14], and toxicity [NETGAMES'15], and models of player mobility [MMVE'14], social apps [ICPE'13 WiP], and player-interaction graphs [COMSNETS'13] [IC'14] [TKDD'15] [TOMMCAP'16]
- P2P Trace Archive [CoNext'10 WS], models for p2p flashcrowds [P2P'11], longitudinal studies of P2P systems [CCGrid'06 WS] [BTWorld, HPDC'10 WS]
- Simulation [DGSim, EuroPar'08] and [OpenDC, ISPDC'17]

A Grid Research Toolbox

- Hypothesis: (a) is better than (b).





Free Open-Access Data Archives

(2006 and 2008) The Grid Workloads Archive (GWA)
(2010) The Peer-to-Peer Trace Archive (P2PTA)
(2012) The Game Trace Archive (GTA)
(2010 and 2013) The Failure Trace Archive (FTA)

Iosup, Dumitrescu, Epema, Li, Wolters. How are Real Grids Used? The Analysis of Four Grid Traces and Its Implications. GRID 2006: 262-269

Iosup, Li, Jan, Anoep, Dumitrescu, Wolters, Epema. The Grid Workloads Archive. Future Generation Comp. Syst. 24(7): 672-686 (2008)

Zhang, Iosup, Pouwelse, Epema. The peer-to-peer trace archive: design and comparative trace analysis. ACM CoNEXT Student Workshop 2010.

Guo, Iosup. The Game Trace Archive. NetGames 2012: 1-6

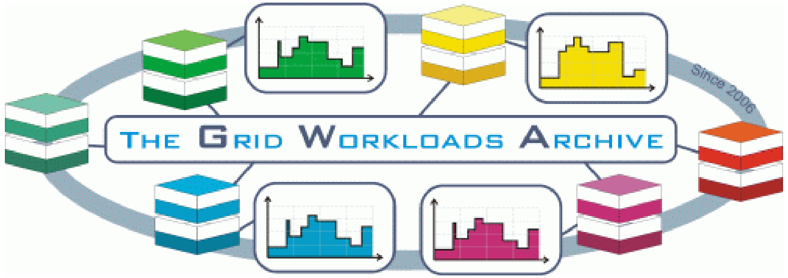
Kondo, Javadi, Iosup, Epema. The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems. CCGRID 2010: 398-407

Javadi, Kondo, Iosup, Epema. The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. JPDC 73(8): 1208-1223 (2013)

The Grid Workloads Archive [1/3]
Motivation and Goals

- Motivation: little is known about real grid use
  - No grid workloads (except "my grid")
  - No standard way to share them

- **The Grid Workloads Archive: easy to share grid workload traces and research associated with them**
  - **Understand** how real grids are used
  - **Address** the challenges facing grid resource management (both research and practice)
  - **Develop and test** grid resource management solutions
  - **Perform** realistic simulations



http://gwa.ewi.tudelft.nl

A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, D. Epema, The Grid Workloads Archive, FGCS 24, 672–686, 2008.

The Grid [and Cloud] Workloads Archive [2/3]
Content

ID	System	Period	Number of observed Sites	CPUs	Jobs	Groups	Users
GWA-T-1	DAS-2	02/05-03/06	5	400	602K	12	332
GWA-T-2	Grid'5000	05/04-11/06	15	~2500	951K	10	473
GWA-T-3	NorduGrid	05/04-02/06	~75	~2000	781K	106	387
GWA-T-4	AuverGrid	01/06-01/07	5	475	404K	9	405
GWA-T-5◇	NGS	02/03-02/07	4	~400	632K	1	379
GWA-T-6◇							206
GWA-T-7‡							18
GWA-T-8‡							19
GWA-T-9‡	TeraGrid	08/05-03/06	1*	96	1.1M	26	121
	Total	13.51 yrs	136	>10000	>7M	191	2340
	Average	1.5 yrs	15	1151	>750K	21	>250

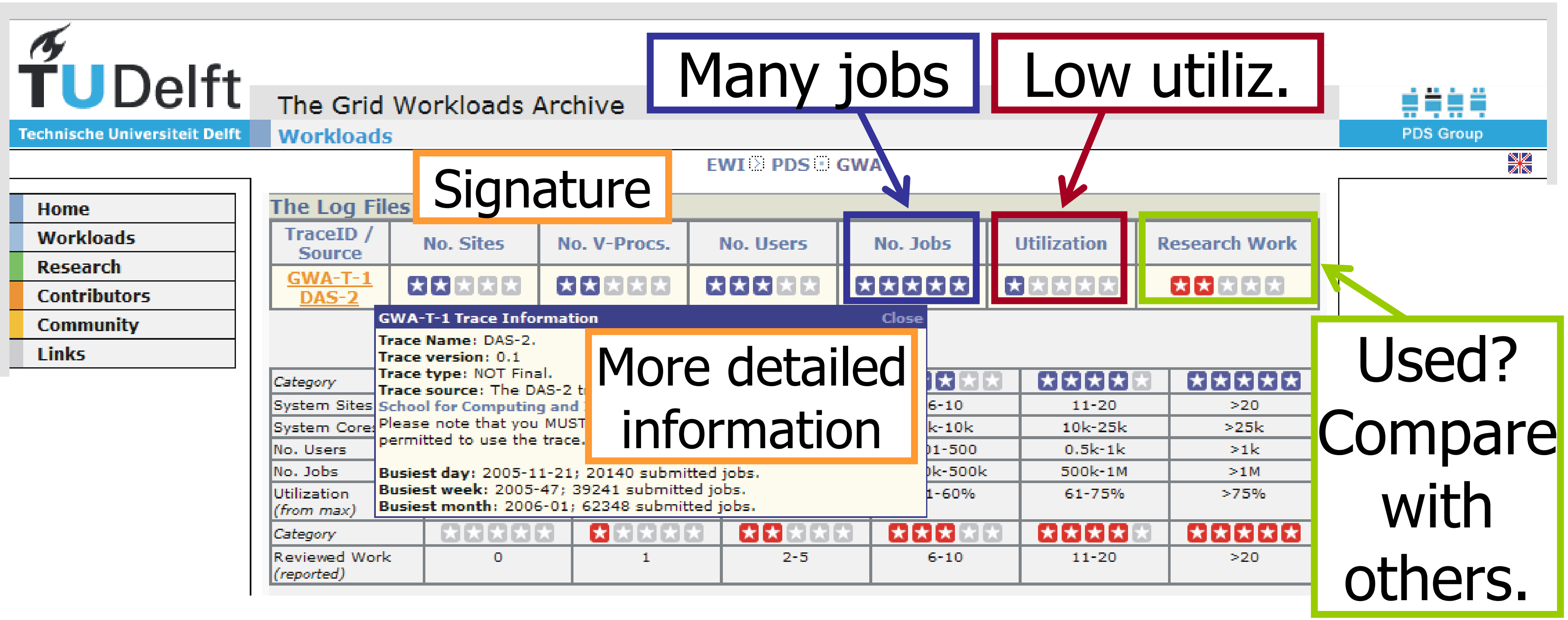
http://gwa.ewi.tudelft.nl

10+ traces online

2 cloud traces

A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, D. Epema, The Grid Workloads Archive, FGCS 24, 672–686, 2008.

The Grid Workloads Archive [3/3]
Presentation



- Workload signature: simple six-category description
- Easy to see which traces are **fit**/**unfit** for your experiment

A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, D. Epema, The Grid Workloads Archive, FGCS 24, 672–686, 2008.

The Failure Trace Archive [1/2]
Motivation and Goals

- Motivation: grid resources and jobs fail to work
  - No grid failure model (except "my/your/our grid failure model")
  - No standard way to share them

- **The Failure Trace Archive: centralized public repository of availability traces of parallel and distributed systems, and tools for their analysis**
  - **Understand** real failures
  - **Facilitate** the design, validation, and comparison of fault-tolerant models and algorithms
  - **Improve** the reliability of distributed systems

http://fta.inria.fr

D. Kondo, B. Javadi, A. Iosup, D. Epema, The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems, CCGrid 2010 (accepted)

The Failure Trace Archive [2/2]
Content & Presentation

System	Type	# of Nodes	Target Component	Period	Year
SETI@home	Desktop Grid	226,208	CPU	1.5 years	2007-2009
Overnet	P2P	3,000	host	2 weeks	2003
Microsoft	Desktop	51,663	host	35 days	1999
LANL	SMP, HPC Clusters	4750	host	9 years	1996-2005
HPC2	HPC Clusters	256	IO	2.5 years	1996-2005
Web sites	Web servers	129	host	8 months	2001-2002
DNS	DNS servers	62,201	host	2 weeks	2004
PlanetLab	P2P	200-400	host	1.5 year	2004-2005
Grenouille03	DSL	4800	host	1 year	2003
Grenouille05	DSL	4800	host	1 year	2005
EGEE	Grid	2500 queues	CE queue	1 month	2007
Grid'5000	Grid	1288	host	1.5 years	2005-2006
Notre Dame	Desktop Grid	700	CPU, host	6 months	2007
ucb94	Desktop Grid	85	CPU	46 days	1994
sdsc03	Desktop Grid	275	CPU	1 month	2003
lri05	Desktop Grid	40	CPU	1 month	2005

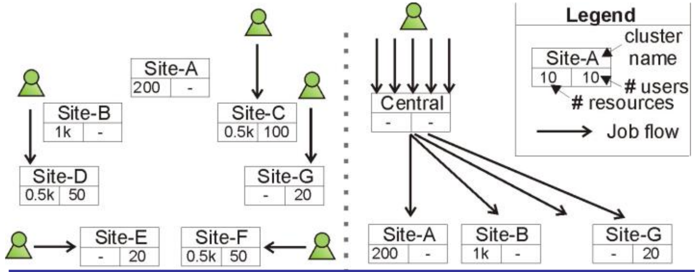
http://fta.scem.uws.edu.au/

15+ traces online

Javadi, Kondo, Iosup, Epema. The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. JPDC 73(8): 1208-1223 (2013)

Goal and Challenges

- Simulate various grid resource management architectures
  - Multi-cluster grids
  - Grids of grids (THE grid)



Two GRM architectures

- Challenges
  - Many types of architectures
  - Generating and replaying grid workloads
  - Management of the simulations
    - Many repetitions of a simulation for statistical relevance
    - Simulations with many parameters
    - Managing results (e.g., analysis tools)
    - Enabling collaborative experiments

The Peer-to-Peer Trace Archive (P2PTA)
Unified Data Format for P2P Traces

Boxun Zhang and Alexandru Iosup

Goal: **Provide a unified data format for storing data traces of different P2P applications.**

http://p2pta.ewi.tudelft.nl/

20+ traces online

Motivation

- Comparison of different p2p traces
  - Performance evaluation
- Setting up input workload for experiments
  - Trace-based simulations
- Data exchange in the p2p research community

Boxun Zhang, Alexandru Iosup, et al. The Peer-to-Peer Trace Archive: design and comparative trace analysis.ACM CoNEXT'10 Student Workshop. Article 21

P2PTA = 15+ Traces, Spanning 10+ Years

Trace ID	Community	Measurement Period	Sampling Rate	No. Files	No. Sessions	Traffic	Contributor
T1'03	SuprNova, (general)	06 Dec 2003 ~ 17 Jan 2004	2.5 min	12	28,423,470	n/a	PDS, TU Delft
T2'05	ThePirateBay, (general)	06 May 2005 ~ 11 May 2005	2.5 min	4800	35,881,338	12 PB/year	PDS, TU Delft
T3'05	Filelist.org, (general)	14 Dec 2005 ~ 04 Apr 2006	6 min	3000	2,172,738	n/a	PDS, TU Delft
T4'05	LegalTorrents.com, (general)	22 Mar 2005 ~ 19 Jul 2005	5 min	41	n/a	698 GB/year	PDS, TU Delft
T4'09							
T5'05							
T11'03	alluvion.org, (general)	27 Oct 2003 ~ 26 Jan 2004	30 min	1,476	173,532	348 GB/year	UMASS
T12'04	Gnutella, (general)	19 Mar 2004 ~ 28 Mar 2004	n/a	2,896,885	n/a	n/a	uni-leipzig
T13'03	eDonkey, (general)	14 Oct 2003 ~ 16 Oct 2003	n/a	1,282,420	n/a	n/a	Fabrice Le Fessant
T13'04	eDonkey, (general)	09 Dec 2003 ~ 02 Feb 2004	n/a	23,965,651	n/a	n/a	Fabrice Le Fessant
T14'07	PP Live network	-	-	-	-	-	Long Vu
T15'05	Skype network	-	-	-	-	-	Saikat Guha
T16'10	BTWorld	-	-	-	-	-	PDS, TU Delft
T17'14	Mainline DHT	-	-	-	-	-	PDS, TU Delft

Simulation of DC Technology

(2006—2015) The Delft Grid Simulator (DGSim)
(2016—ongoing) OpenDC: collaborative exploration of DC technology

Iosup, Sonmez, Epema. DGSim: Comparing Grid Resource Management Architectures through Trace-Based Simulation. Euro-Par 2008: 13-25

Sonmez, Yigitbasi, Abrishami, Iosup, Epema. Performance analysis of dynamic workflow scheduling in multicluster grids. HPDC 2010: 49-60

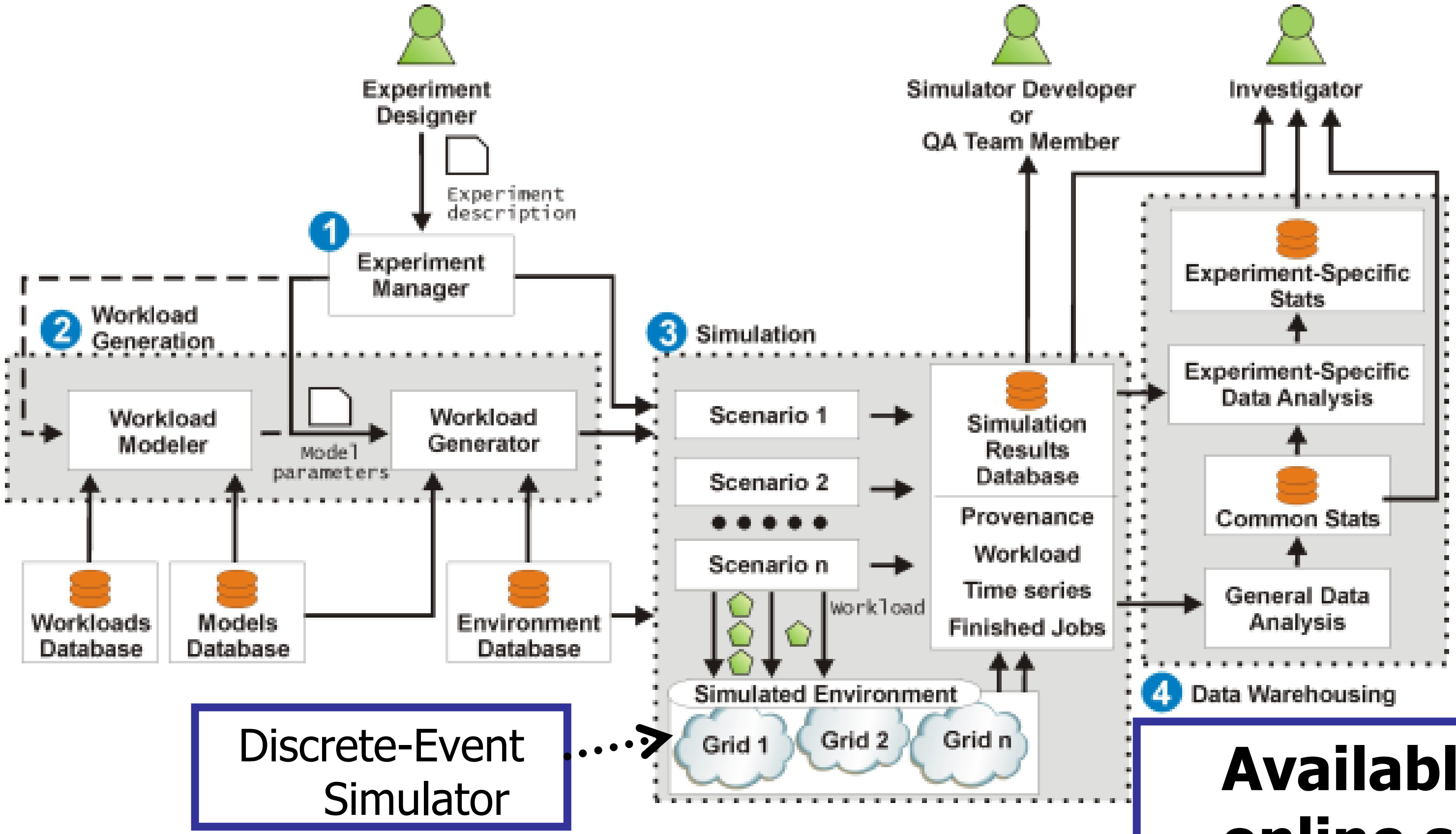
Deng, Song, Ren, Iosup. Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC 2013: 55:1-55:12

van Beek, Donkervliet, Hegeman, Hugtenburg, Iosup. Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters. IEEE Computer 48(7): 46-54 (2015)

Iosup, Andreadis, van Beek, Bijman, van Eyk, Neacsu, Overweel, Talluri, Versluis, Visser. The OpenDC Vision: Towards Collaborative Datacenter Simulation and Exploration for Everybody. ISPDC 2017.

DGSim: Simulating Multi-Cluster Grids [2/2]

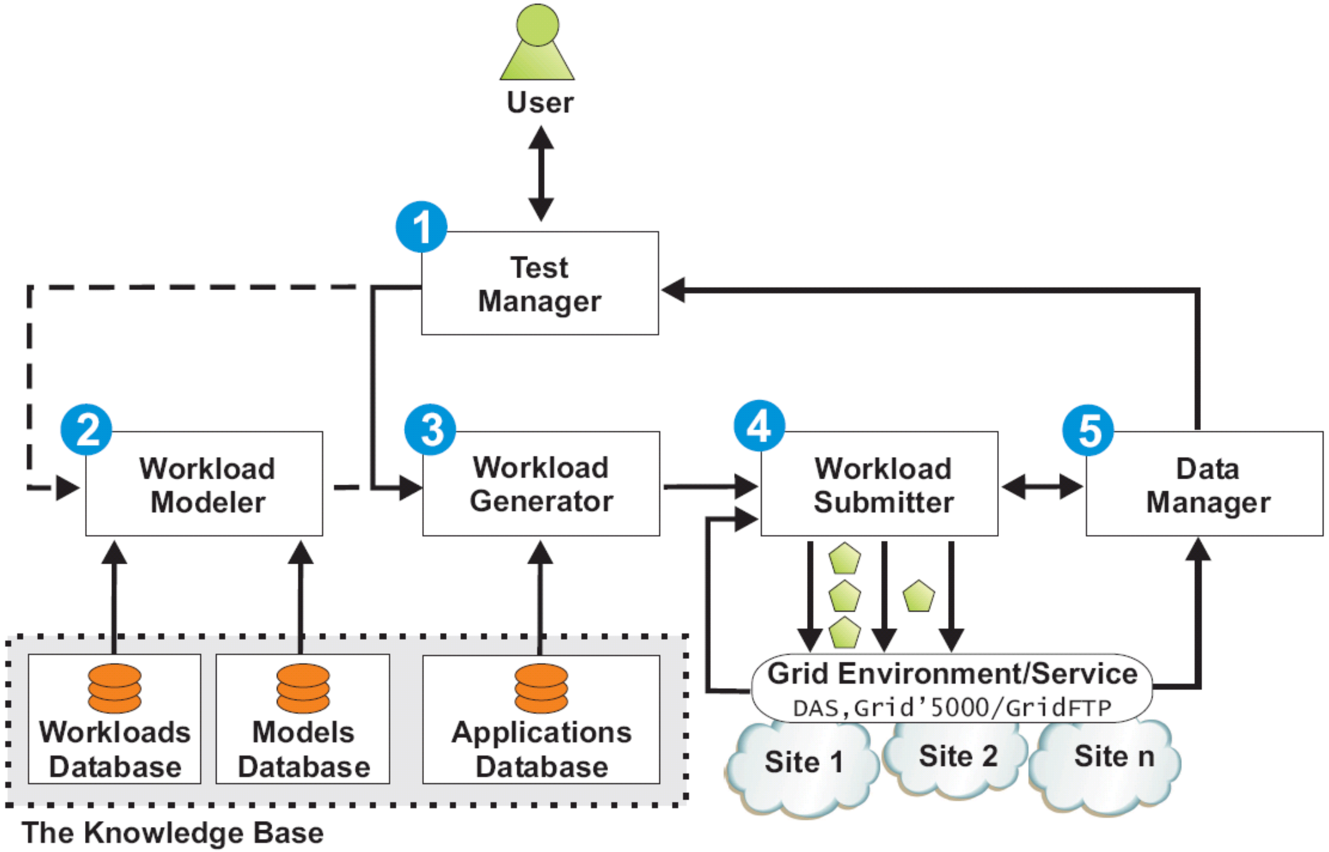
Overview



Available online soon

A. Iosup, O. Sonmez, D. Epema: DGSim: Comparing Grid Resource Management Architectures through Trace-Based Simulation. Euro-Par 2008: 13-25

GrenchMark: Testing in LSDCSs [1/4]
Architecture Overview



Iosup and Epema: GRENCHMARK: A Framework for Analyzing, Testing, and Comparing Grids. CCGRID 2006: 313-320

… but More Complicated Than You Think

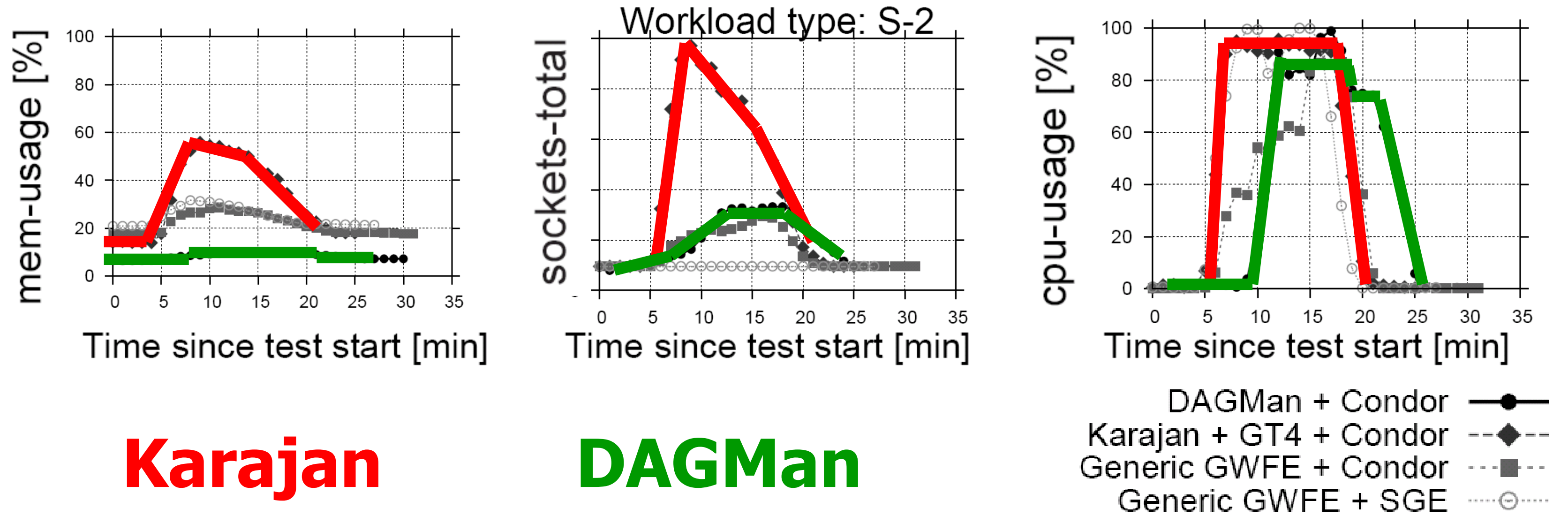
- **Workload structure**
  - User-defined and statistical models
  - Dynamic jobs arrival
  - Burstiness and self-similarity
  - Feedback, background load
  - Machine usage assumptions
  - Users, VOs
- **Metrics**
  - A(W) Run/Wait/Resp. Time
  - Efficiency, MakeSpan
  - Failure rate [!]
- **Notions**
  - Co-allocation, interactive jobs, malleable, moldable, …
- **Measurement methods**
  - Long workloads
  - Saturated / non-saturated system
  - Start-up, production, and cool-down scenarios
  - Scaling workload to system
- **Applications**
  - Synthetic
  - Real
- **Workload definition language**
  - Base language layer
  - Extended language layer
- **Other**
  - Can use the same workload for both simulations and real environments

GrenchMark: Performance Evaluation in Grids [4/4]

Raw Perf.: Performance vs. Res. Consumption

Middleware	MS [s]
DAGMan	1,327 ± 138
Karajan	1,111 ± 154

Karajan performs better than DAGMan, but runs quickly out of resources.



C. Stratan, A. Iosup, D. Epema: A performance study of grid workflow engines. GRID 2008: 25-32

TU Delft
Delft University of Technology

NWO STW COMMIT/

OpenDC

Collaborative Datacenter Simulation and Exploration for Everybody

Prof. dr. ir. Alexandru Iosup

Project Lead

Leon Overweel

Product Lead and Software Engineer responsible for the web server, database, and API specification

Georgios Andreadis

Software Engineer responsible for the frontend web application and splash page

Fabian S. Mastenbroek

Team OpenDC

Sacheendra Talluri

M.Sc. student, TU Delft

Vincent van Beek

Ph.D. student, TU Delft

Tim Hegeman

M.Sc. student, TU Delft

Jesse Donkervliet

M.Sc. student, TU Delft

Laurens Versluis

Ph.D. student, Vrije Universiteit Amsterdam

Mihai Neacsu

M.Sc. student, Vrije Universiteit Amsterdam

Why do we need **Open**DC?

The **datacenter** industry…

- "Produces" **cloud services**
- Is worth over **$15 bn** & growing
- Has many hard-to-grasp **concepts** (scheduling, workloads, devops, ...)
- Is **understaffed**

OpenDC focuses on...

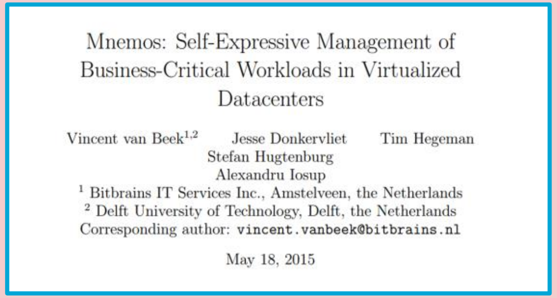
1. **Exploration**
2. **Scientific method**
3. **Education**
4. **Toolkit for many: software & data**

Take-Home: **OpenDC** brings to the table…

1. Datacenter Technology & Methods

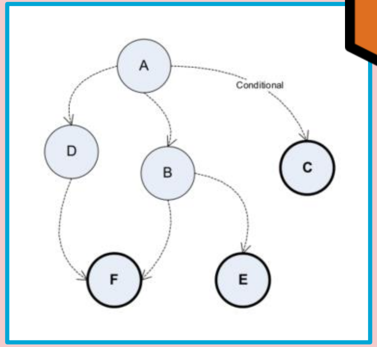
- Risk Analysis + Management
- Efficiency → SME Availability
- Heterogeneity

2. Scientific Methods

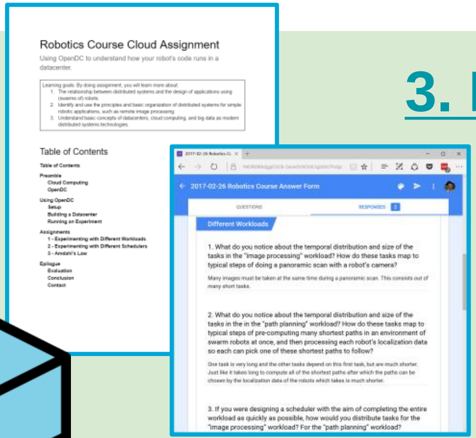

Mnemos: Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters

Vincent van Beek[1,2] Jesse Donkervliet Tim Hegeman
Stefan Hugtenburg
Alexandru Iosup
[1] Bitbrains IT Services Inc., Amstelveen, the Netherlands
[2] Delft University of Technology, Delft, the Netherlands
Corresponding author: vincent.vanbeek@bitbrains.nl

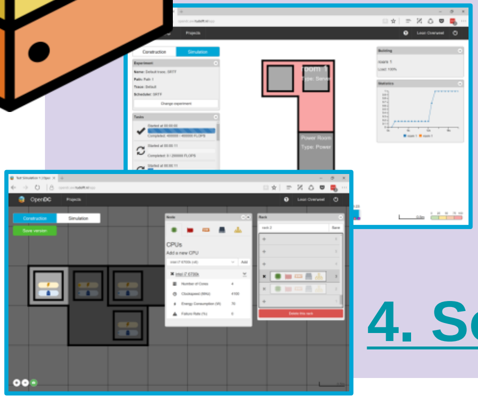
May 18, 2015



3. Education Practices



4. Software & Data Artifacts



OpenDC 1. Datacenter Tech. & Methods

Explore a **variety of concepts**...

- Scalability + Elasticity
- Efficiency for SMEs + DCs
- Availability + Reliability
- Risk Analysis + Management
- User + DC Heterogeneity

… with **PhD**, **MSc**, and **BSc** projects.

- Availability + Availability-on-Demand
- Complex Workflow Scheduling
- Application Auto-Scaling
- FaaS Management and Applications
- Memory-Based Storage
- Portfolio Scheduling

OpenDC 3. Education Practices

OpenDC software **already used** for:

M.Sc. **Project-Based Learning** @ VUA & TUD

B.Sc. Honours Programme **Classroom-Based Courses**

B.Sc. Honours Programme **Project-Based Learning**

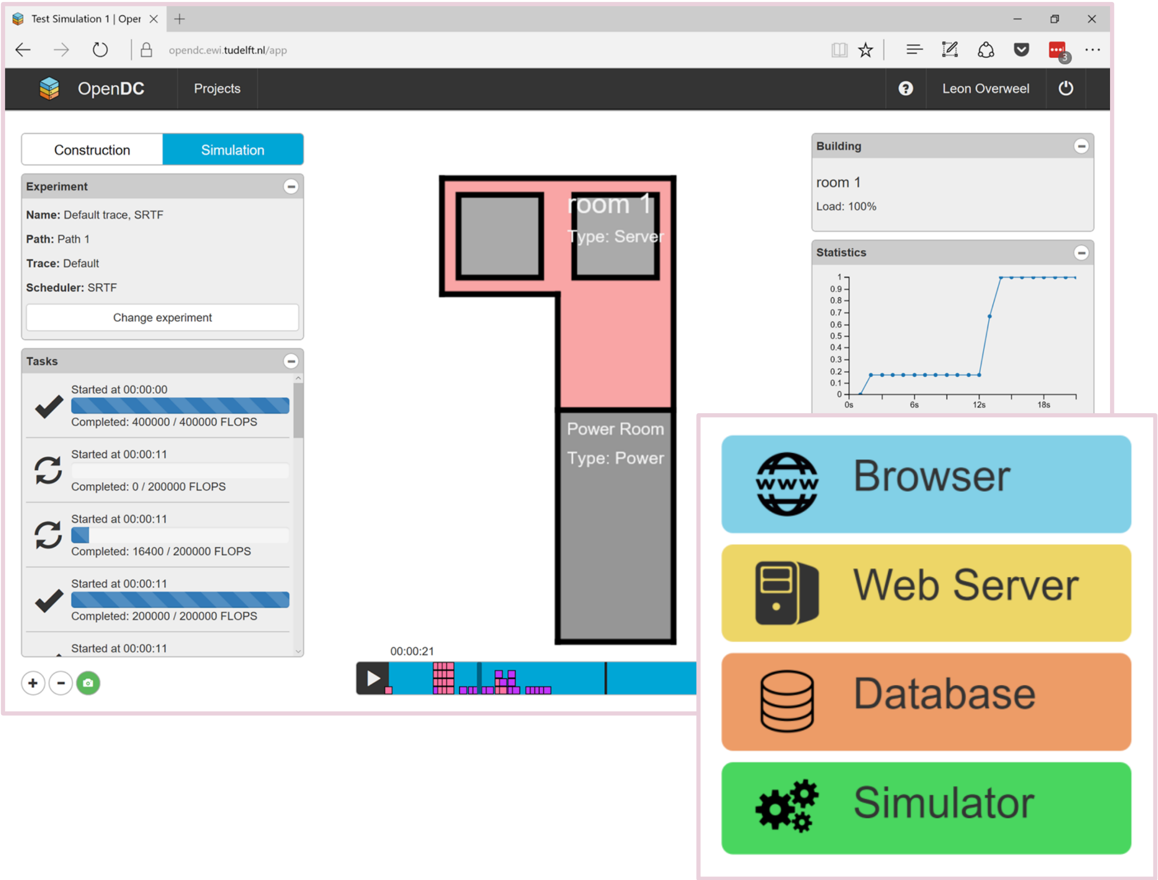
… and we **plan to use** it for:

Periodic **workshops** for **refugees** in the Netherlands with **Restart Network**

Promoting **science in schools** with the **Royal Netherlands Academy of Arts and Sciences**

Engaging **high school** students through **workshops** with the **Royal Dutch Engineers Society**

OpenDC 4. Software (and Data) Artifacts: see article



Current capabilities:

- Define dynamic DC **topologies**
- Run experiments on different **schedulers** and **workloads**
- Playback experimental results

Roadmap:

- **UI + API** for workloads + schedulers
- Componentized sim. for research

Availability:

- **Online** → Hosted by TU Delft
- **Locally** → Source on GitHub

@Large Research
Massivizing Computer Systems

VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft

OpenDC 2. Scientific Methods

How to conduct **scientific surveys** of **RM & Scheduling techniques** in DCs?

How to provide a **useful yet reduced** set of **metrics** for modern DC operation?

How to design a **deep yet practical** **methodological apparatus** for obtaining such metrics?

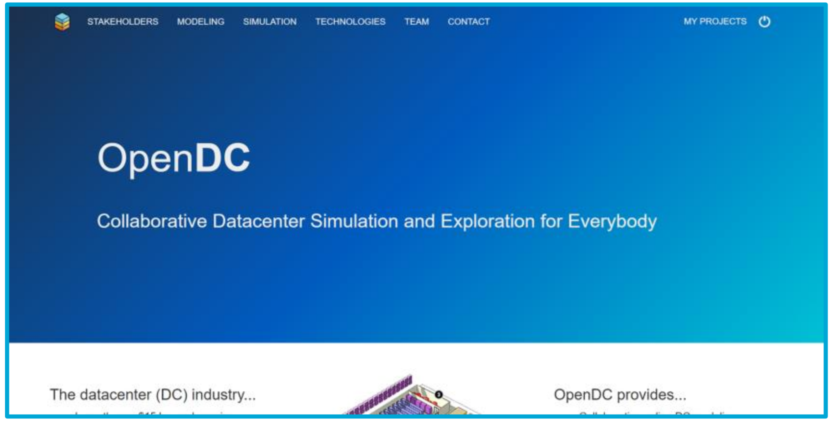
How to design a **reference architecture** for **DC stacks** / **cloud schedulers** /… ?

How do we conduct a **global scheduling competition**?

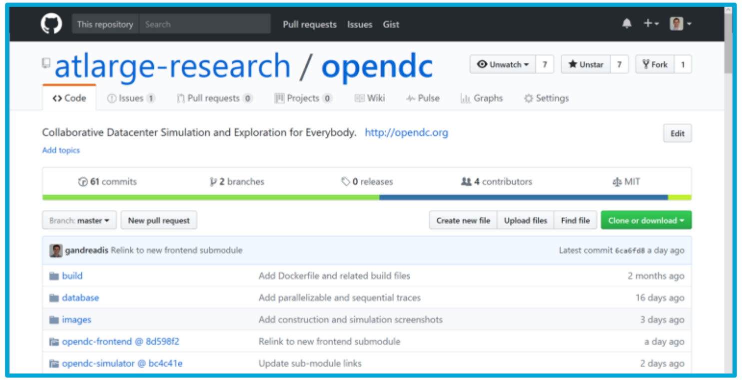
How to build **environments** where **reproducibility** is **ensured by the instrument**?

What is the **performance-validity trade-off** for datacenter simulation?

Find **OpenDC** online!



- opendc.org
- github.com/atlarge-research/opendc
- opendc@atlarge-research.com
- atlarge-research.com



spec RG Cloud

research.spec.org/working-groups/rg-cloud-working-group.html

@Large Research
Massivizing Computer Systems

VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft

Workload Modeling

(2006—2011) Grid workloads
(2011—ongoing) Cloud workloads
(2012—ongoing) Big Data workloads
(2015—ongoing) Business-critical workloads
(2009—ongoing) Online and social gaming workloads

Iosup, Epema. Grid Computing Workloads. IEEE Internet Computing 15(2): 19-26 (2011)

Iosup, Ostermann, Yigitbasi, Prodan, Fahringer, Epema. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. IEEE Trans. Parallel Distrib. Syst. 22(6): 931-945 (2011)

Hegeman, Ghit, Capota, Hidders, Epema, Iosup. The BTWorld use case for big data analytics: Description, MapReduce logical workflow, and empirical evaluation. BigData Conference 2013: 622-630

Shen, van Beek, Iosup. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. CCGRID 2015: 465-474

Jia, Shen, van de Bovenkamp, Iosup, Kuipers, Epema. Socializing by Gaming: Revealing Social Relationships in Multiplayer Online Games. TKDD 10(2): 11:1-11:29 (2015)

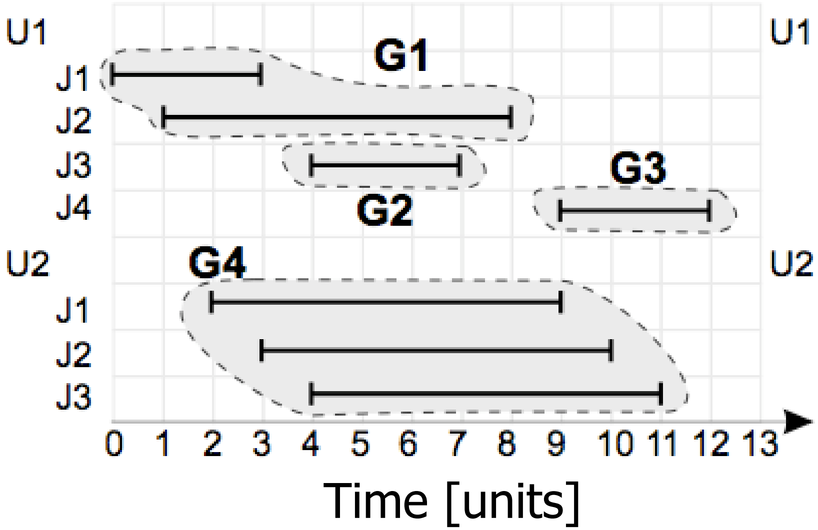
What is a Bag of Tasks (BoT)? A Systems View

BoT = set of jobs sent by a user…

$$W_u = \{J_i | user(J_i) = u\}$$

…that is submitted at most Δs after the first job

$$ST(J') \leq ST(J) + \Delta$$



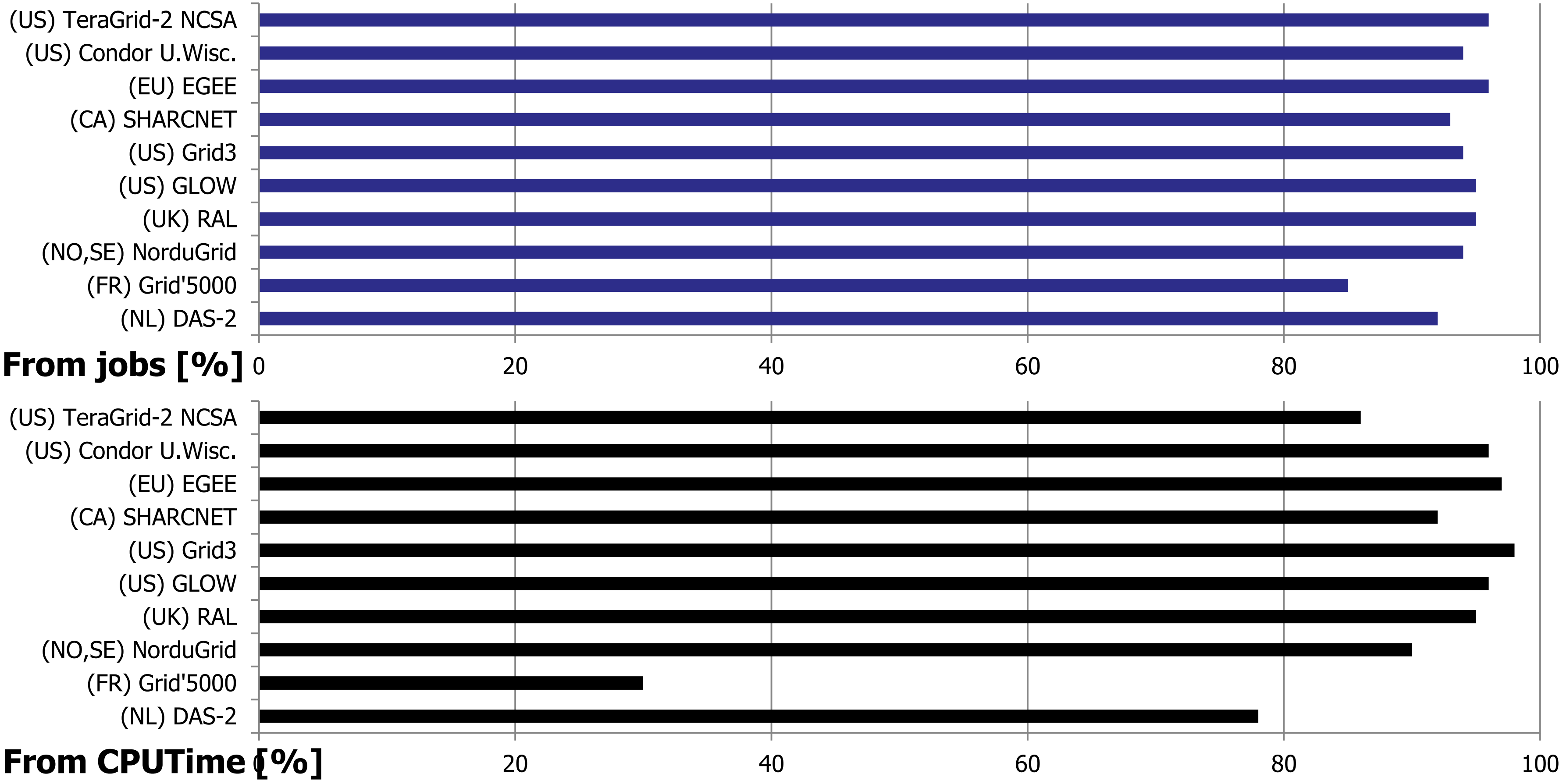
- Why Bag of *Tasks*? From the perspective of the user, jobs in set are just tasks of a larger job
- A single useful result from the complete BoT
- Result can be combination of all tasks, or a selection of the results of most or even a single task

Iosup et al., The Characteristics and Performance of Groups of Jobs in Grids, Euro-Par, LNCS, vol.4641, pp. 382-393, 2007.

Applications of the BoT Programming Model

- Parameter sweeps
  - Comprehensive, possibly exhaustive investigation of a model
  - Very useful in engineering and simulation-based science
- Monte Carlo simulations
  - Simulation with random elements: fixed time yet limited inaccuracy
  - Very useful in engineering and simulation-based science
- Many other types of batch processing
  - Periodic computation, Cycle scavenging
  - Very useful to automate operations and reduce waste

BoTs Are the Dominant Programming Model for Grid Computing (Many Tasks)

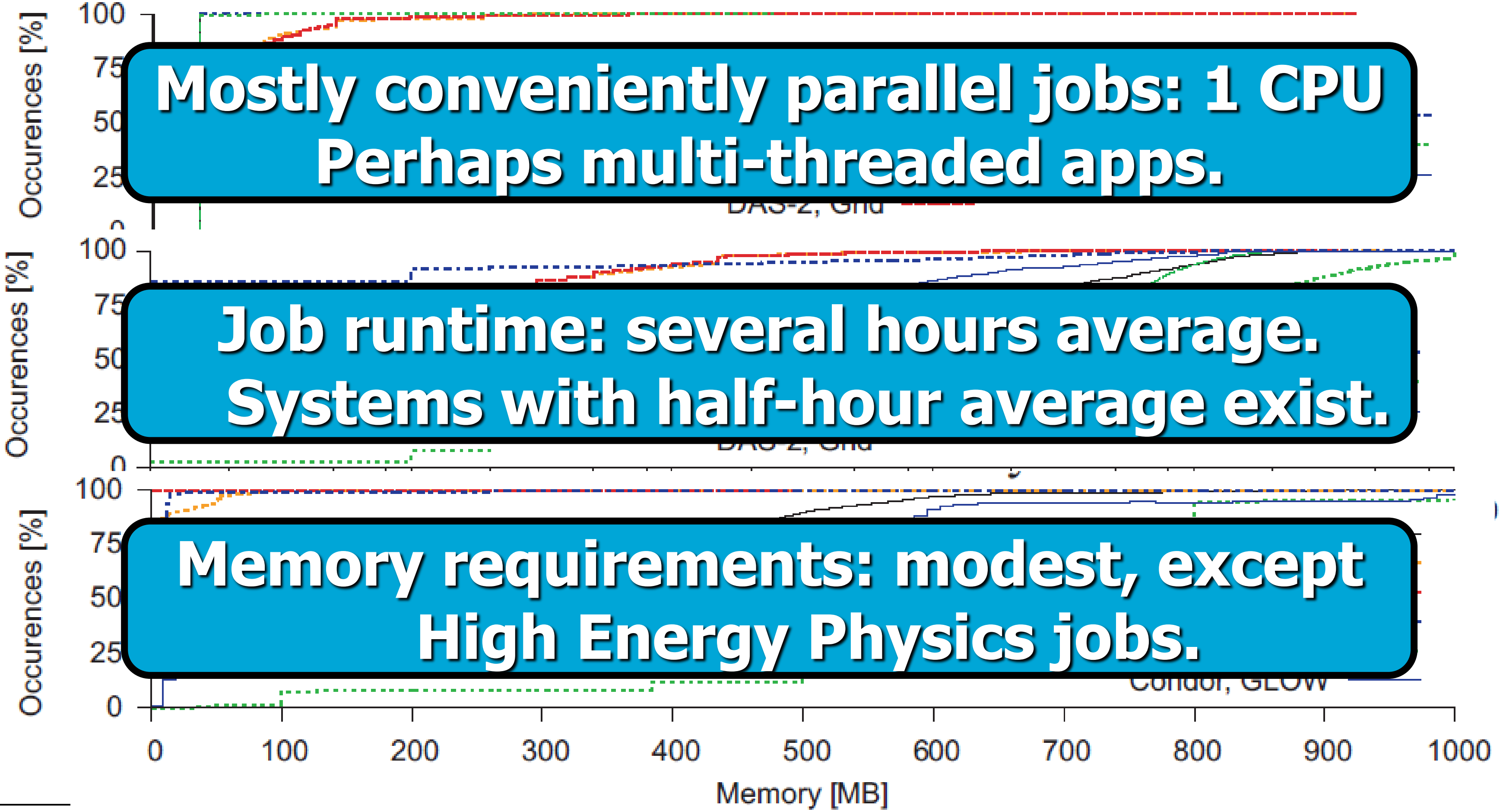


Iosup and Epema: Grid Computing Workloads.
IEEE Internet Computing 15(2): 19-26 (2011)

Q0

TU Delft
Delft University of Technology

BoTs by Numbers: CPUs, Runtime, Mem



Mostly conveniently parallel jobs: 1 CPU
Perhaps multi-threaded apps.

Job runtime: several hours average.
Systems with half-hour average exist.

Memory requirements: modest, except
High Energy Physics jobs.

Iosup et al., The Grid Workloads Archive, FGCS, 2008.

Iosup and Epema, Grid Computing Workloads, IEEE Internet Computing, 2011.

Actual numbers.

BoTs by numbers: I/O, Files, Remote Sys

T-12 part	I/O [KOps] Total	Rd	Wr	Wr %	I/O Traffic [MB] Total	Rd	Wr %
[illegible]	[illegible]	[illegible]	[illegible]	20%	469	174	63%
[illegible]	[illegible]	[illegible]	[illegible]	20%	144	114	21%
[illegible]	[illegible]	[illegible]	[illegible]	3%	161	130	19%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	389	33	92%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	330	31	91%

I/O: modest, except HEP

Rd:Wr varies widely

I/O,HEP: 65MBps/experiment

Upper bound for typical sci.apps.

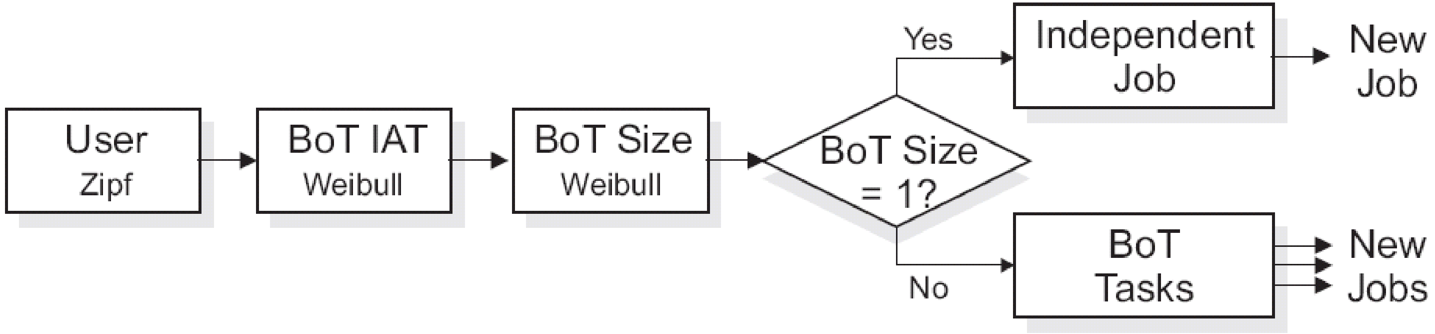
T-12 part	File Transfer [MB] Total	In	In / Out %		Remote Sys. Calls [MB] Total	In	In / Out %	
t1	10,865	8,259	76%	24%	28	16	59%	41%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	60%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	58%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0%
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	44	40	91%	9%

Remote Sys.: small Xfers, latency important

Netw: 2-10GB, input mostly

Iosup and Epema, Grid Computing Workloads, IEEE Internet Computing, 2011.

BoT Workload Model

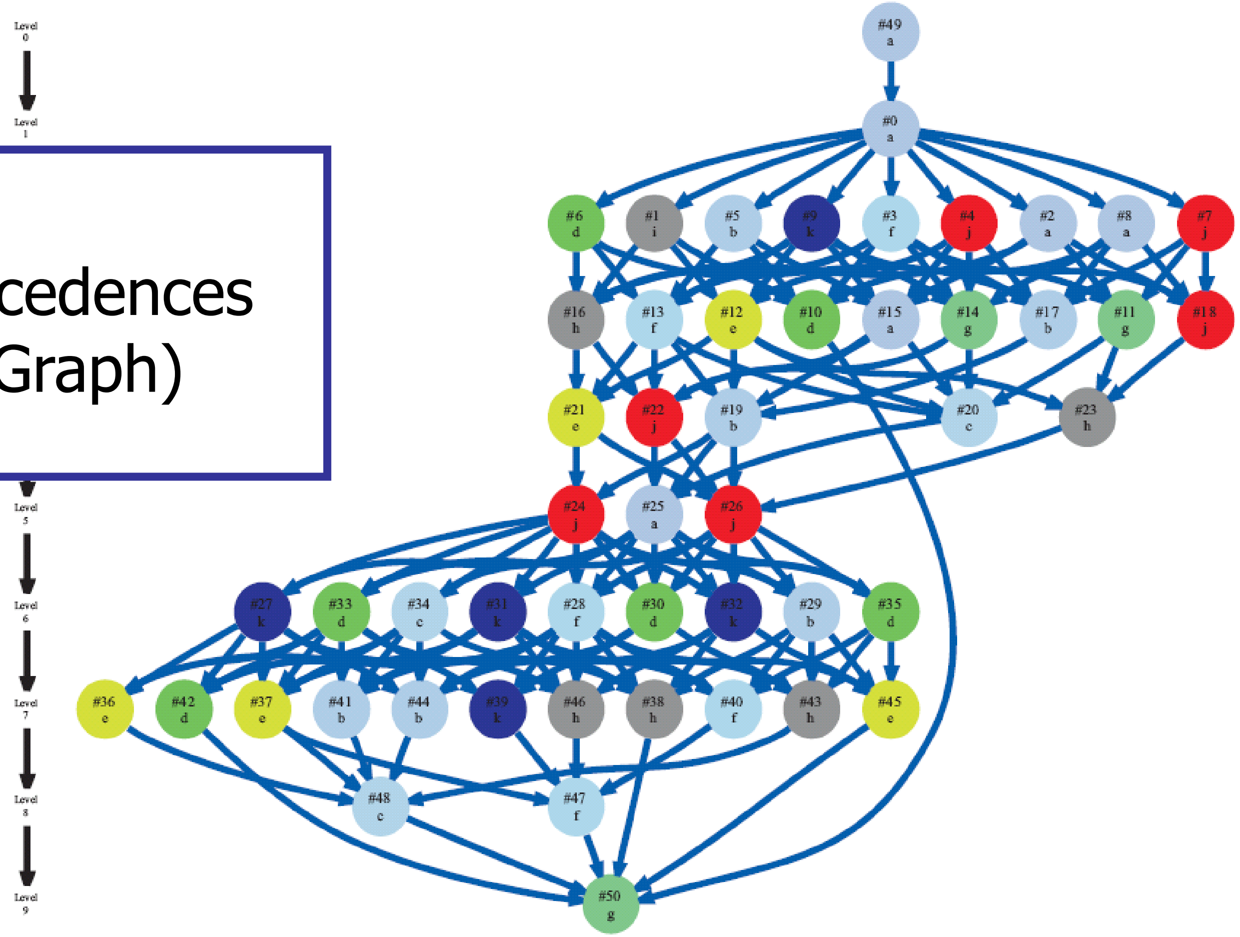


- Single arrival process for both BoTs and parallel jobs
- Validated with 7 grid workloads

A. Iosup, O. Sonmez, S. Anoep, and D.H.J. Epema. The Performance of Bags-of-Tasks in Large-Scale Distributed Systems, HPDC, pp. 97-108, 2008.

What is a Wokflow?

WF = set of jobs with precedences
(think Direct Acyclic Graph)



Applications of the Workflow Programming Model

- Complex applications
  - Complex filtering of data
  - Complex analysis of instrument measurements
- Applications created by non-CS scientists*
  - Workflows have a natural correspondence in the real-world, as descriptions of a scientific procedure
  - Visual model of a graph sometimes easier to program
- Precursor of the MapReduce Programming Model (next slides)

*Adapted from: Carole Goble and David de Roure, Chapter in "The Fourth Paradigm", http://research.microsoft.com/en-us/collaboration/fourthparadigm/

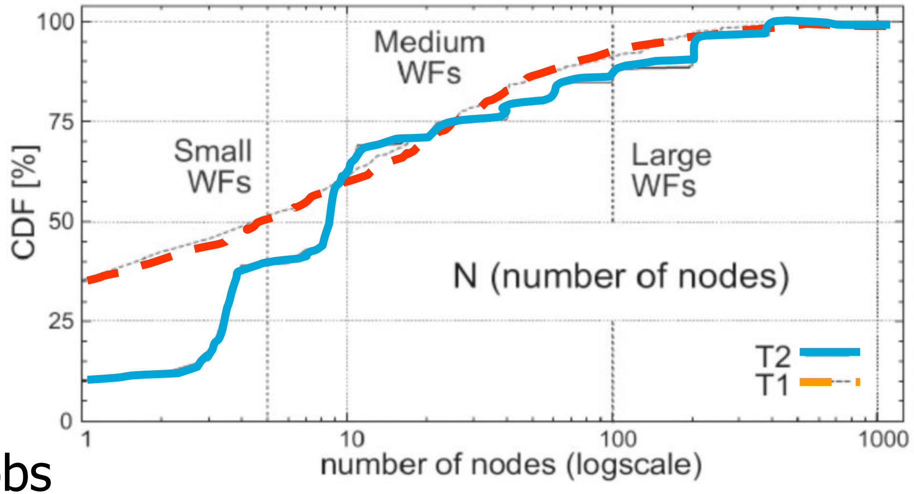
Q0

Workflows Exist in Grids, but Did No Evidence of a Dominant Programming Model

- Traces

Trace	Source	Duration	Number of WFs	Number of Tasks	CPUdays
T1	DEE	09/06-10/07	4,113	122k	152
T2	EE2	05/07-11/07	1,030	46k	41

- Selected Findings



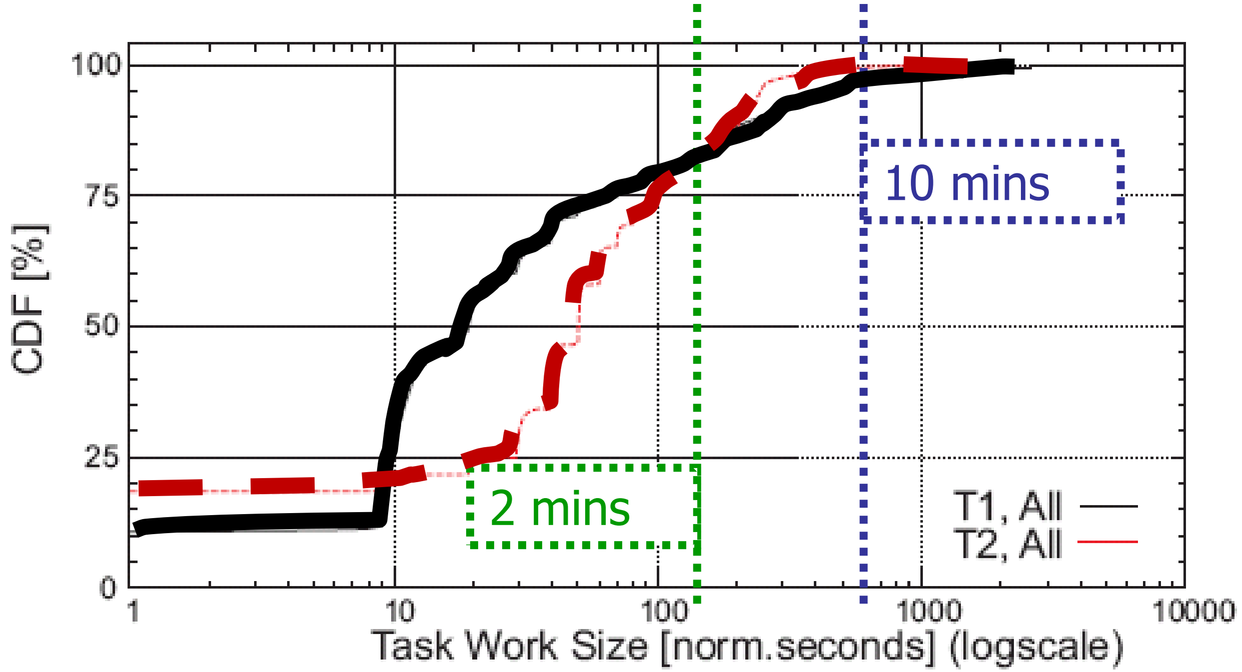
  - Loose coupling
  - Graph with 3-4 levels
  - Average WF size is 30/44 jobs
  - 75%+ WFs are sized 40 jobs or less, 95% are sized 200 jobs or less

Ostermann et al., On the Characteristics of Grid Workflows, CoreGRID Integrated Research in Grid Computing (CGIW), 2008.

Delft University of Technology

Q0

Workflows: Intrinsic Characteristics
Task Work Size



- >80% WFs take <2 minutes on 1000-SI2k machine
- >95% WFs take <10 minutes on 1000-SI2k machine

Ostermann et al., On the Characteristics of Grid Workflows, CoreGRID Integrated Research in Grid Computing (CGIW), 2008.

Analysis of MapReduce Workloads
Workload Characteristics at Google, Yahoo, etc.

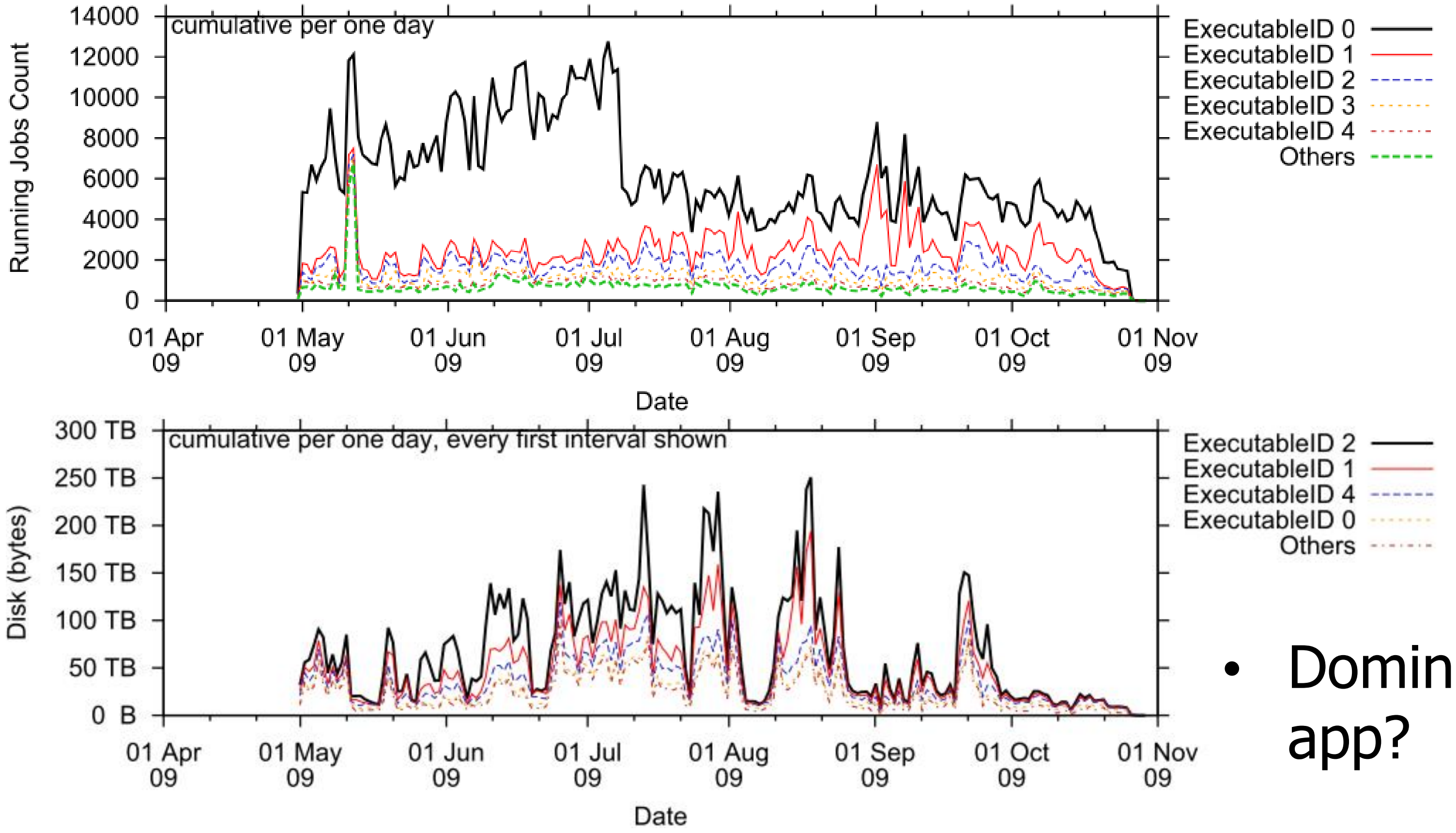
Workload	Period	Task Information		Failed Jobs	MapReduce Only	Number Of	
		Aggregated per Job	For Each Task			Jobs	Tasks
SN1	6 months	+	–	–	+	1,129,193	?
SN2	9 days	+	–	+	+	60,978	9,365,863
Yahoo! M	2 weeks	+	+	+	+	28,248	27,317,243
Google	29 days	+	+	+	–	667,992	44,920,671

- Analysis of job/task characteristics
- Identification of applications
- (also modeling)

Th. De Ruiter, A. Iosup. A workload model for MapReduce. MSc Thesis. 2012. http://repository.tudelft.nl/view/ir/uuid:1647e1cb-84fd-46ca-b1e1-21aaf38ef30b/

Analysis of MapReduce Workloads
Workload Characteristics at Google, Yahoo, etc.

SN1



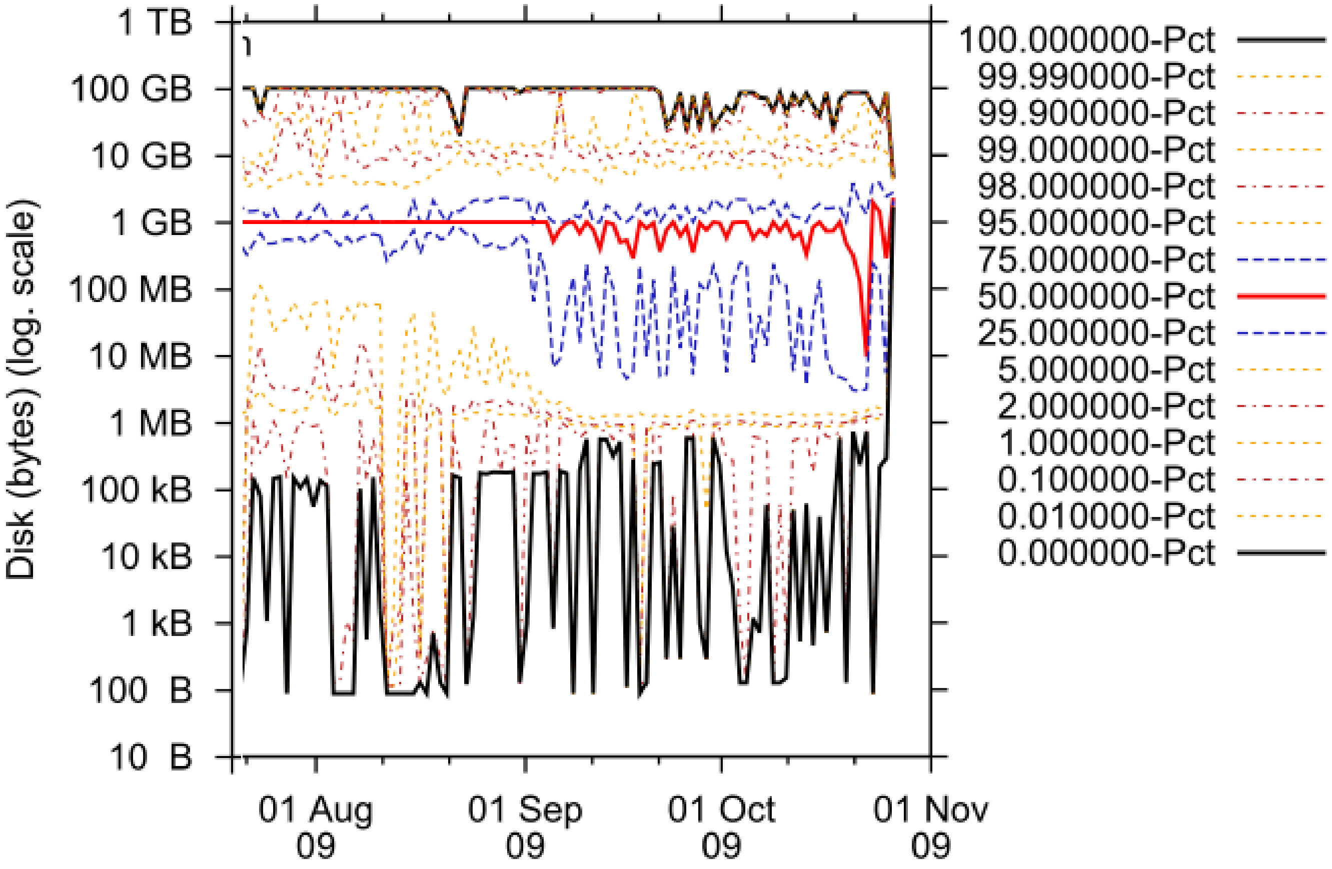
- Dominant app?

Th. De Ruiter, A. Iosup. A workload model for MapReduce. MSc Thesis. 2012. http://repository.tudelft.nl/view/ir/uuid:1647e1cb-84fd-46ca-b1e1-21aaf38ef30b/

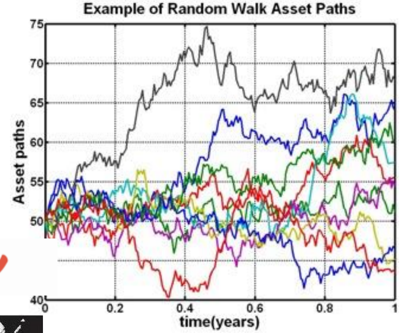
TU Delft
Delft University of Technology

Analysis of MapReduce Workloads
Workload Characteristics at Google, Yahoo, etc.

- SN1
- Variability?



Th. De Ruiter, A. Iosup. A workload model for MapReduce. MSc Thesis. 2012. http://repository.tudelft.nl/view/ir/uuid:1647e1cb-84fd-46ca-b1e1-21aaf38ef30b/



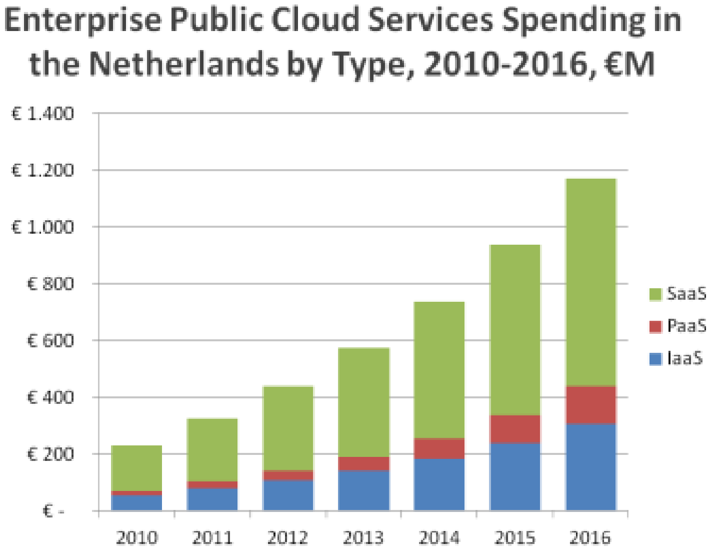
Monte Carlo simulation

TOWERS WATSON

Algorithmics

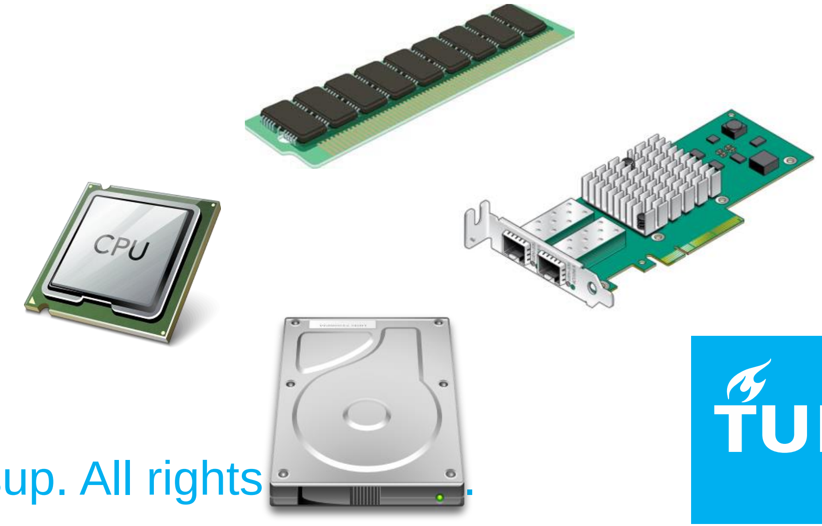
ORACLE

Apache



Source: http://www.themetisfiles.com

Business Critical Workloads

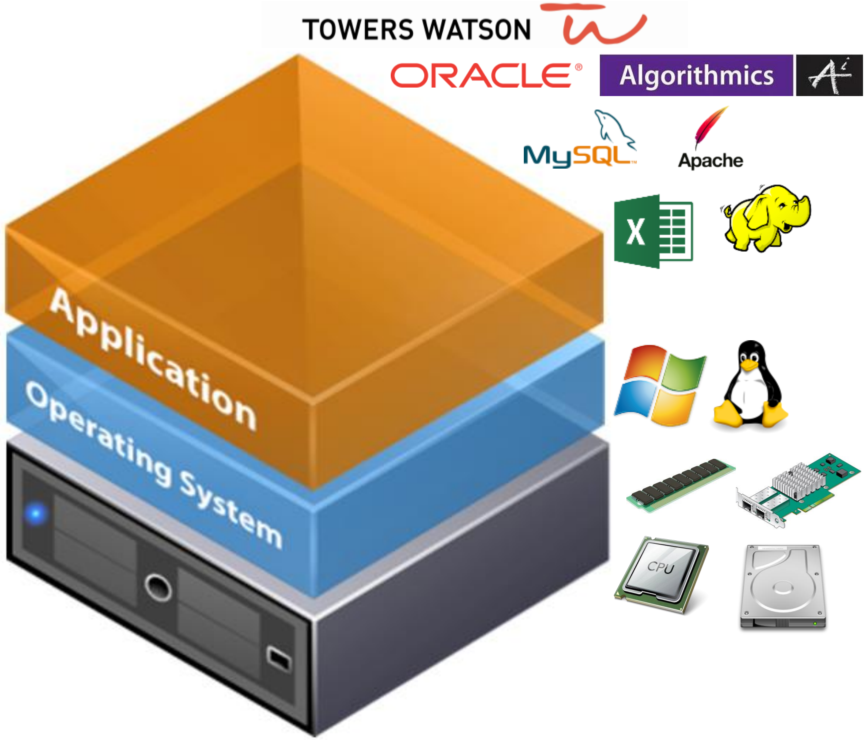


VU

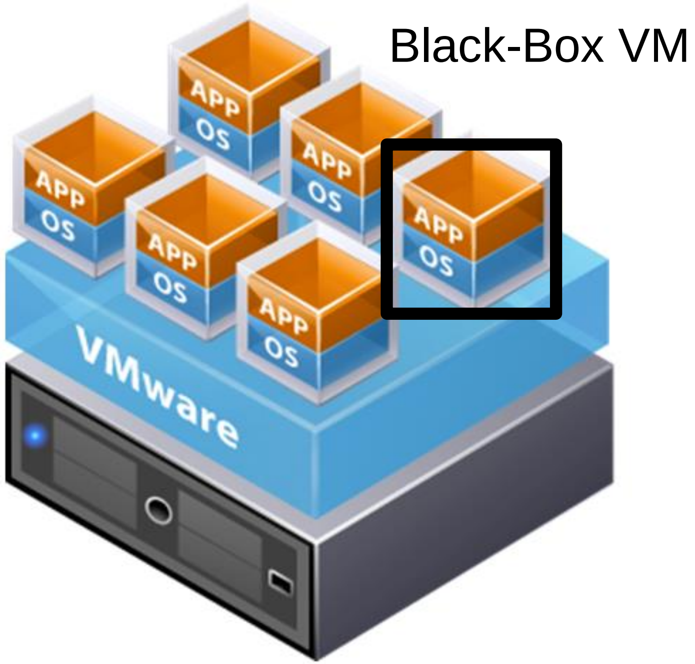
TU Delft

© 2017 Alexandru Iosup. All rights

What Changed for Cloud-Hosted Workloads?



Traditional Architecture



Virtual Architecture

VU VRIJE UNIVERSITEIT AMSTERDAM

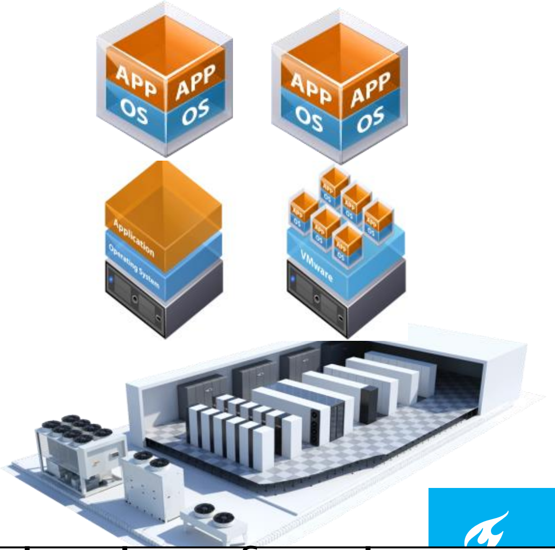
© 2017 Alexandru Iosup. All rights reserved.

TUDelft

Collected Two Unique Workload Traces

Name of the trace	# VMs	Period of data collection	Storage technology	Total memory	Total cores
fastStorage	1,250	1 month	SAN	17,729 GB	4,057
Rnd	500	3 months	NAS and SAN	5,485 GB	1,444
Total	1,750	5,446,811 CPU hours		23,214 GB	5,501

- All resources:
  - CPU, Memory, Storage, and Network
- Large scale
- Long term



VU VRIJE UNIVERSITEIT AMSTERDAM

S. Shen et al. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. CCGRID 2015: 465-474

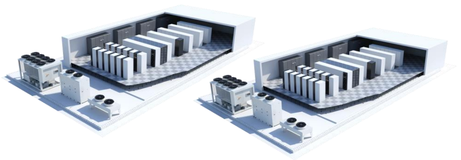
Conducted Unique Workload Analysis

Prior work:
- Google
- Facebook
- Taobao
- Scientific workloads
- Grids vs Google

First study of both:

- Requested and

- Used resources



- For all resources

S. Shen et al. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. CCGRID 2015: 465-474

Our findings: Business-Critical vs. Known workloads

- Long running VMs vs short running jobs
- Compared to parallel workloads, small in size (cpu and memory)
  - Many opportunities for scheduling efficiency (e.g., used<<requested, pow-2, periodicity)
- Much more diverse in nature, compared to
  data analysis workloads from Facebook, Google, and Tabao
  - Monte Carlo Simulation (e.g., finance)
  - Data analysis of business data (e.g., finance)
  - Office automation (e.g., web, mail)
  - High available web-services for complex applications (e.g., retail, CC systems)
  - DC value-adding services, e.g., backup

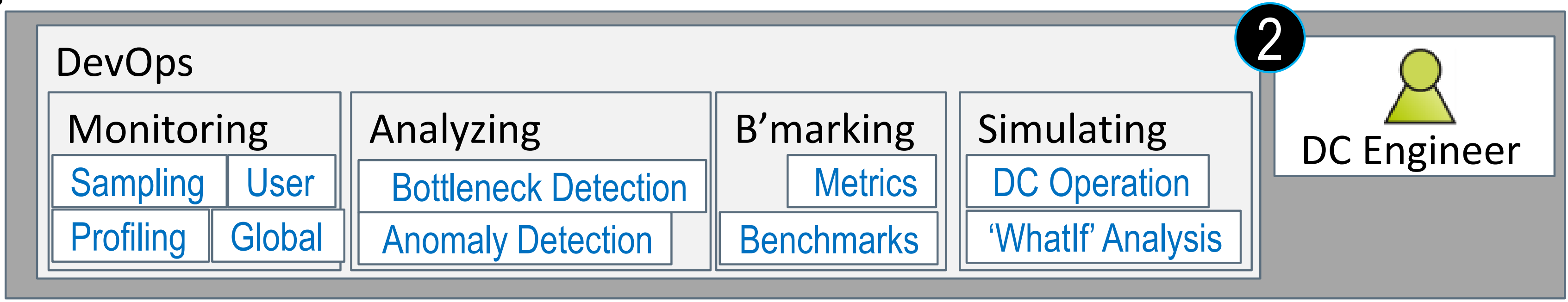
VRIJE UNIVERSITEIT AMSTERDAM

S. Shen et al. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. CCGRID 2015: 465-474

2. DevOps

Knowledge / Software tools

- Monitoring
  - Largest measurements of BitTorrent (2005, 2010)
  - DC measurements (2006—ongoing)
  - Large-scale cloud observation (2008—ongoing)
  - Availability and performance in DCs (2008—ongoing)
  - Granula
- Analyzing
  - Bottleneck and performance anomaly detection for big data
  - Non-stationary systems
  - Bursty workloads
  - Structured workloads
  - Grade10, Granula
- Benchmarking
  - GrenchMark & C-Meter
  - LDBC Graphalytics
- Simulating
  - Portfolio-scheduling simulation
  - Simulating grids, p2p
  - Simulating DCs
  - DGSim & OpenDC



Alexandru Iosup
Chair

Nikolas Herbst
Vice-Chair

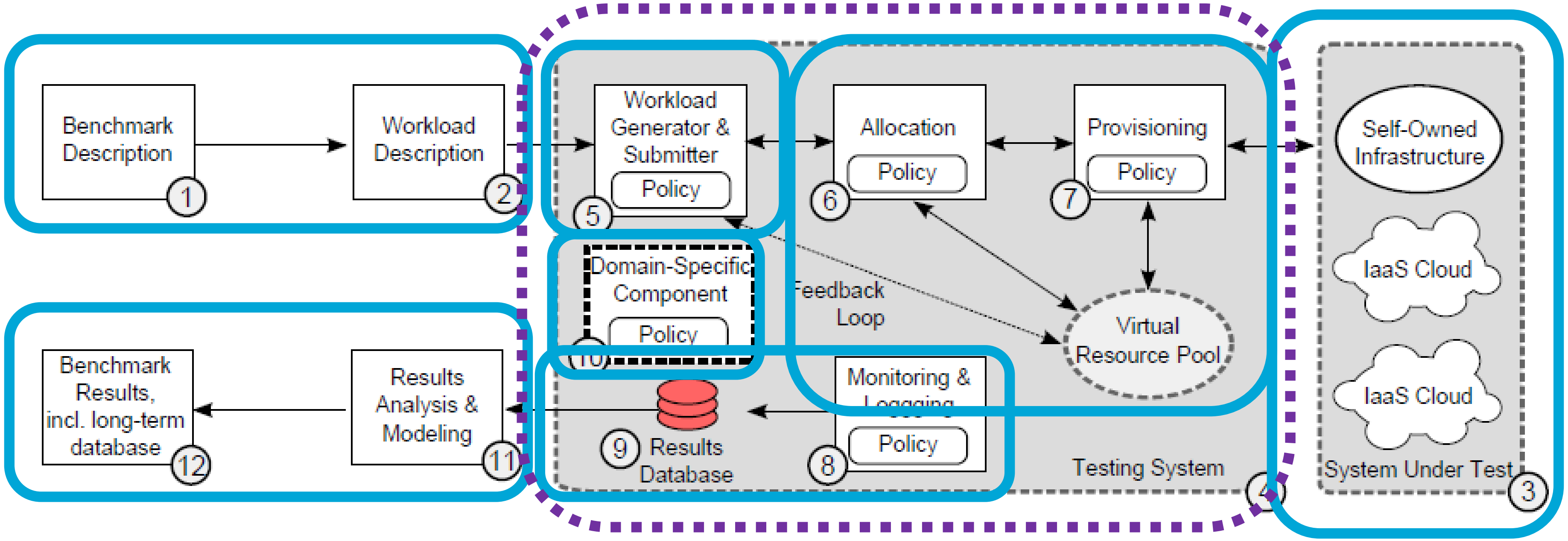
The SPEC RG Cloud Group

Methodology, Benchmarking, and Performance Analysis of Cloud Systems and Applications

"A broad approach, **relevant for both academia and industry**, to cloud benchmarking, quantitative evaluation, and experimental analysis."
"To develop new **methodological elements** for gaining deeper understanding not only of **cloud performance**, but also of **cloud operation and behavior"**
"… through diverse quantitative evaluation tools"

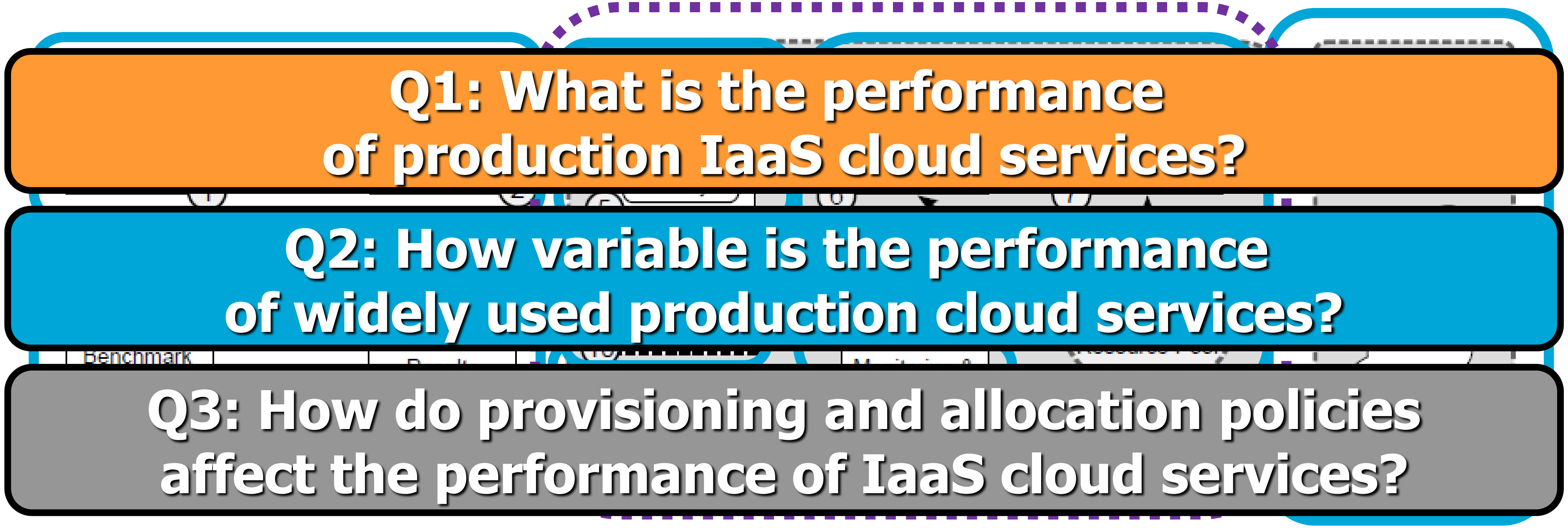
http://research.spec.org/working-groups/rg-cloud-working-group.html

A General Approach for IaaS Cloud Benchmarking



Iosup, Prodan, Epema. IaaS Cloud Benchmarking: Approaches, Challenges, and Experience. Cloud Computing for Data-Intensive Applications 2014: 83-104

A General Approach for IaaS Cloud Benchmarking



Q1: What is the performance of production IaaS cloud services?

Q2: How variable is the performance of widely used production cloud services?

Q3: How do provisioning and allocation policies affect the performance of IaaS cloud services?

Iosup, Prodan, Epema. IaaS Cloud Benchmarking: Approaches, Challenges, and Experience. Cloud Computing for Data-Intensive Applications 2014: 83-104

10 Main Challenges in 4 Categories*

* The future

* List not exhaustive

- **Methodological**
  1. Experiment compression
  2. Beyond black-box testing through testing short-term dynamics and long-term evolution
  3. Impact of middleware
- **Workload-related**
  1. Statistical workload models
  2. Benchmarking performance isolation under various multi-tenancy workloads
- **System-Related**
  1. Reliability, availability, and system-related properties
  2. Massive-scale, multi-site benchmarking
  3. Performance isolation, multi-tenancy models
- **Metric-Related**
  1. Beyond traditional performance: variability, elasticity, etc.
  2. Closer integration with cost models

Iosup, Prodan, Epema. IaaS Cloud Benchmarking: Approaches, Challenges, and Experience. Cloud Computing for Data-Intensive Applications 2014: 83-104

Some Previous Work (>50 important references across our studies)

Virtualization Overhead

- Loss below 5% for computation [Barham03] [Clark04]
- Loss below 15% for networking [Barham03] [Menon05]
- Loss below 30% for parallel I/O [Vetter08]
- Negligible for compute-intensive HPC kernels [You06] [Panda06]

Cloud Performance Evaluation

- Performance and cost of executing a sci. workflows [Dee08]
- Study of Amazon S3 [Palankar08]
- Amazon EC2 for the NPB benchmark suite [Walker08] or selected HPC benchmarks [Hill08]
- CloudCmp [Li10]
- Kosmann et al.

© 2017 Alexandru Iosup. All rights reserved.

Q1

Production IaaS Cloud Services in 2007-2008

- **Production IaaS cloud:** lease resources (infrastructure) to users, operate on the market and have active customers

Name	Cores (ECUs)	RAM [GB]	Archi. [bit]	Disk [GB]	Cost [$/h]
Amazon EC2					
m1.small	1 (1)	1.7	32	160	0.1
m1.large	2 (4)	7.5	64	850	0.4
m1.xlarge	4 (8)	15.0	64	1,690	0.8
c1.medium	2 (5)	1.7	32	350	0.2
c1.xlarge	8 (20)	7.0	64	1,690	0.8
GoGrid (GG)					
GG.small	1	1.0	32	60	0.19
GG.large	1	1.0	64	60	0.19
GG.xlarge	3	4.0	64	240	0.76
Elastic Hosts (EH)					
EH.small	1	1.0	32	30	£0.042
EH.large	1	4.0	64	30	£0.09
Mosso					
Mosso.small	4	1.0	64	40	0.06
Mosso.large	4	4.0	64	160	0.24

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Q1

Our Method

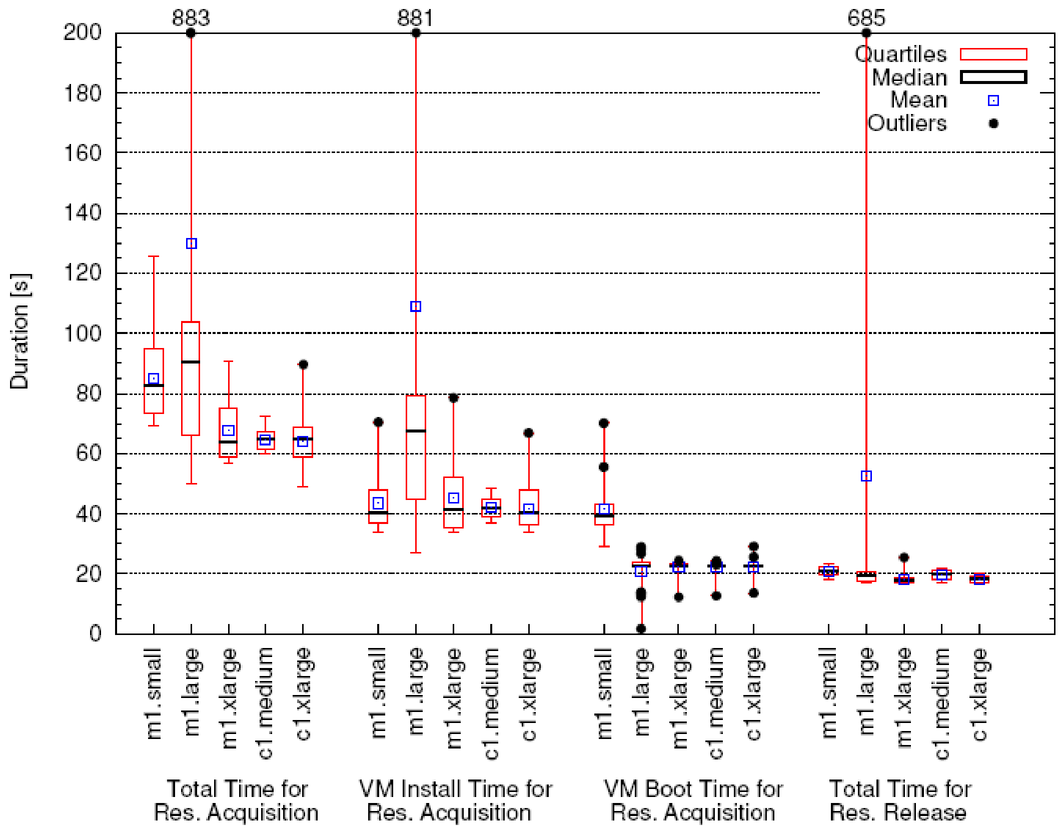
- Based on general performance technique: model performance of individual components; system performance is performance of workload + model [Saavedra and Smith, ACM TOCS'96]
- Adapt to clouds:
  1. Cloud-specific elements: resource provisioning and allocation
  2. Benchmarks for single- and multi-machine jobs
  3. Benchmark CPU, memory, I/O, etc.:

Type	Suite/Benchmark	Resource	Unit
SI	LMbench/all [24]	Many	Many
SI	Bonnie/all [25], [26]	Disk	MBps
SI	CacheBench/all [27]	Memory	MBps
MI	HPCC/HPL [28], [29]	CPU	GFLOPS
MI	HPCC/DGEMM [30]	CPU	GFLOPS
MI	HPCC/STREAM [30]	Memory	GBps
MI	HPCC/RandomAccess [31]	Network	MUPS
MI	HPCC/b_{eff}(lat.,bw.) [32]	Comm.	μs, GBps

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Q1

Single Resource Provisioning/Release



- Time depends on instance type
- Boot time non-negligible

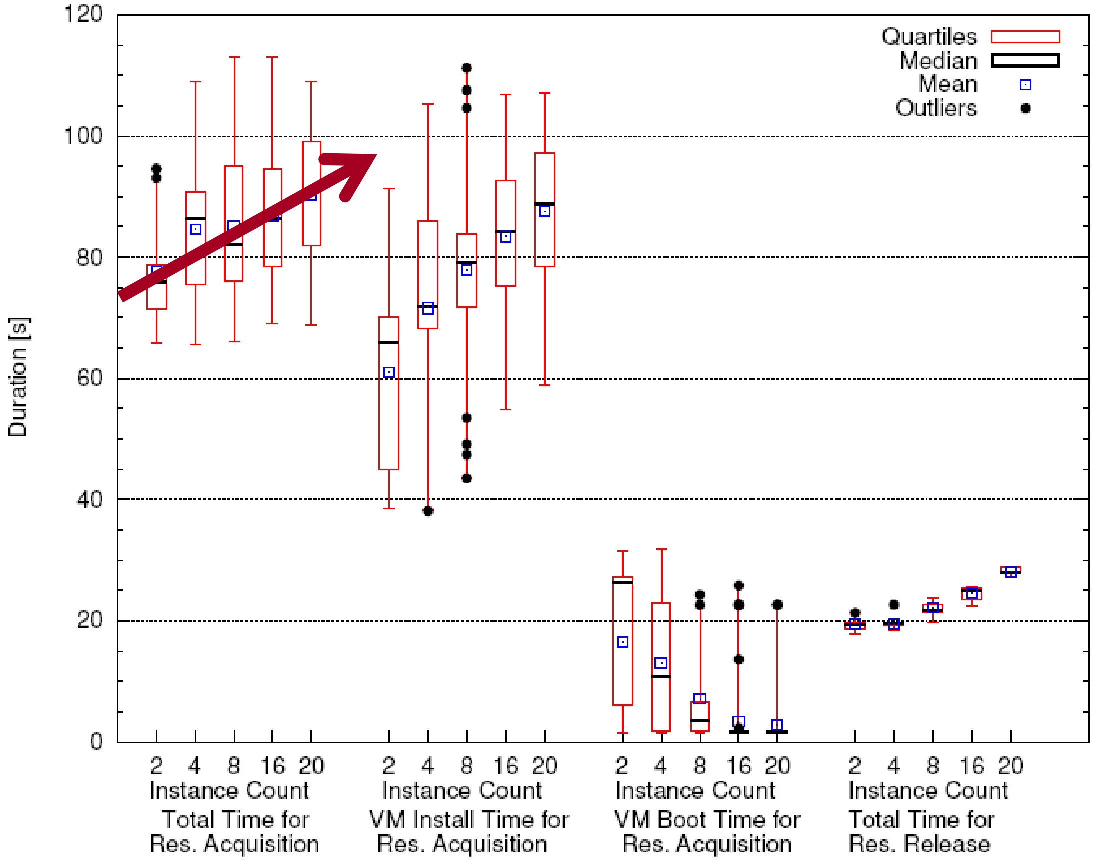
VRIJE UNIVERSITEIT AMSTERDAM

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

TU Delft

Q1

Multi-Resource Provisioning/Release



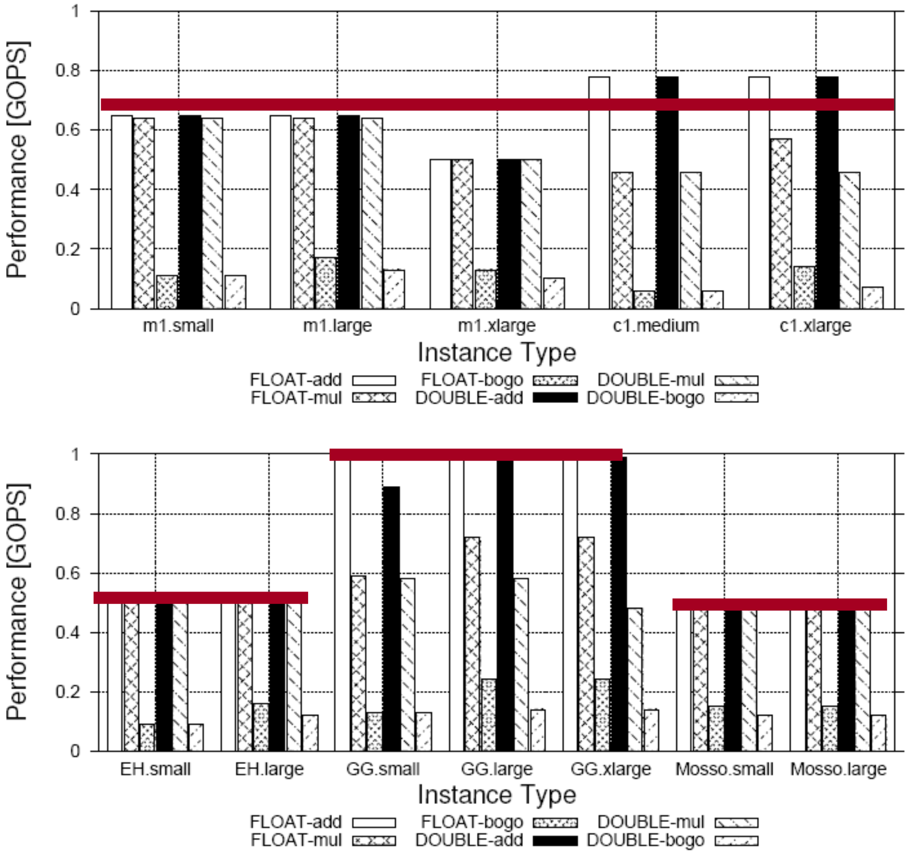
- Time for ***multi-*** resource increases with number of resources

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Q1

CPU Performance of Single Resource

- ECU definition: “a 1.1 GHz 2007 Opteron” ~ 4 flops per cycle at full pipeline, which means at peak performance one ECU equals 4.4 gigaflops per second (GFLOPS)
- Real performance 0.6..0.1 GFLOPS = ~1/4..1/7 theoretical peak



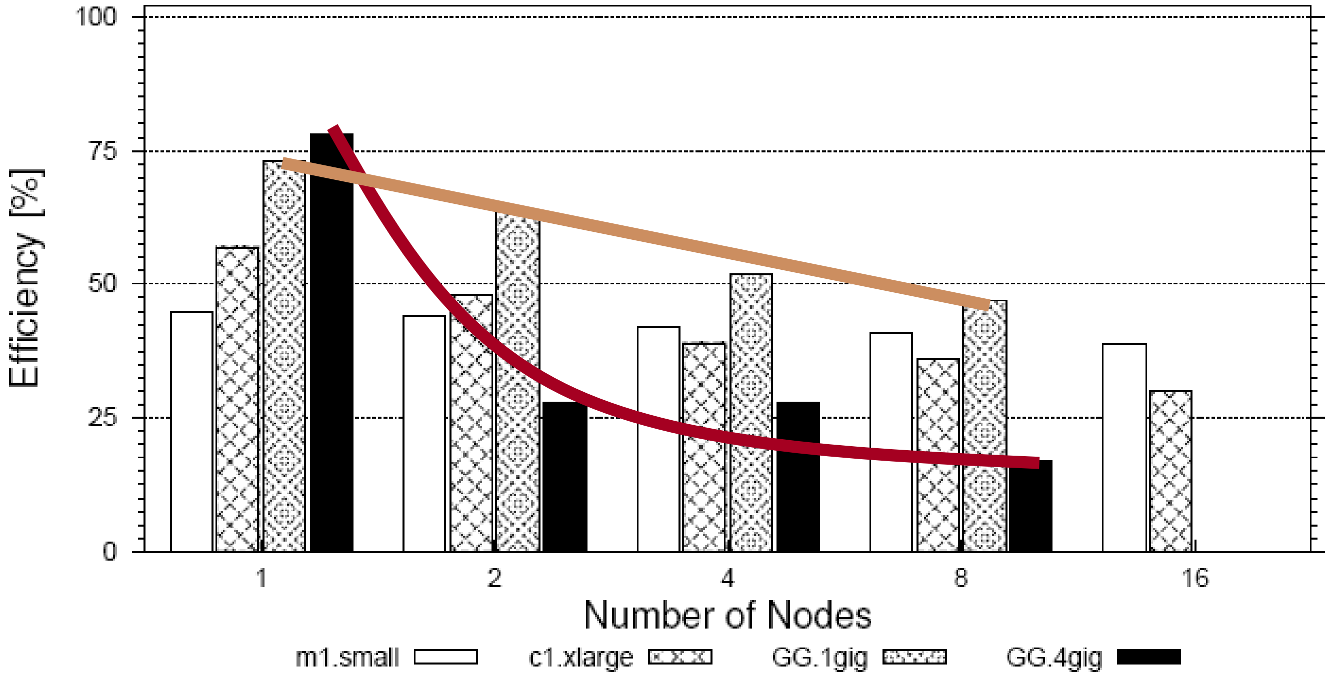
Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft

Q1

HPLinpack Performance (Parallel)



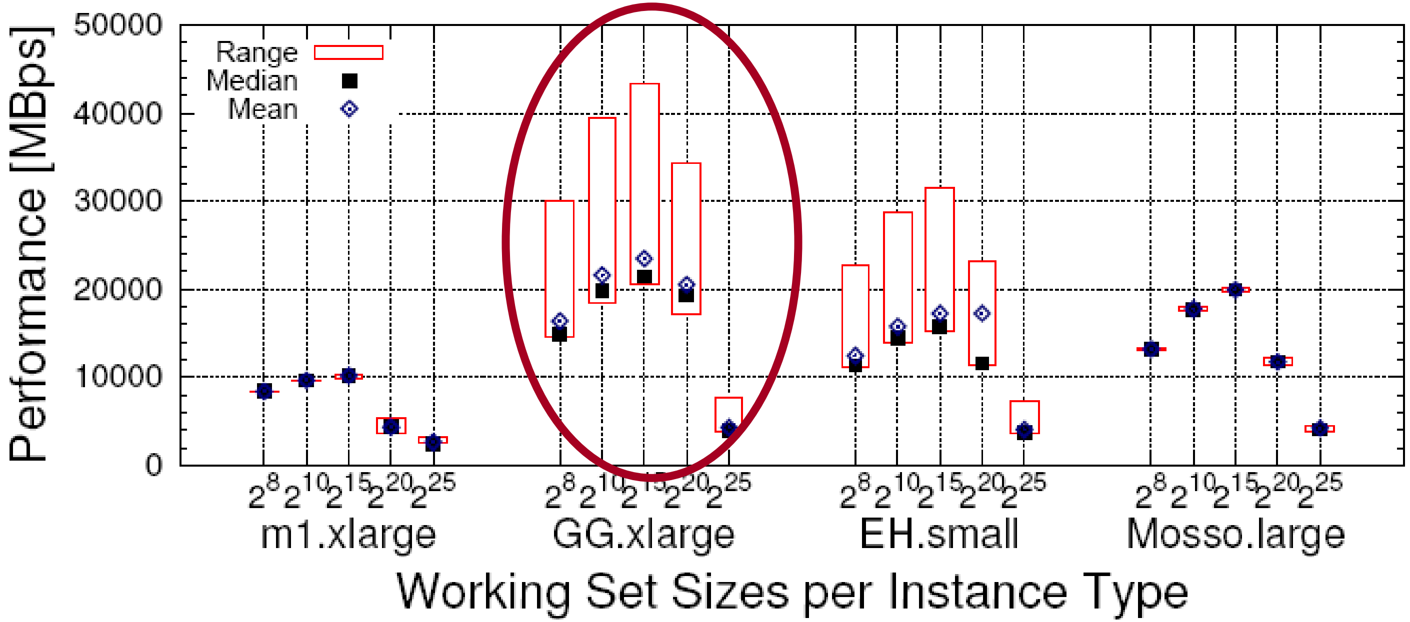
- Low efficiency for parallel compute-intensive applications
- Low performance vs cluster computing and supercomputing

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Q1

Q2

Performance Stability (Variability)



- Performance variability is high for the best-performing instances

Iosup et al., Performance Analysis of Cloud Computing Services for Many Tasks Scientific Computing, (IEEE TPDS 2011).

Q2

Production Cloud Services

- **Production cloud:** operate on the market and have active customers

  - **IaaS/PaaS: Amazon Web Services (AWS)**
    - EC2 (Elastic Compute Cloud)
    - S3 (Simple Storage Service)
    - SQS (Simple Queueing Service)
    - SDB (Simple Database)
    - FPS (Flexible Payment Service)
  - **PaaS: Google App Engine (GAE)**
    - Run (Python/Java runtime)
    - Datastore (Database) ~ SDB
    - Memcache (Caching)
    - URL Fetch (Web crawling)

Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

Q2

Our Method [1/3]
Performance Traces

- CloudStatus*
  - Real-time values and weekly averages for most of the AWS and GAE services

- Periodic performance probes
  - Sampling rate is under 2 minutes

* www.cloudstatus.com

VU VRIJE UNIVERSITEIT AMSTERDAM

Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

TU Delft

Our Method [2/3]
Analysis

1. Find out whether variability is present
   - Investigate several months whether the performance metric is highly variable
2. Find out the characteristics of variability
   - Basic statistics: the five quartiles (Q_0-Q_4) including median (Q_2), mean, std.deviation
   - Derivative statistic: the IQR (Q_3-Q_1)
   - CoV > 1.1 indicate high variability
3. Analyze the performance variability time patterns
   - Investigate for each performance metric presence of daily/monthly/weekly/yearly time patterns
   - E.g., for monthly patterns divide the dataset into twelve subsets and for each subset compute the statistics and plot for visual inspection

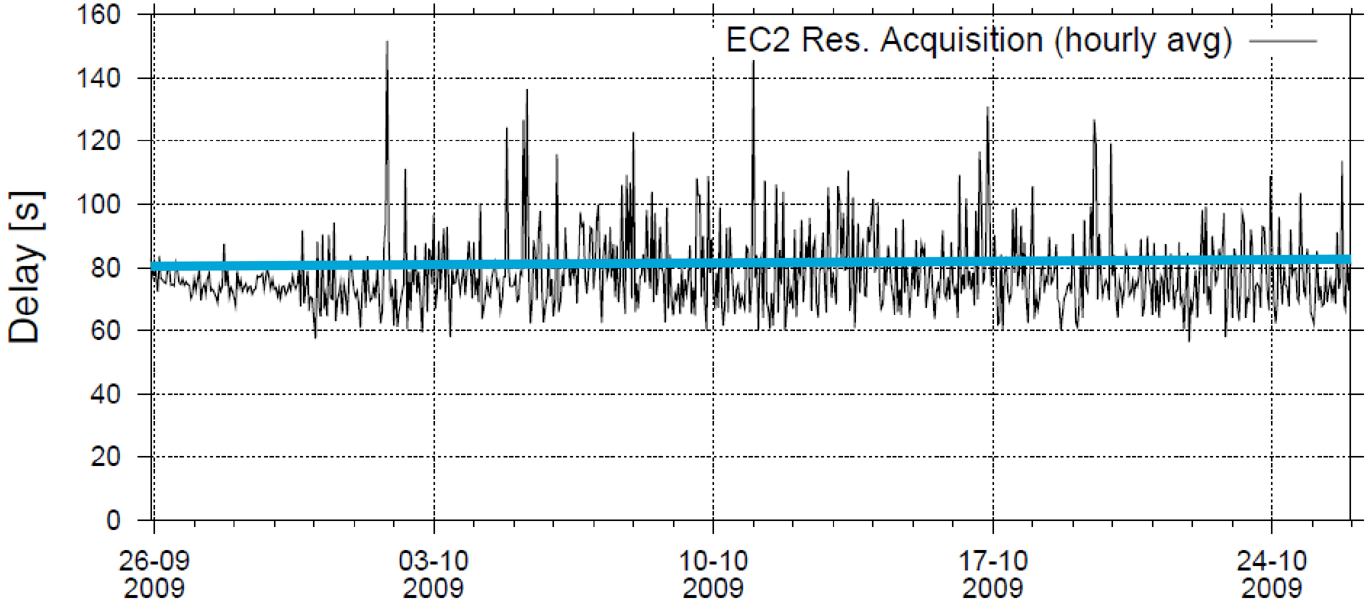
VU VRIJE UNIVERSITEIT AMSTERDAM

Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

TU Delft

Our Method [3/3]
Is Variability Present?

- **Validated Assumption:** The performance delivered by production services is variable.

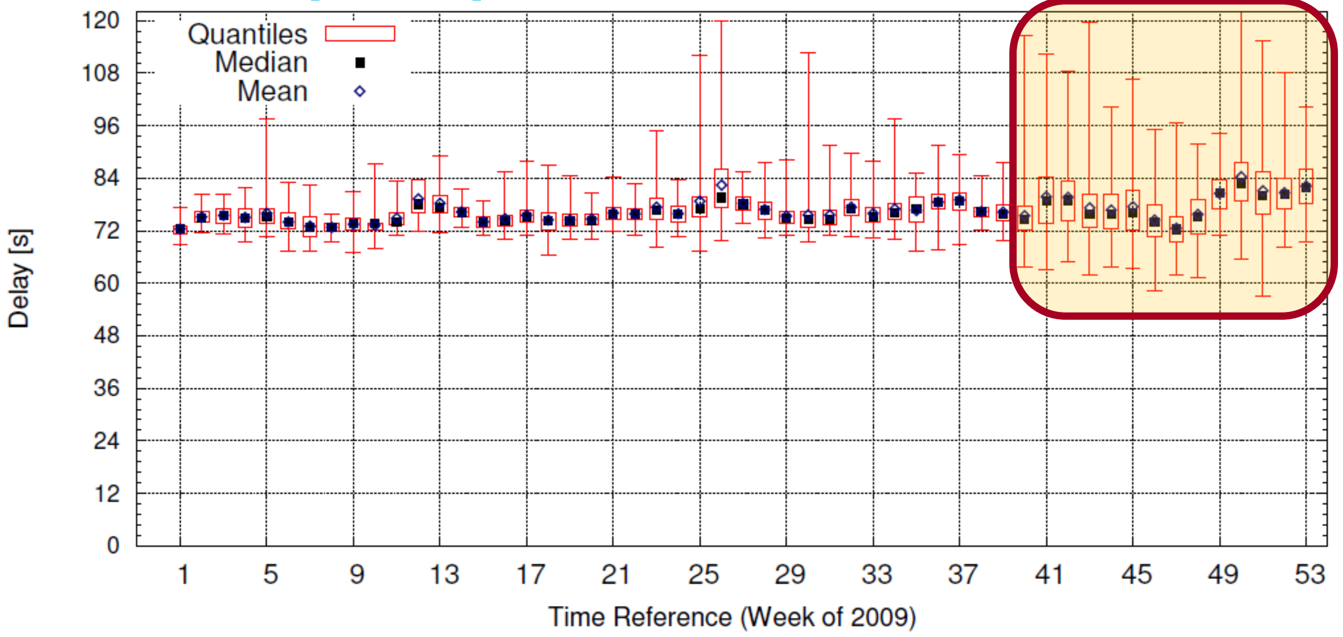


Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

AWS Dataset (1/4): EC2

Variable Performance

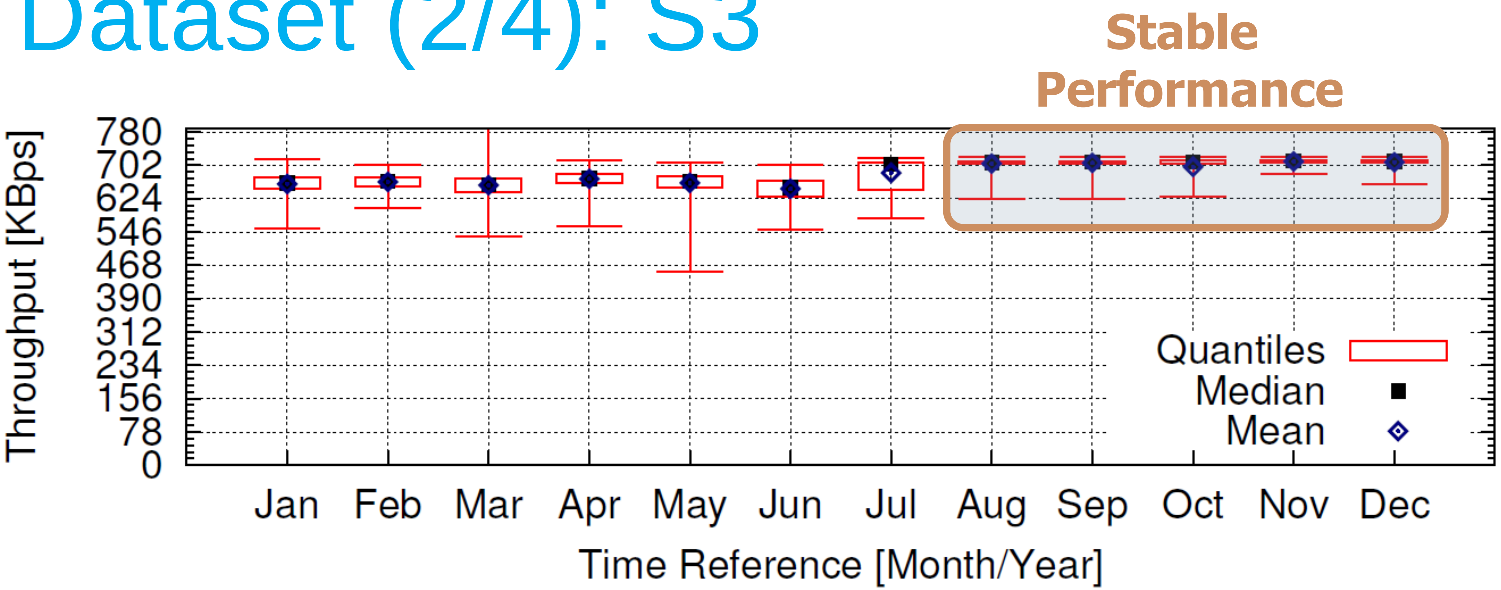
Q2



- **Deployment Latency [s]:** Time it takes to start a small instance, from the startup to the time the instance is available
- Higher IQR and range from week 41 to the end of the year; possible reasons:
    - Increasing EC2 user base → Impact on applications using EC2 for auto-scaling

AWS Dataset (2/4): S3

Q2

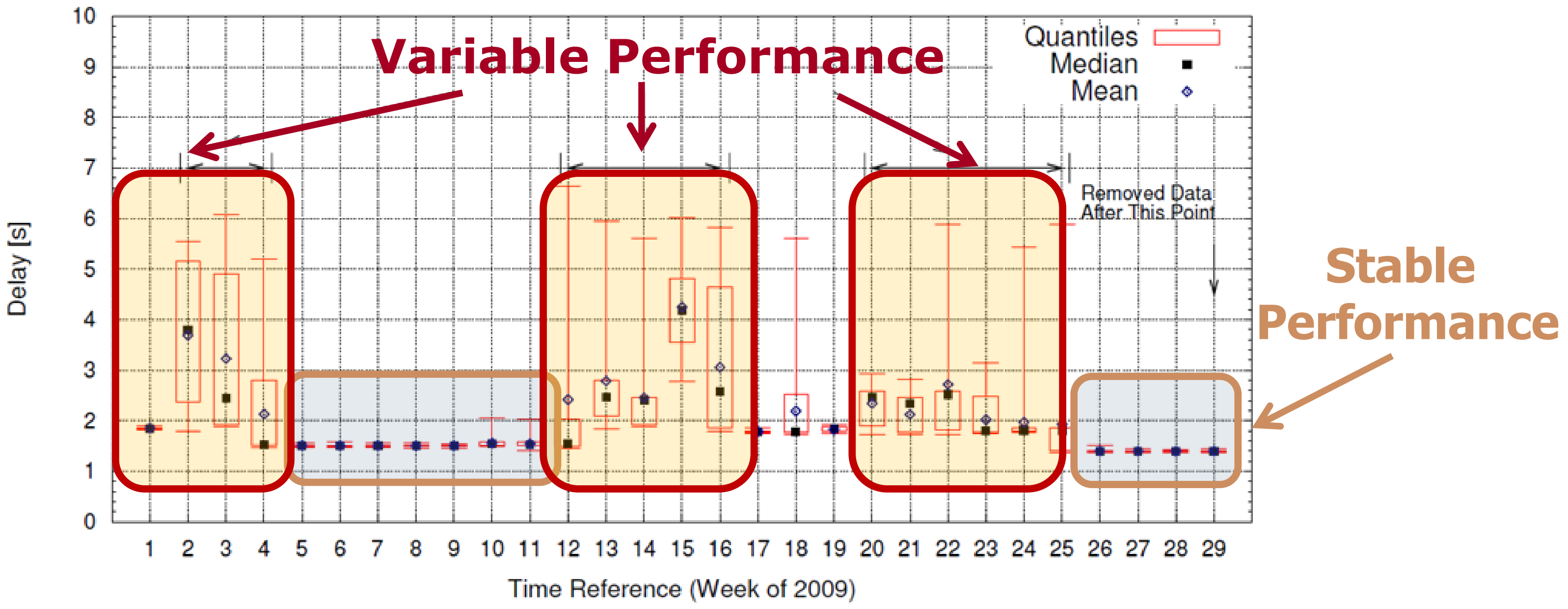


- **Get Throughput [bytes/s]:** Estimated rate at which an object in a bucket is read
- The last five months of the year exhibit much lower IQR and range
  - More stable performance for the last five months
  - Probably due to software/infrastructure upgrades

Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

AWS Dataset (3/4): SQS

Q2



- **Average Lag Time [s]:** Time it takes for a posted message to become available to read. Average over multiple queues.
- Long periods of stability (low IQR and range)
- Periods of high performance variability also exist

© 2017 Alexandru Iosup. All rights reserved.

AWS Dataset (4/4): Summary

Q2

- **All services exhibit time patterns in performance**
- EC2: periods of special behavior
- SDB and S3: daily, monthly and yearly patterns
- SQS and FPS: periods of special behavior

Iosup, Yigitbasi, Epema. On the Performance Variability of Production Cloud Services, (IEEE CCgrid 2011).

Summary

- Lower performance than theoretical peak in IaaS services
  - Especially CPU (GFLOPS)
  - (2007) Explored in study of 4 production clouds, each with several IaaS services
- Performance variability in IaaS and PaaS services
  - Explored in longitudinal study of Amazon Web Services and Google App Engine
  - (2008-2010) Data from cloudstatus.com
- Compared results with some of the commercial alternatives, such as supercomputers and clusters (see report)

© 2017 Alexandru Iosup. All rights reserved.

Alexandru Iosup, Tim Hegeman, Wing-Lung Ngai, Stijn Heldens.

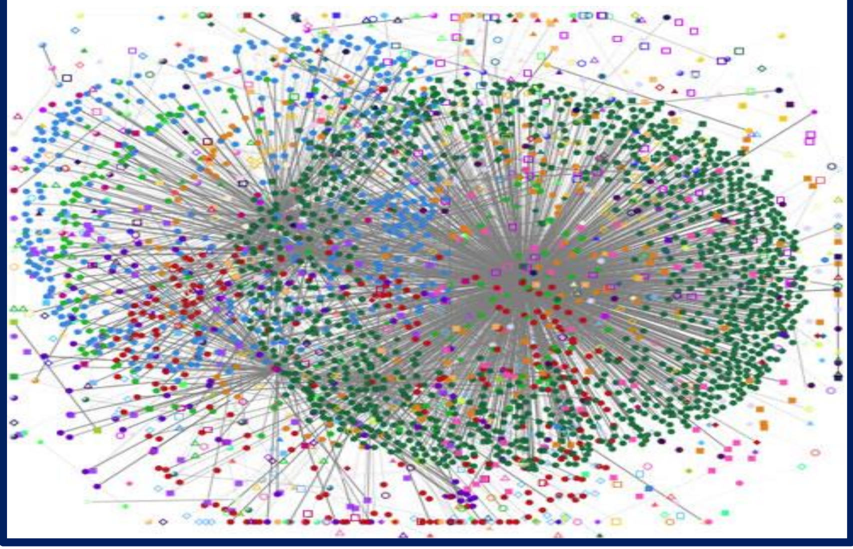
LDBC Graphalytics

A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Systems

Iosup, Hegeman, Ngai, Heldens, Prat-Pérez, Manhardt, Chafi, Capota, Sundaram, Anderson, Tanase, Xia, Nai, Boncz. LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms. PVLDB 9(13): 1317-1328 (2016)

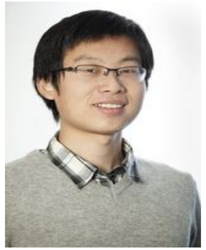
The data deluge: large-scale graphs
tens of Billions of Edges

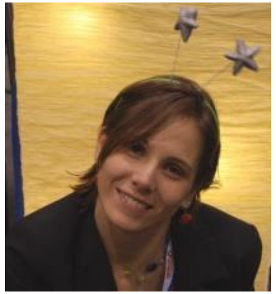
Graph Processing

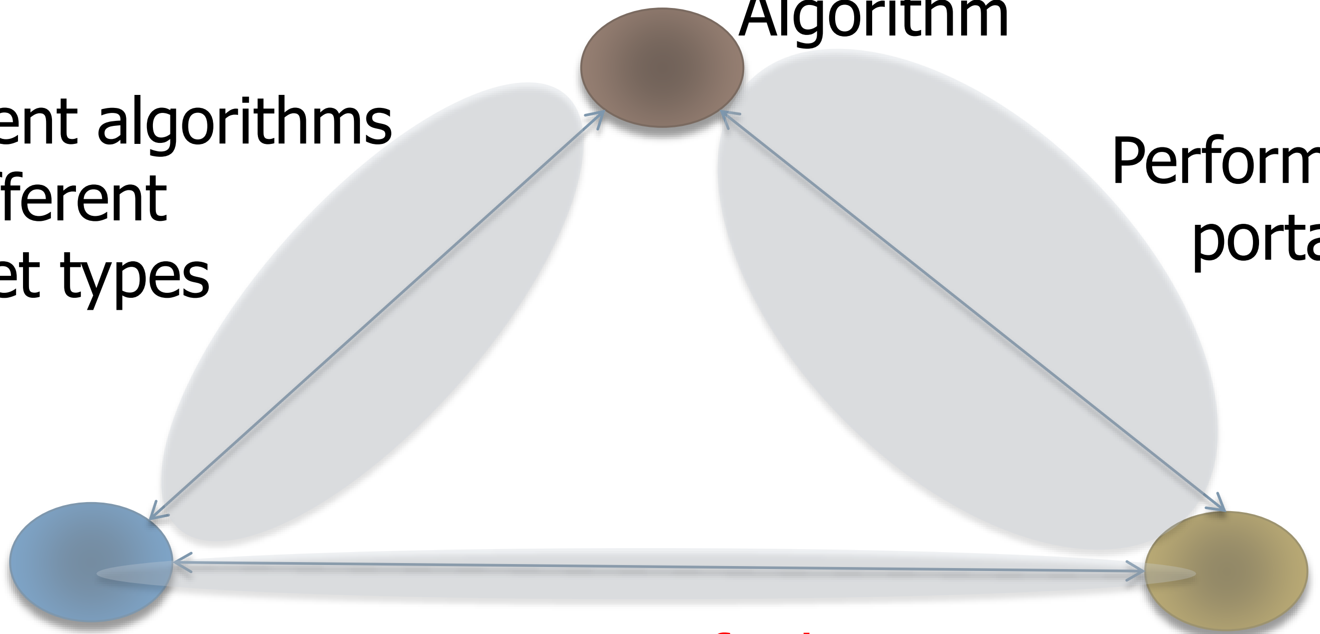


© 2017 Alexandru Iosup. All rights reserved.

Ecosystem Navigation = Understanding the **P**latform-**A**lgorithm-**D**ataset Triangle







Different algorithms for different dataset types

Performance enabled, portability disabled

How does actual data impact performance?

How does deployment impact performance?

No systematic findings yet

A. L. Varbanescu et al. Can Portability Improve Performance? An Empirical Study of Parallel Graph Analytics. ICPE 2015: 277-287

A. Iosup et al. Towards Benchmarking IaaS and PaaS Clouds for Graph Analytics. WBDB 2014: 109-131

Graph Processing Platforms

Which platforms perform well?

What to tune?
What to re-design?

What Is the Performance of Graph Processing Platforms?

Metrics Diversity | **Graph Diversity** | **Algorithm Diversity**

- Graph500
  - Single application (BFS), Single class of synthetic datasets. @ISC16: future diversification.
- Few existing platform-centric comparative studies
  - Prove the superiority of a given system, limited set of metrics
- GreenGraph500, GraphBench, XGDBench
  - Issues with representativeness, systems covered, metrics, …

What Is the Performance of Graph Processing Platforms?

Metrics Diversity	Graph Diversity	Algorithm Diversity

LDBC The graph & RDF benchmark reference

Graphalytics = comprehensive benchmarking suite for graph processing across many platforms

http://ldbcouncil.org/ldbc-graphalytics

http://graphalytics.org/

LDBC The graph & RDF benchmark reference

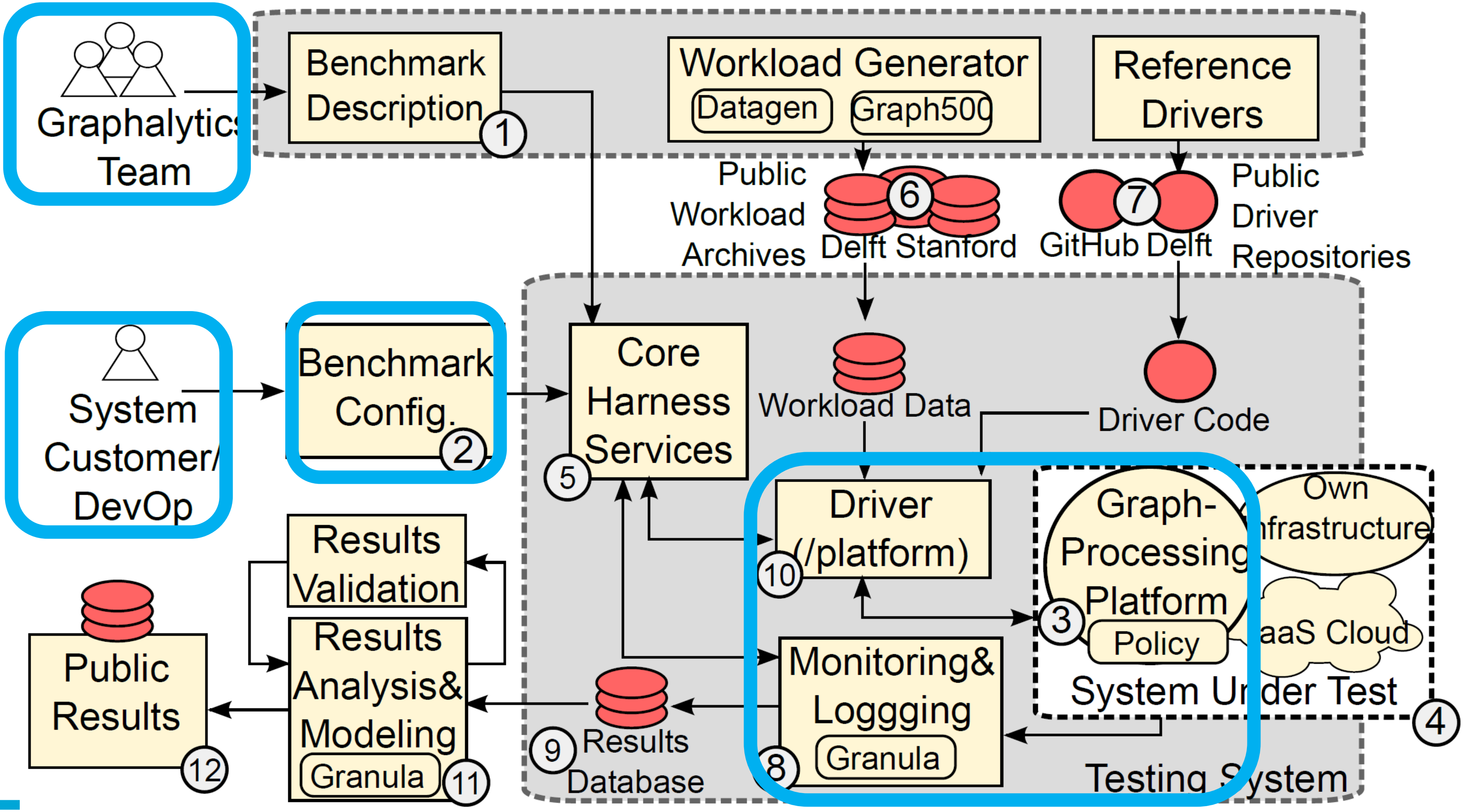
Graphalytics, in a nutshell

- An LDBC benchmark **http://ldbcouncil.org/ldbc-graphalytics**
- Advanced benchmarking harness
- Many classes of algorithms used in practice
- Diverse real and synthetic datasets
- Diverse set of experiments representative for practice
- Renewal process to keep the workload relevant
- Extended toolset for manual choke-point analysis
- Enables comparison of many platforms, community-driven and industrial

[Iosup et al., VLDB'16] [Guo et al., CCGRID'15] [Guo et al., IPDPS'14]

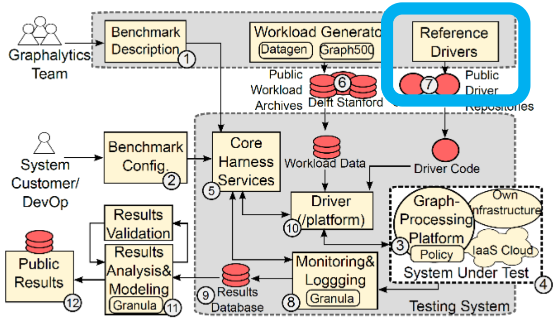
http://graphalytics.org

Graphalytics = Benchmarking Harness



Iosup et al. LDBC Graphalytics: A Benchmark for Large Scale Graph Analysis on Parallel and Distributed Platform, PVLDB'16.

Graphalytics = Representative Classes of Algorithms and Datasets



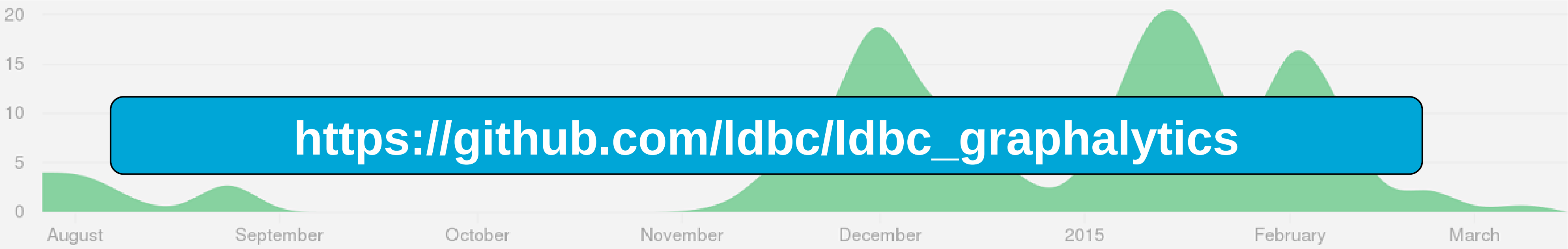
- 2-stage selection process of algorithms and datasets

Class	Examples	%
Graph Statistics	Diameter, Local Clust. Coeff. PageRank	20
Graph Traversal	BFS SSSP, DFS	50
Connected Comp.	Reachability, BiCC, Weakly CC	10
Community Detection	Clustering, Nearest Neighbor, Community Detection w Label Propagation	5
Other	Sampling, Partitioning	<15

+ property/weighted graphs: Single-Source Shortest Paths (~35%)

Guo et al. How Well do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis, IPDPS'14.

Graphalytics = Modern Software Engineering Process



- Graphalytics code reviews
  - Internal release to LDBC partners (first, Feb 2015; last, Feb 2016)
  - Public release, announced first through LDBC (Apr 2015)
  - First full benchmark specification, LDBC criteria (Q1 2016)
- Jenkins continuous integration server
- SonarQube software quality analyzer

Wing Lung Ngai

Tim Hegeman

Stijn Heldens

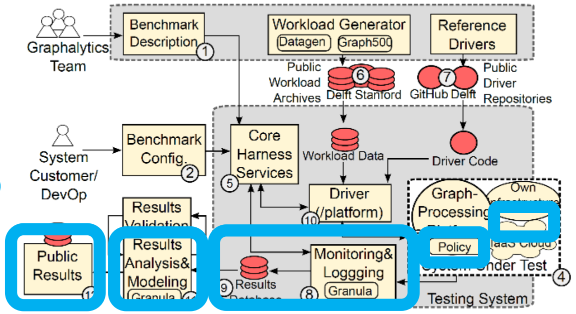
Alexandru Iosup

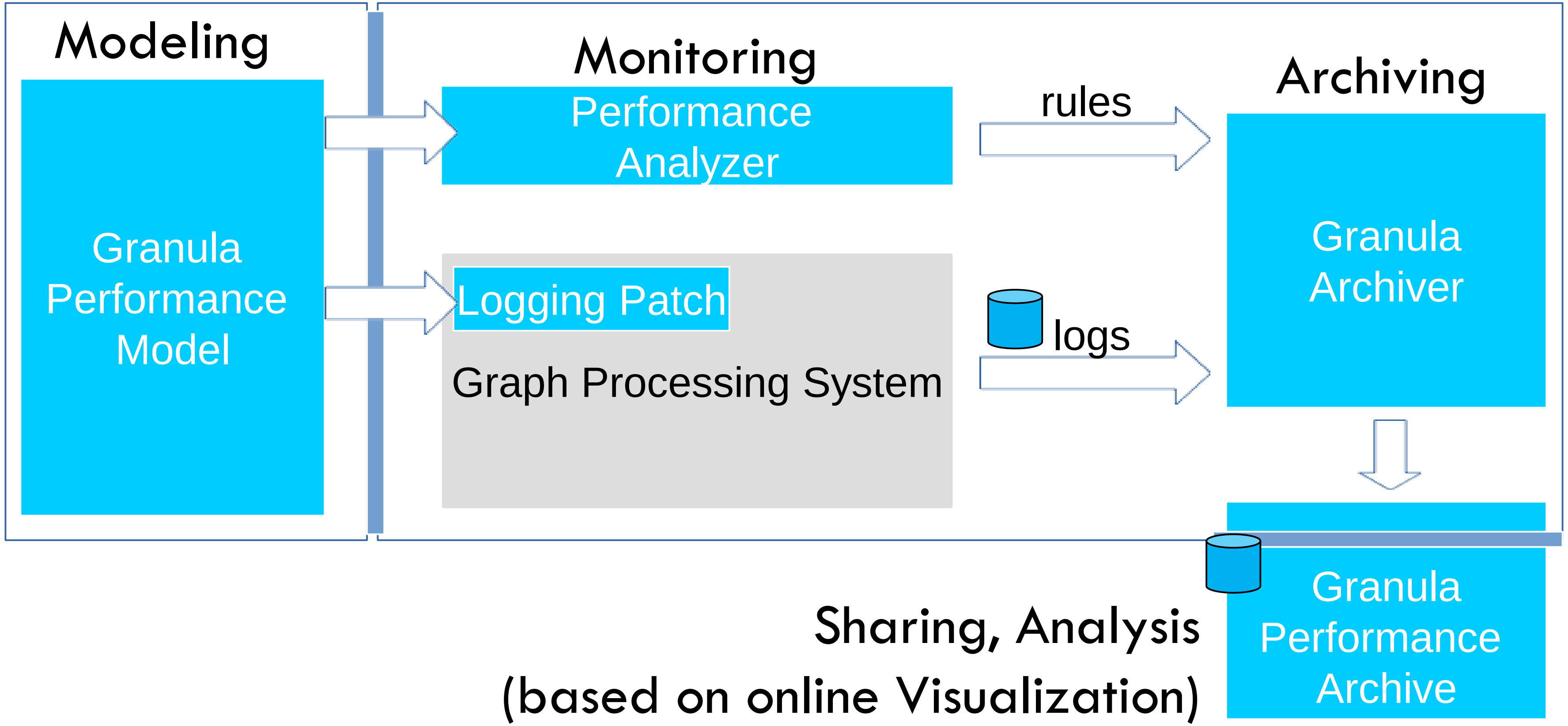
Graphalytics Granula

Monitoring, Archiving, and Sharing Data about Large-scale Graph-Processing Platforms (LSGPPs)
Incremental Performance Modeling, and Fine-grained Performance Analysis of LSGPPs

Ngai, Hegeman, Heldens, Iosup: Granula: Toward Fine-grained Performance Analysis of Large-scale Graph Processing Platforms. GRADES@SIGMOD/PODS 2017: 8:1-6

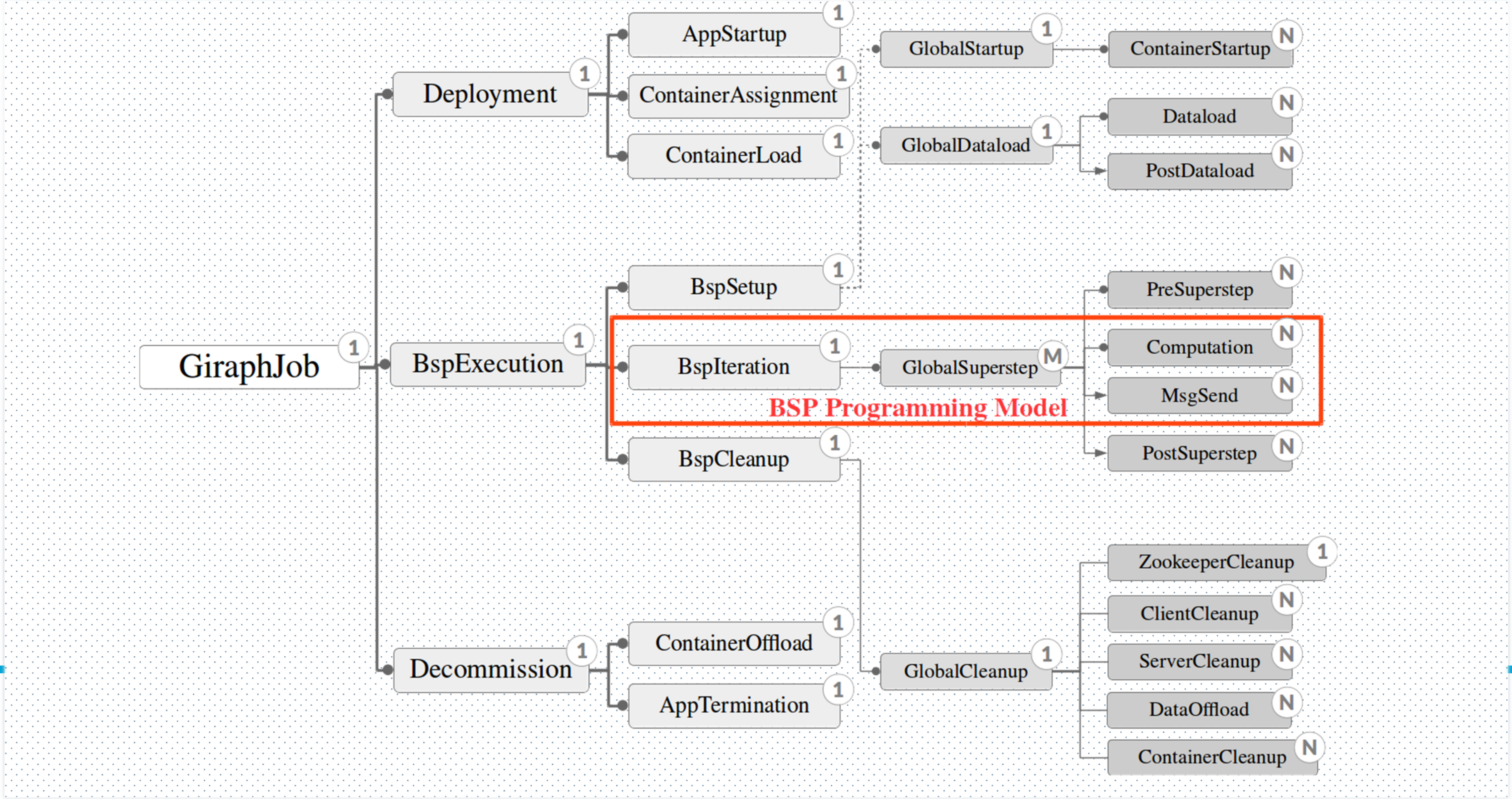
Granula: Portable Performance Analysis



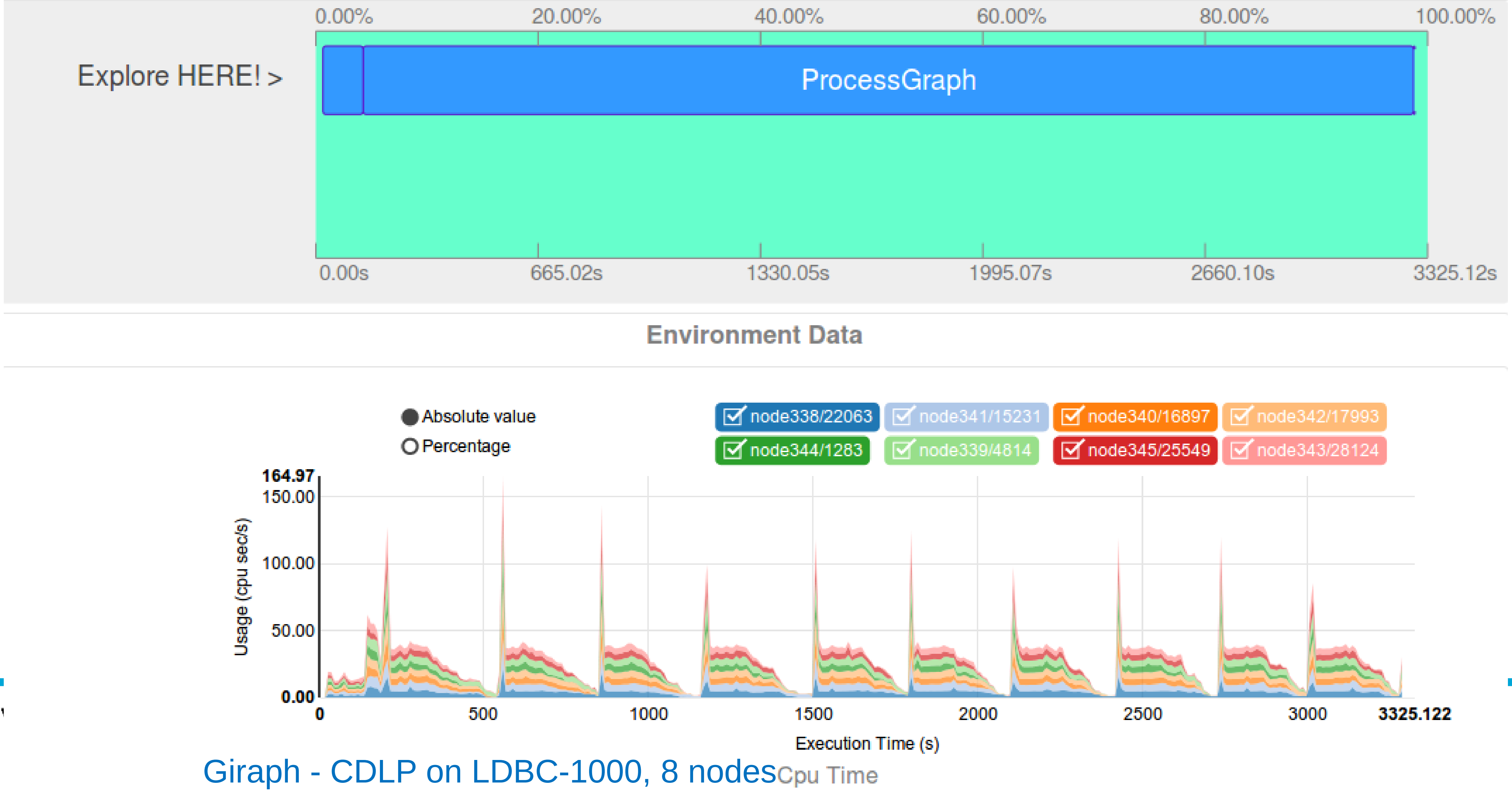


Minimal code invasion + automated data collection at runtime + portable archive (+ web UI) → portable bottleneck analysis

Incremental Performance Modelling with Granula

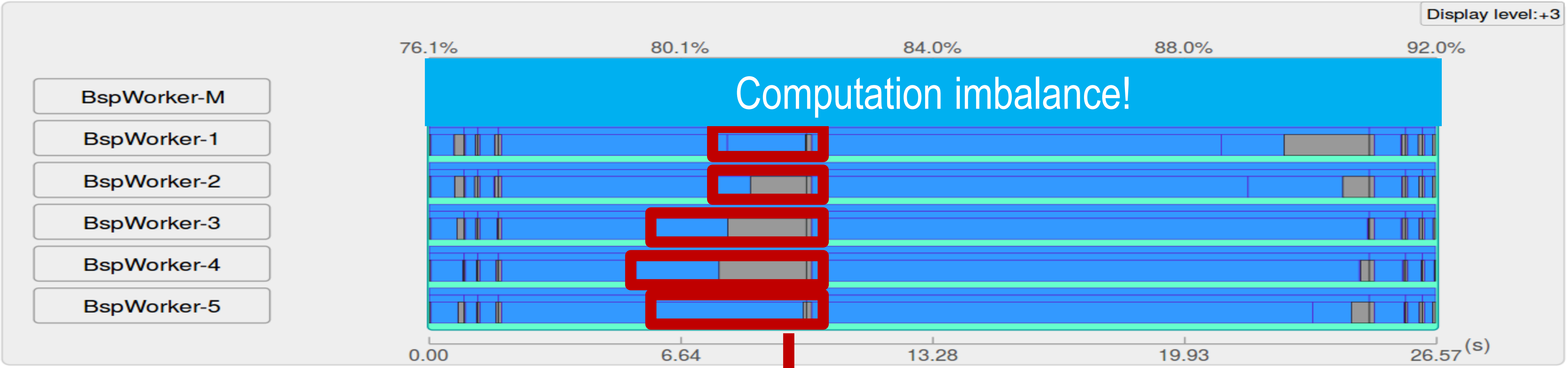


Performance Monitoring, Archiving, Visualization with Granula

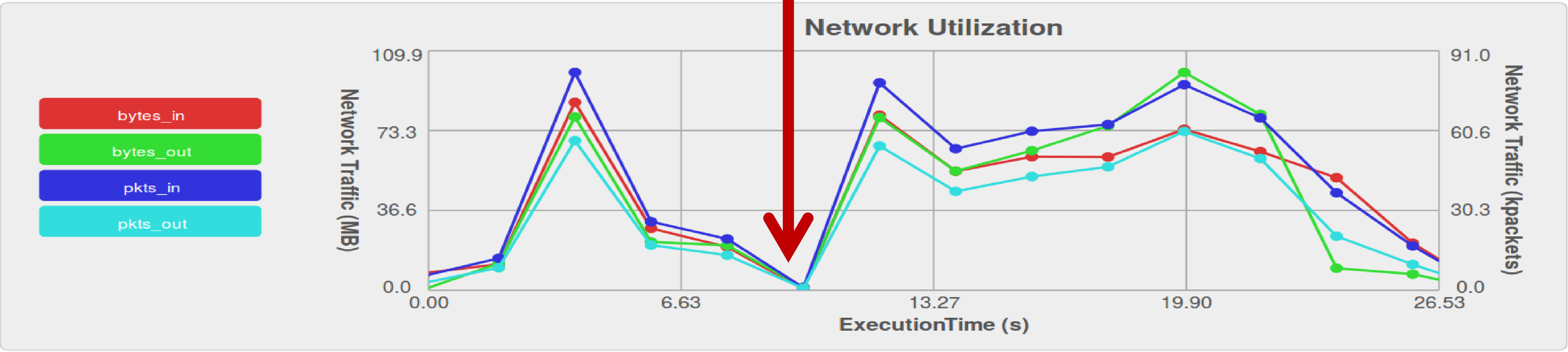


Giraph - CDLP on LDBC-1000, 8 nodes

Granula: Performance Modeling, Visualization, Analysis



Giraph - BFS on LDBC-1000, 5 nodes



Tim Hegeman

Alexandru Iosup

Graphalytics Grade10

A System for Fine-grained Performance Analysis, Bottleneck Identification, and Performance-Issue Detection in Large-scale Graph Processing Platforms

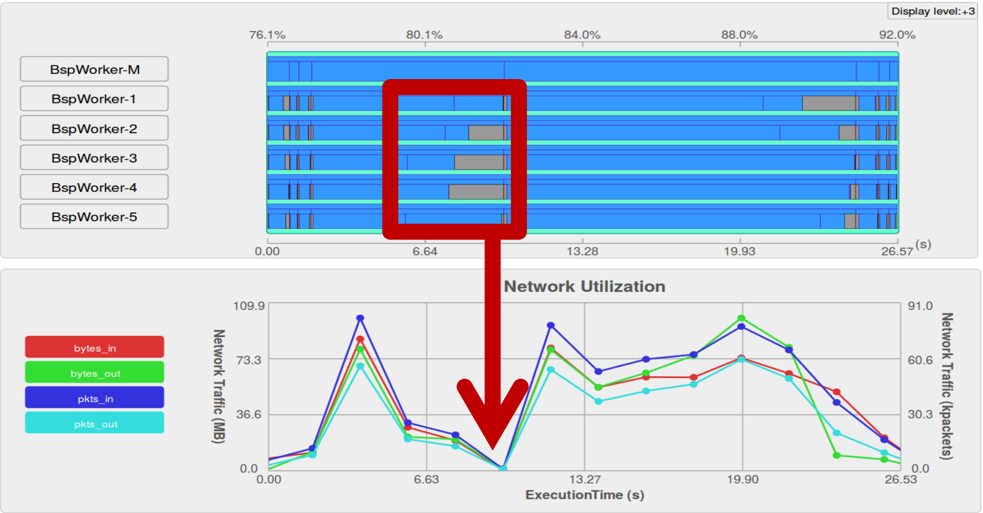
(Sep 2017)

(unpublished, so please do not record or share)

Grade10: Performance Bottleneck Identification

Analytical modeling is time-consuming. Profiling (aggregating) and full tracing are data-intensive. All are expertise-driven.

Grade10 analyses Granula and resource utilization data for you.



Possible performance bottlenecks:

- 20% slowdown due to imbalance in ‘Computation’ phase
- HW resource bottlenecks of ‘GlobalSuperstep’: CPU 60%, network 30%, none 10%

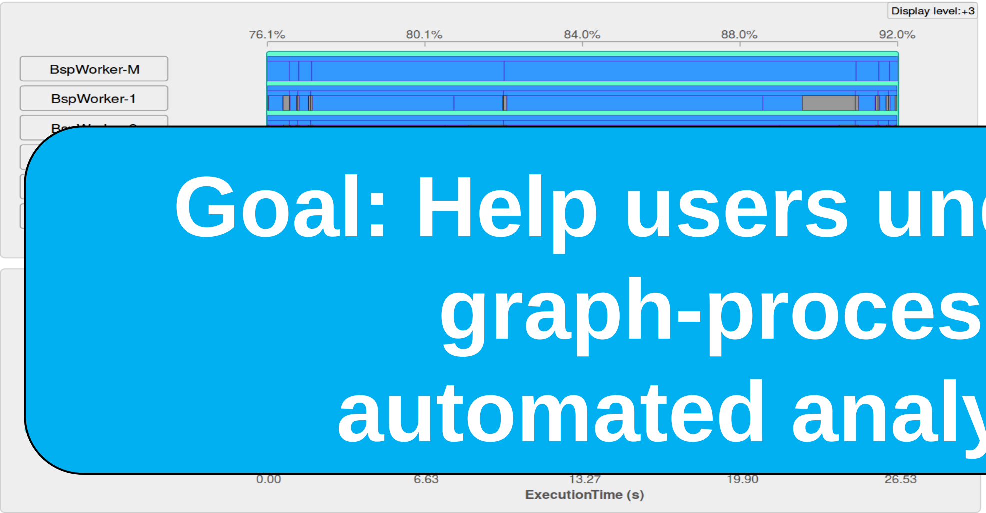
VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft Delft University of Technology

Grade10: Performance Bottleneck Identification

Analytical modeling is time-consuming. Profiling (aggregating) and full tracing are data-intensive. All are expertise-driven.

Grade10 analyses Granula and resource utilization data for you.



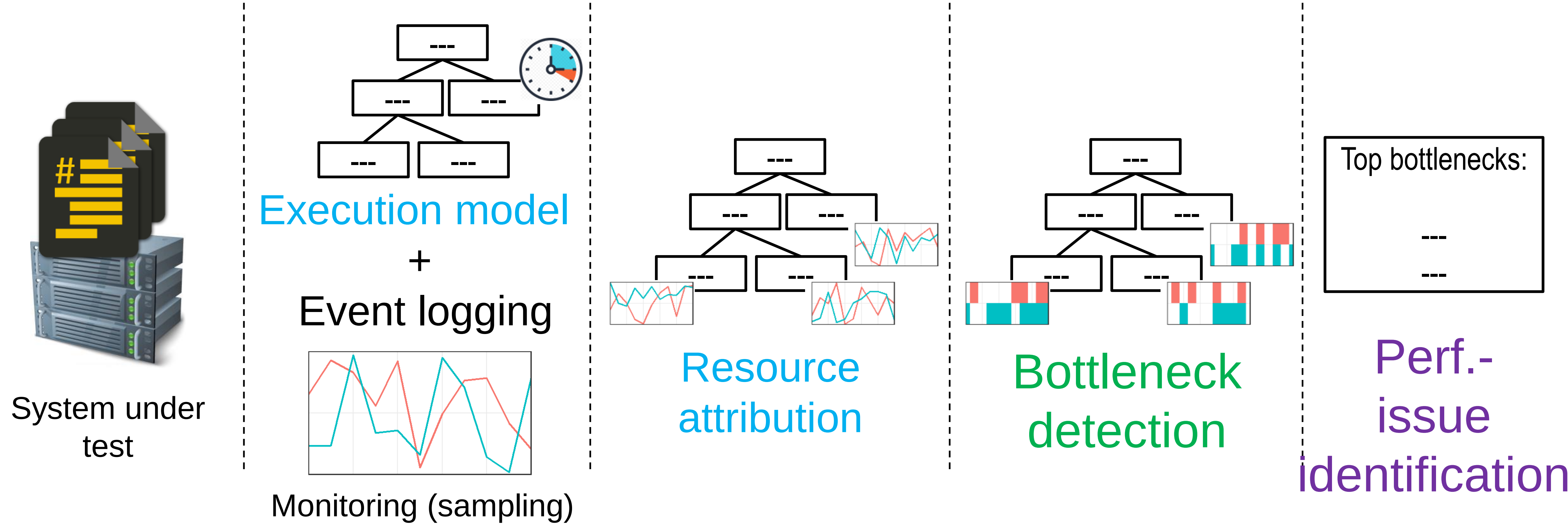
Possible performance bottlenecks:

Goal: Help users understand the performance of graph-processing systems through automated analysis of performance data

VU VRIJE UNIVERSITEIT AMSTERDAM

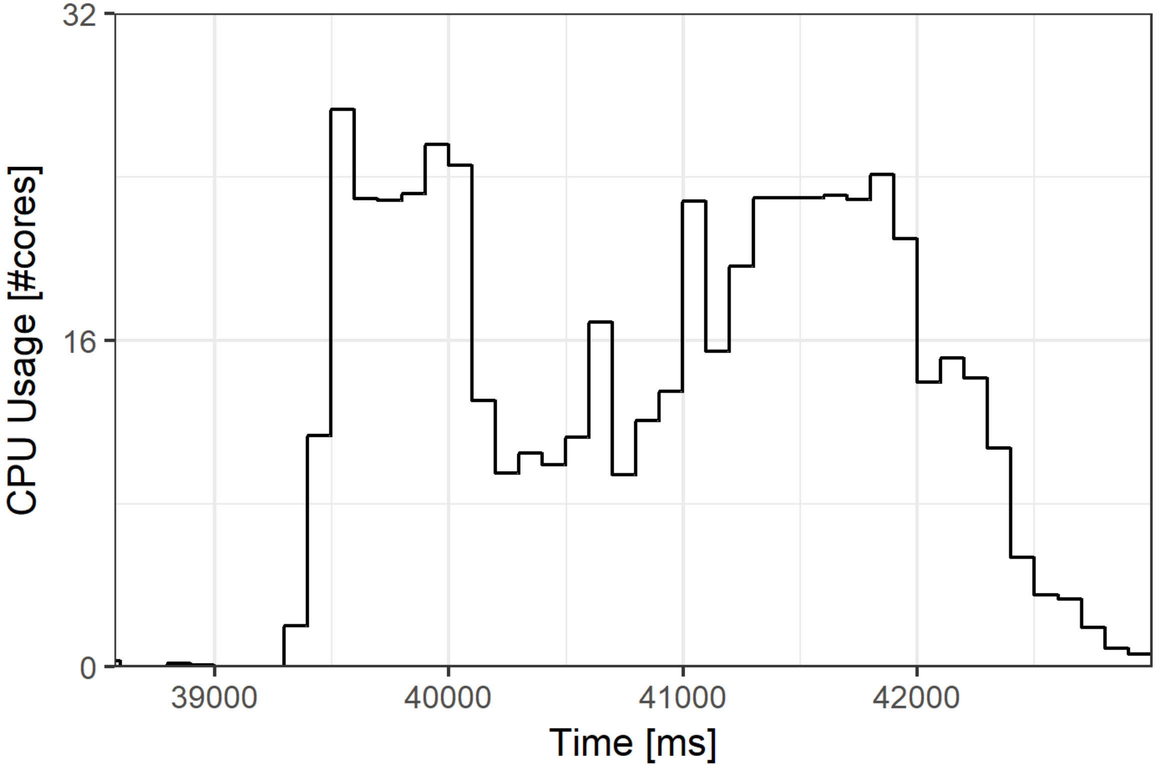
TU Delft Delft University of Technology

Grade10: Automated Bottleneck Detection and Performance Issue Identification



VU VRIJE UNIVERSITEIT AMSTERDAM
TU Delft Delft University of Technology

Preliminary Result: Analysing a Giraph Job

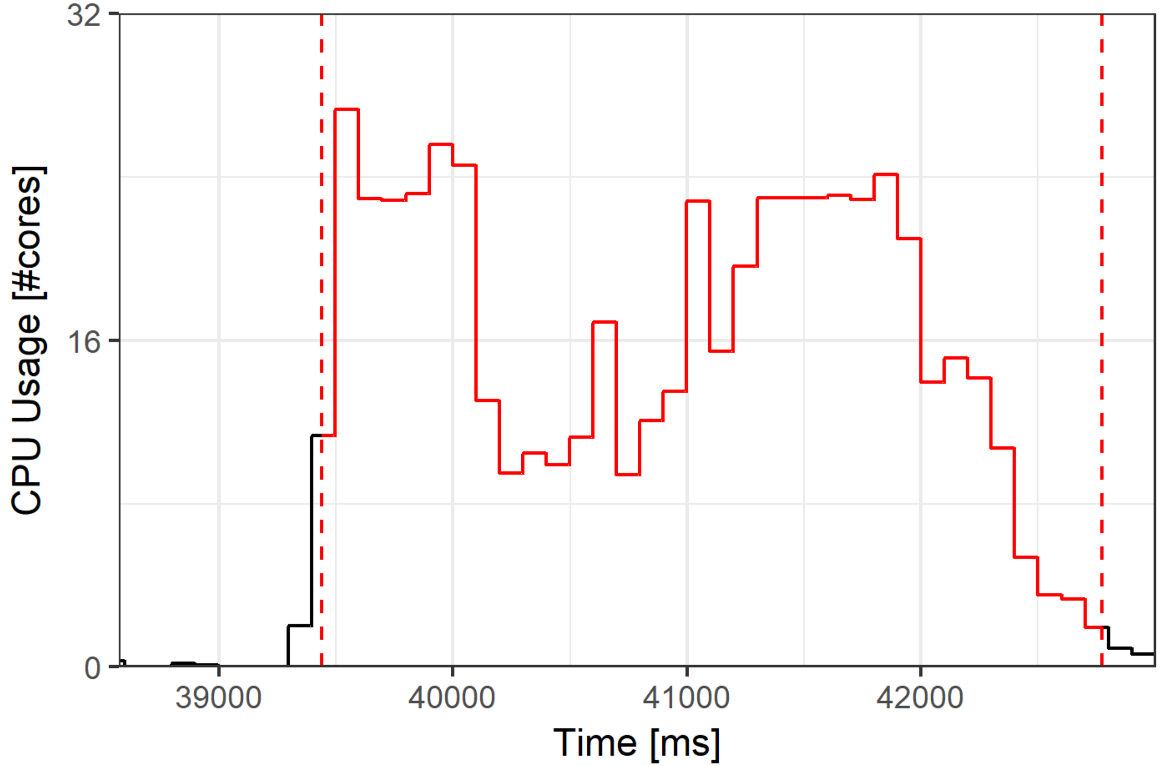


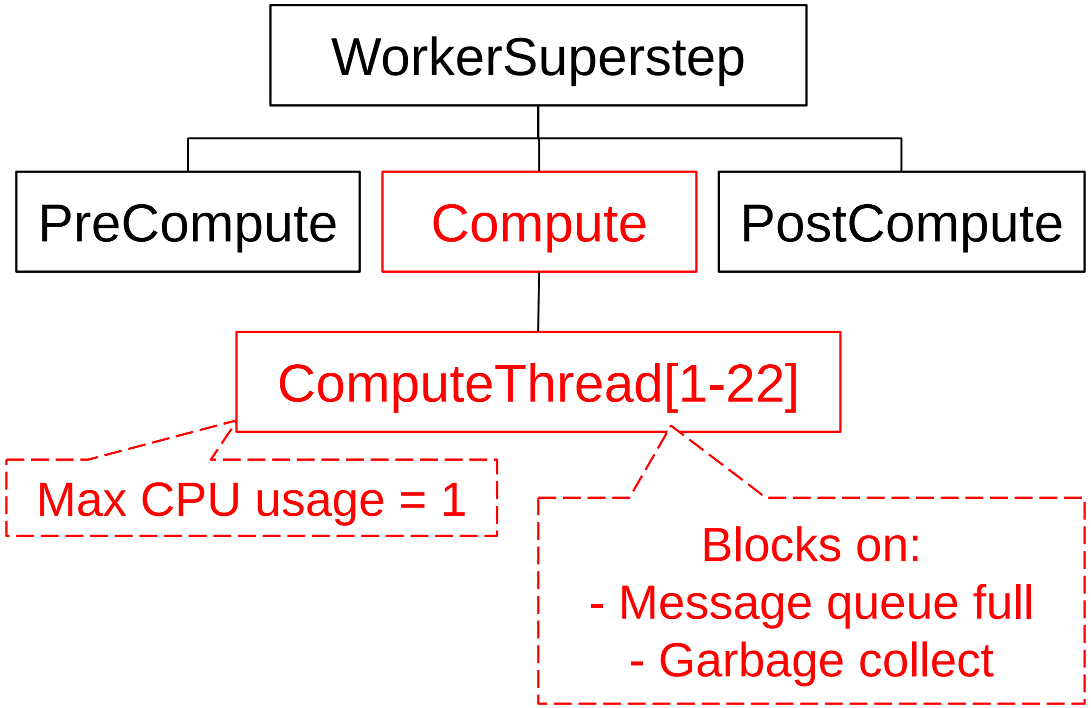
WorkerSuperstep

CPU usage < 32 cores (100%), so no bottleneck

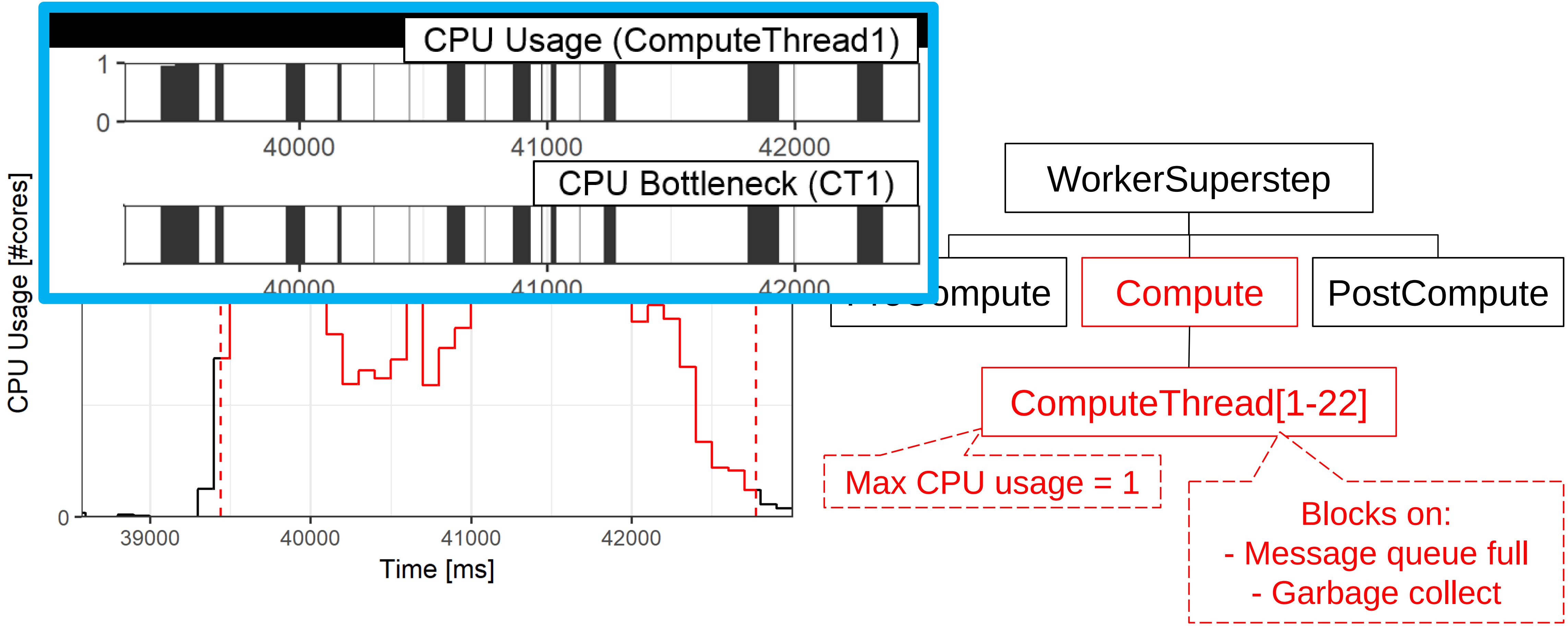
... yet

Preliminary Result: Analysing a Giraph Job



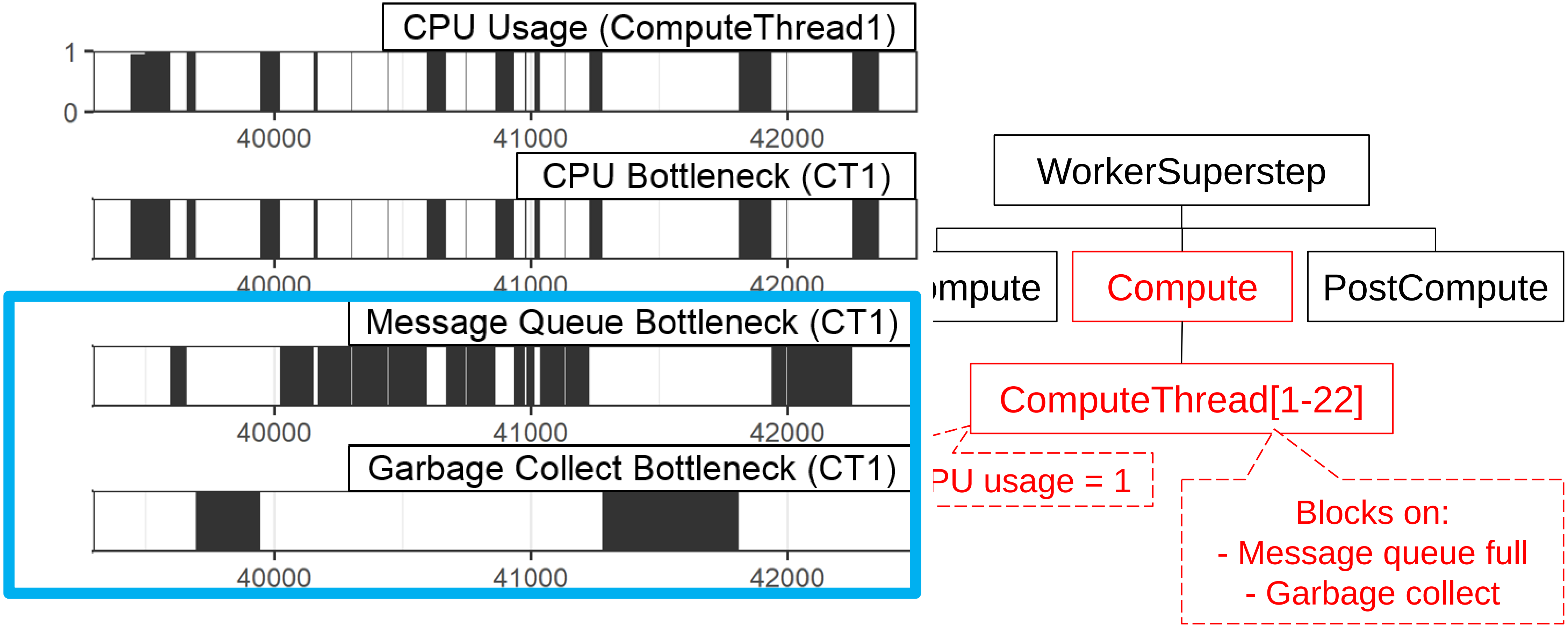


Preliminary Result: Analysing a Giraph Job

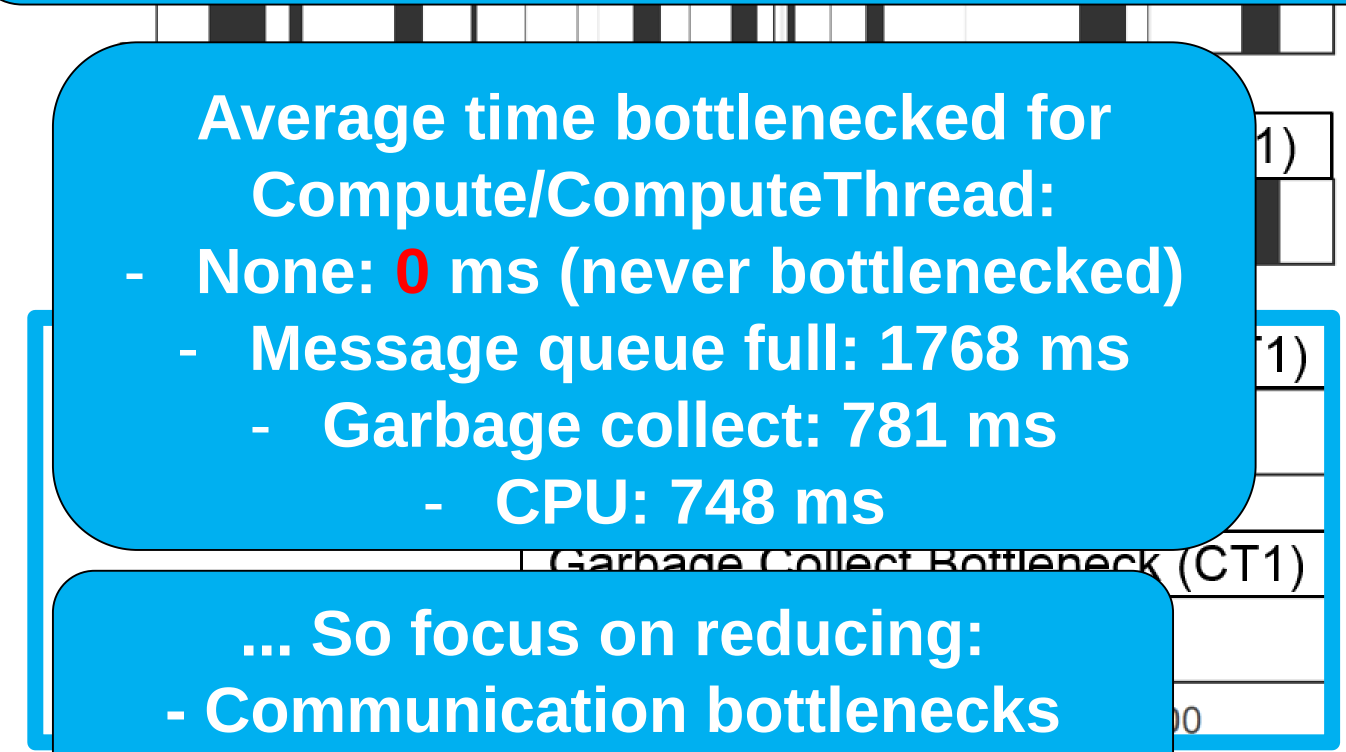


VU VRIJE UNIVERSITEIT AMSTERDAM
TUDelft Delft University of Technology

Preliminary Result: Analysing a Giraph Job



Grade10 : Help users understand the performance of graph-processing systems through automated analysis of performance data



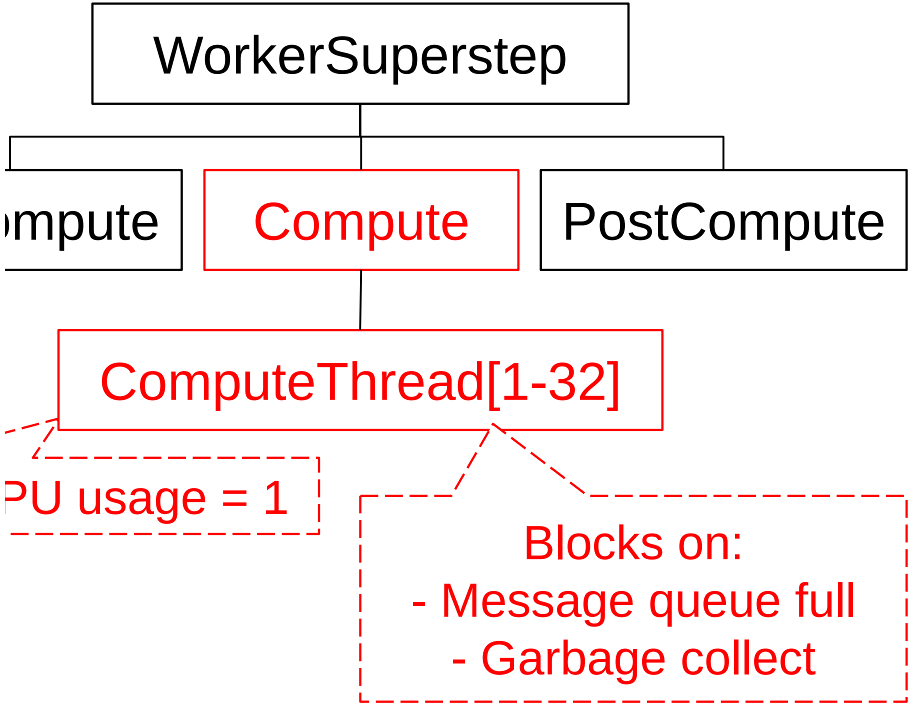
Average time bottlenecked for Compute/ComputeThread:

- **None: 0 ms (never bottlenecked)**
- **Message queue full: 1768 ms**
- **Garbage collect: 781 ms**
- **CPU: 748 ms**

... So focus on reducing:

- Communication bottlenecks

- GC overheads (good luck!)



Jerom
van der Sar

Jesse
Donkervliet

Alexandru
Iosup

Yardstick

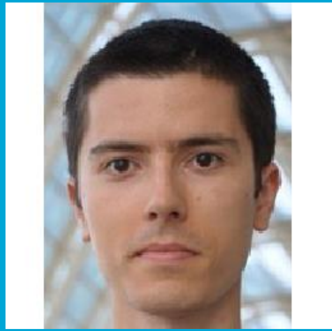
A Benchmark for Minecraft-like Games

(Jun 2017)

(unpublished, so please do not record or share)

Bogdan
Ghiț

Tim
Hegeman



Mihai
Capotã

Dick
Epema

Alexandru
Iosup

Taming Big Data Vicissitude

Tuning the BTWorld MapReduce-based workflow for time-based Big Data analytics

Ghit, Capota, Hegeman, Hidders, Epema, Iosup. V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows. CCGRID 2014: 927-932

Hegeman, Ghit, Capota, Hidders, Epema, Iosup. The BTWorld use case for big data analytics: Description, MapReduce logical workflow, and empirical evaluation. BigData Conference 2013: 622-630

Wojciechowski, Capota, Pouwelse, Iosup. BTWorld: towards observing the global BitTorrent file-sharing network. HPDC Workshops 2010: 581-588

The New “Jevon’s Effect”: The “Data Deluge”

Data Deluge =

data generated by humans and devices (IoT)

- Interacting
- Understanding
- Deciding
- Creating

Need to address
Volume, Velocity, Variety of Big Data*

Vicissitude of Big Data = dynamic mix of big data issues (Vs) that lead in big data systems to different bottlenecks over time

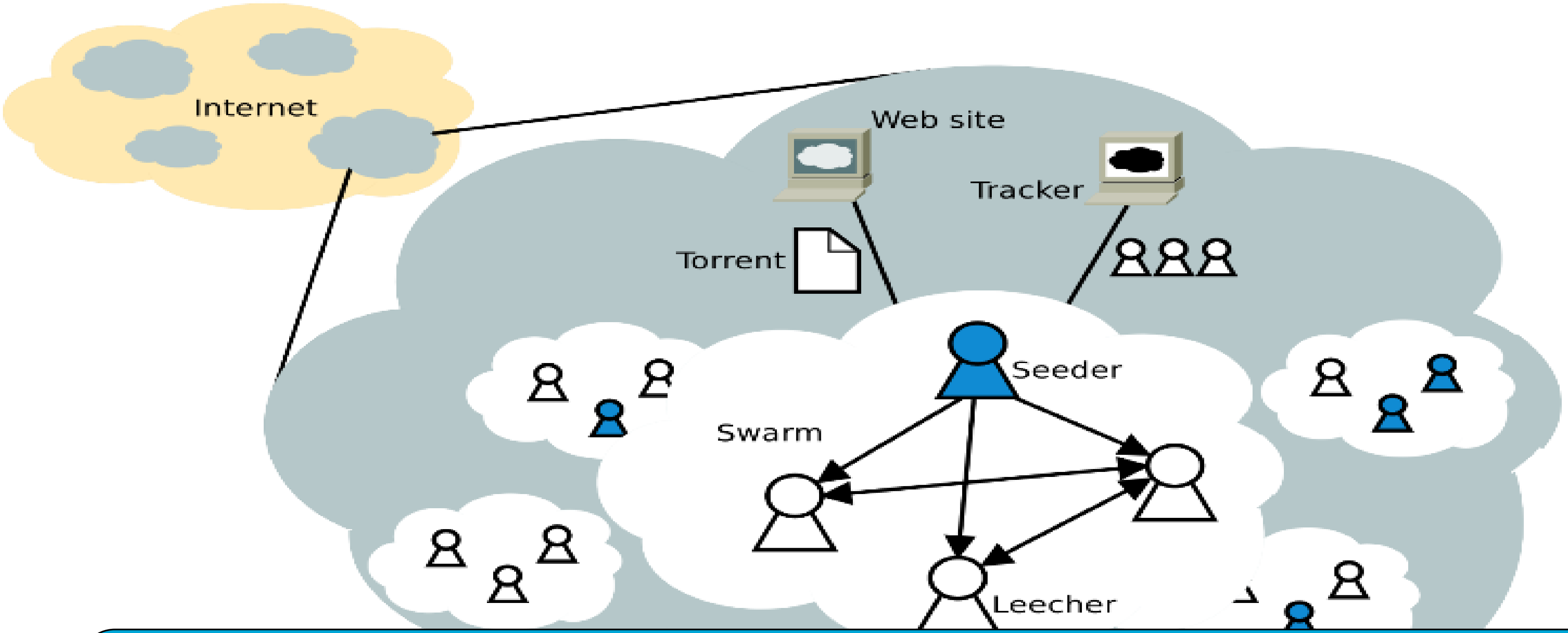
VU UNIVERSITEIT AMSTERDAM

Sources: IDC, EMC.

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Monitoring A Typical Global System: BitTorrent



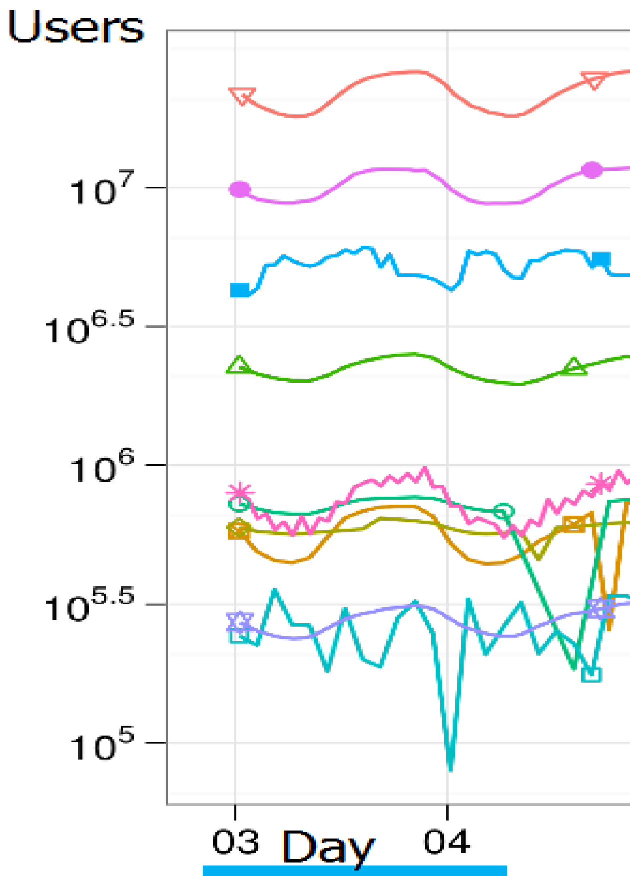
Most used protocol on Internet, by upload volume [1]
One third (US) to half (EU) of residential upload
Over 100 million users [2]

[1] https://sandvine.com/downloads/general/global-internet-phenomena/2013/2h-2013-global-internet-phenomena-report.pdf
[2] http://www.bittorrent.com/company/about/ces_2012_150m_users

© 2017 Alexandru Iosup. All rights reserved.

BTWorld: a Typical Big Data Project

- Ongoing longitudinal study, 5 YEARS
- Data-driven project to understand BitTorrent: data first, ask questions later
  - Over 15 TB of structured and semi-structured data added during the project
  - Queries added during project, e.g.,
    How does the BitTorrent population vary?
    How does BitTorrent change over time?



VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

The MapReduce Ecosystem (a big problem in big data)

- Widely used in industry and academia
  - Similar to other big data stacks
- Complex software to tune
  - 100s of parameters
  - Non-linear effects common
- Lots of issues cause crashes [1]
- Focus on Small and Medium Enterprises (60% GPD)
  - No resources or even competence to fix issues
  - Difficult to make stack work for own problems

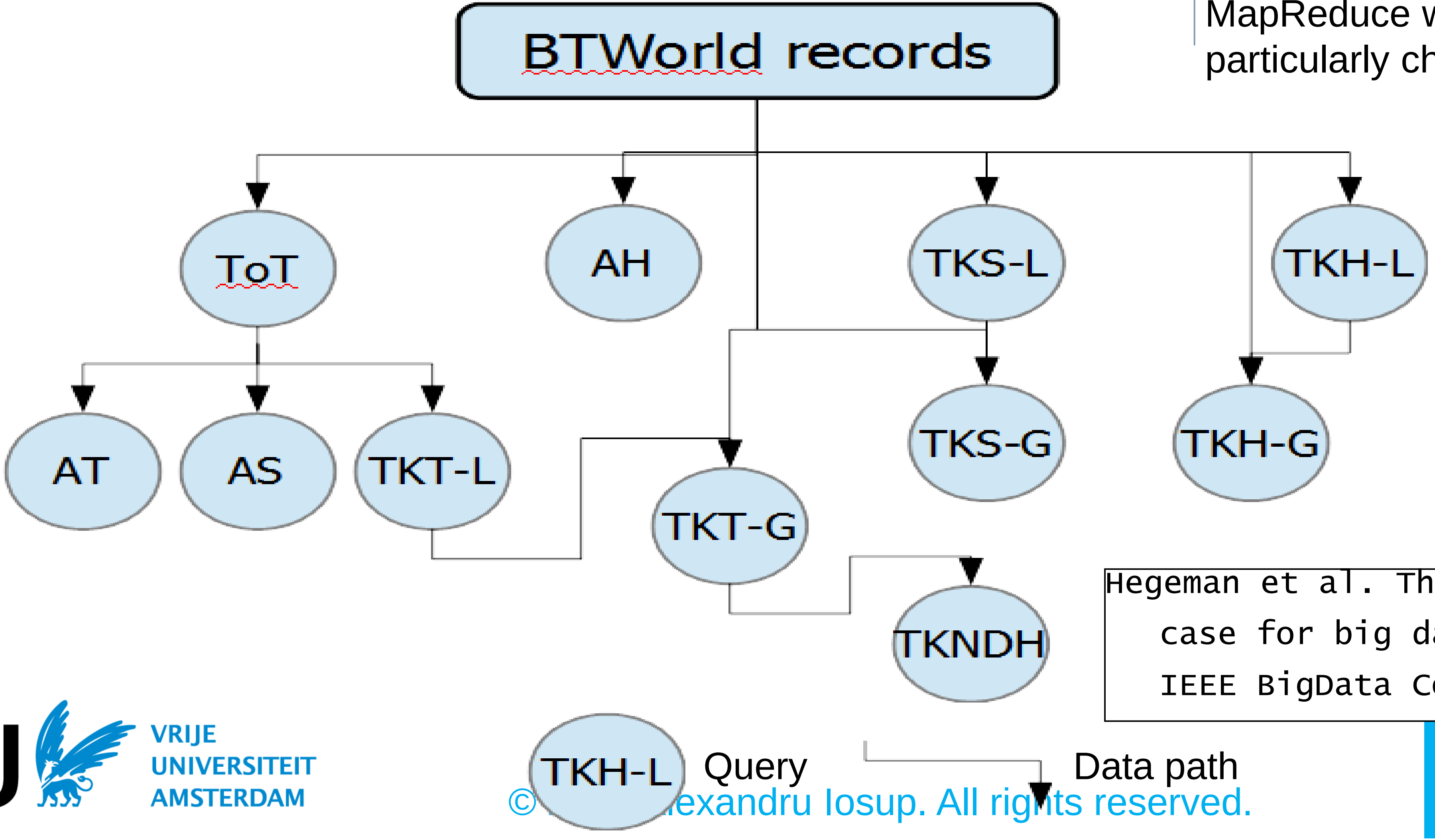
Pig, Hive, …
↓
MapReduce Model
↓
Hadoop/ YARN
↓
HDFS

[1] Ewen et al., "Spinning Fast Iterative Data Flows", PVLDB 2012

© 2017 Alexandru Iosup. All rights reserved.

The Abstract BTWorld Workflow

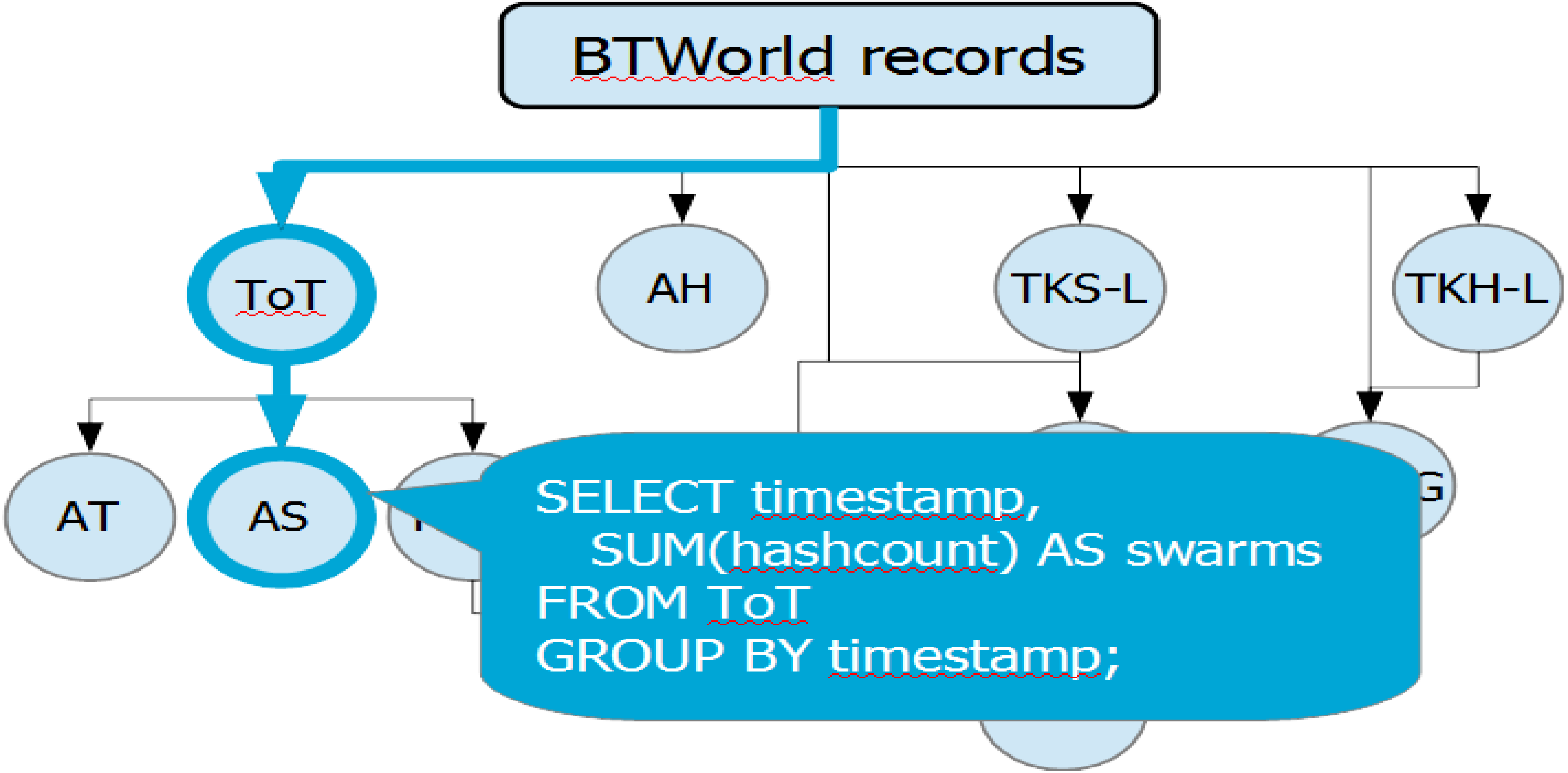
Workflows pose significant scheduling challenges, and MapReduce workflows can be particularly challenging



Hegeman et al. The BTWorld use case for big data analytics. IEEE BigData Conference 2013

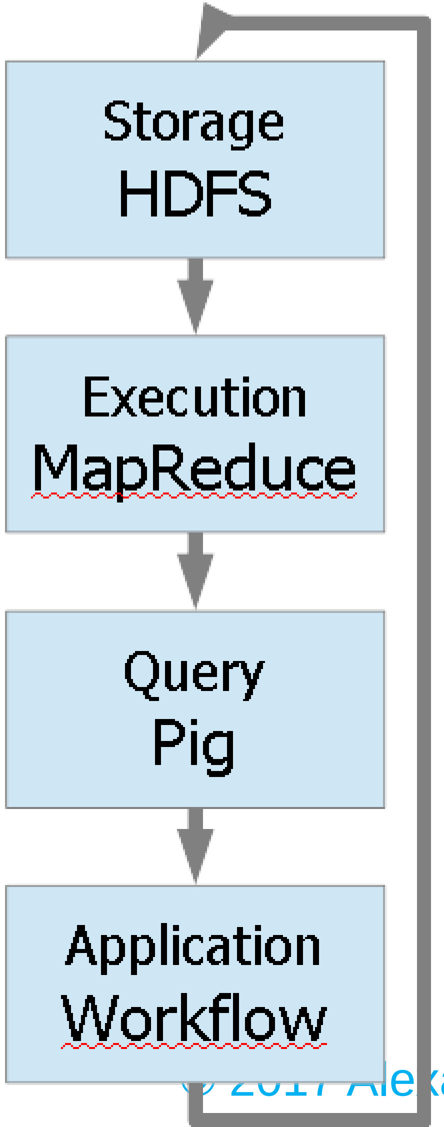
© ... lexandru Iosup. All rights reserved.

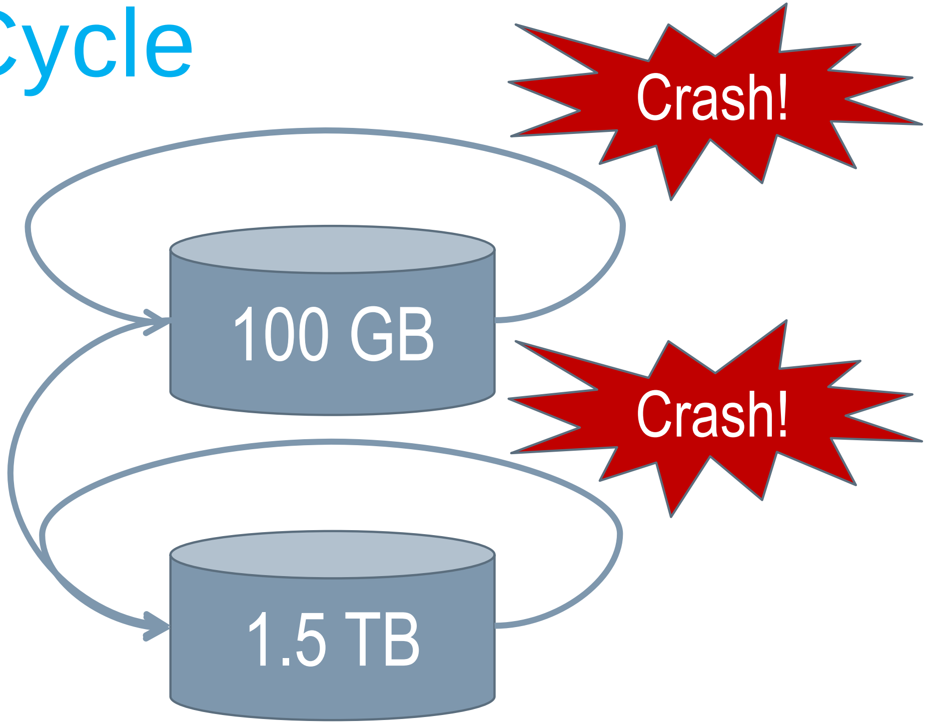
The BTWorld Workload



© 2017 Alexandru Iosup. All rights reserved.

Our Optimization / Tuning Cycle





- HDFS: reduced replication, concatenate small files
- MapReduce: memory per task vs number of tasks, mappers then reducers, etc.
- Pig: specialized joins, multistage adaptive joins
- Workflow: reuse data between stages, common queries

B. Ghit et al. V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows. CCGRID 2014

Approach Addresses a More General Problem

Domain	Data Collection	Entities	Identifiers
BitTorrent	Trackers	Swarms	Hashes
Finance	Stock markets	Stock listings	Stocks
Tourism	Travel agents	Vacation packages	Venues

Alexandru

Won IEEE Scale Challenge 2014!

3. Distributed Resources / Ops Services

- Cloud, grid, cluster, and hybrid computing models
  - Support for workloads of Bags-of-Tasks and Many-tasks
  - Support for workloads of Workflows
- Mechanisms and Architectures
  - Social computing for file sharing
  - Eventual consistency for online games
- Resource management
  - Distributed CPU+GPU operation
  - VM placement
- Systems
  - 2fast
  - Opencraft Meerkat

3 Distributed Resources and Services | Operations Services

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Paweł
Garbacki

Alexandru
Iosup

Dick
Epema

Maarten
van Steen

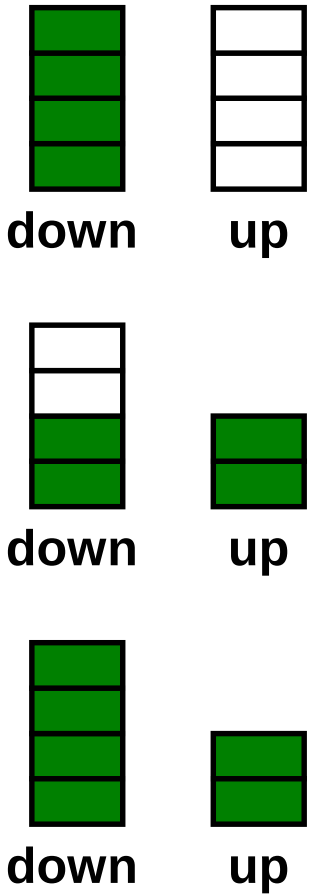
2fast

Collaborative Downloads in P2P Networks

P. Garbacki, A. Iosup, D.H.J. Epema, and M. van Steen, "2Fast: Collaborative Downloads in P2P Networks," *6-th IEEE International Conference on Peer-to-Peer Computing*, 2006 (**best-paper award**).

Peer-to-peer data transfer protocols

- Gnutella, Kazaa
  - no incentives for bandwidth sharing
  - free-riders sensitive
  - poor utilization of upload bandwidth
- BitTorrent (BT), Slurpie
  - tit-for-tat enforces fairness
  - temporal fairness cannot handle asymmetric links
  - poor utilization of download bandwidth
- **2Fast: BT+collaborative downloads**
  - no tit-for-tat within a single session
  - cross-session bandwidth sharing
  - full utilization of upload AND download links



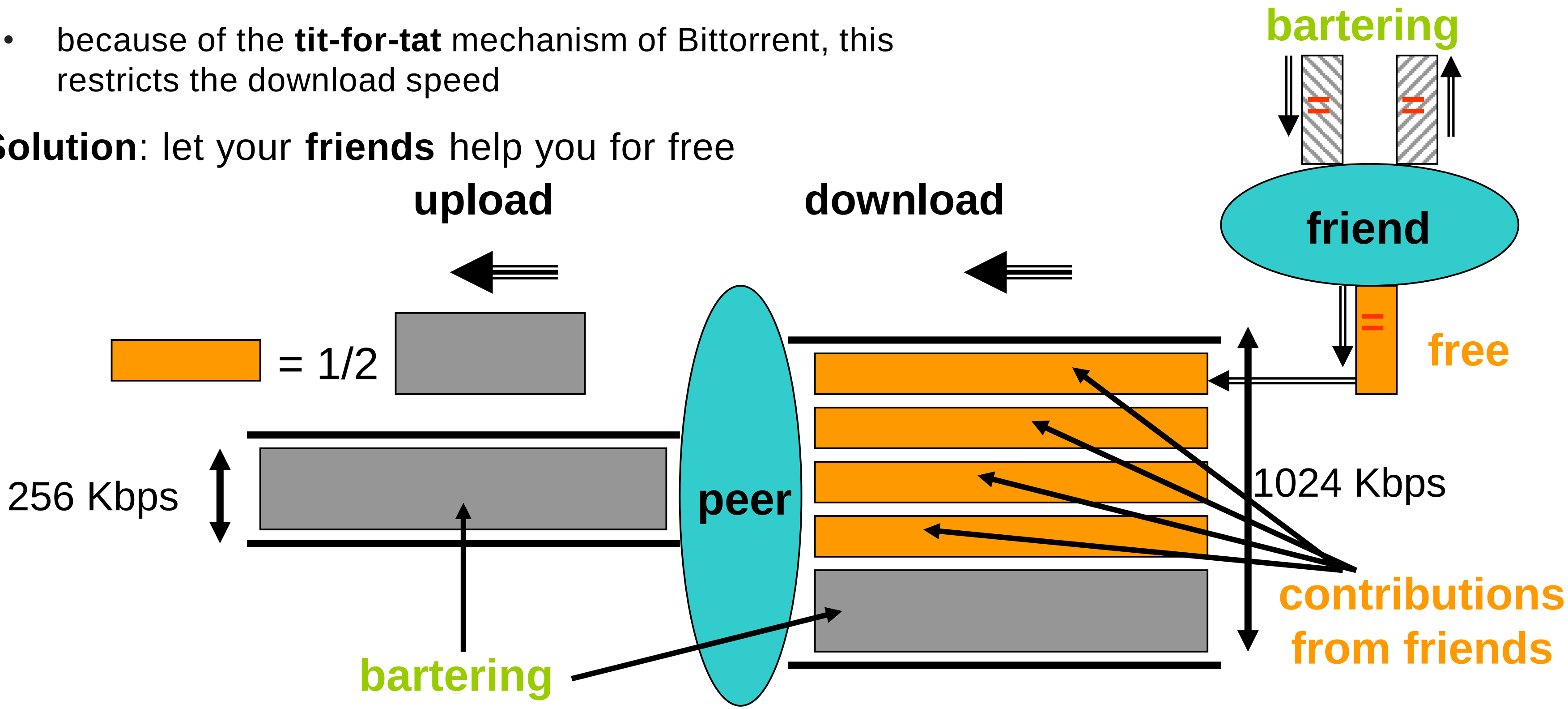
VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Cooperative downloads: basic idea

- **Problem**:
  - most users have **asymmetric** upload/download links
  - because of the **tit-for-tat** mechanism of Bittorrent, this restricts the download speed
- **Solution**: let your **friends** help you for free



VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Two protocol extensions

1. **Redundant chunks download**
   - **problem**: discrimination of helpers; more restrictive chunk selection + fewer chunks to offer, so limited bartering possibilities
   - **solution**: the same chunk may be downloaded by different helpers
2. **Sharing of swarm information**
   - **problem**: slow start; finding suitable bartering partners takes time
   - **solution**: collaborating peers exchange information on other peers in the swarm

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Download speed-up

- Every helper **equally splits its upload capacity** between bartering and helping the collector
- So **every additional helper** increases the download speedup of the collector by 0.5, up to a point
- The **maximum number of useful helpers** (and so the maximum speedup) can easily be computed

- N, S: the numbers of **leechers** and **seeders** in the system
- c, μ: the download/upload capacity of all peers
- **Download bandwidth** of the collector with **h helpers**:

$$\underbrace{\frac{S}{N}\mu}_{\text{free from seeders}} + \underbrace{\mu}_{\text{from bartering}} + \underbrace{\frac{1}{2}\sum_{i=1}^{h}\left(\frac{S}{N}+1\right)\mu}_{\text{from helpers}}$$

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Experimental setup

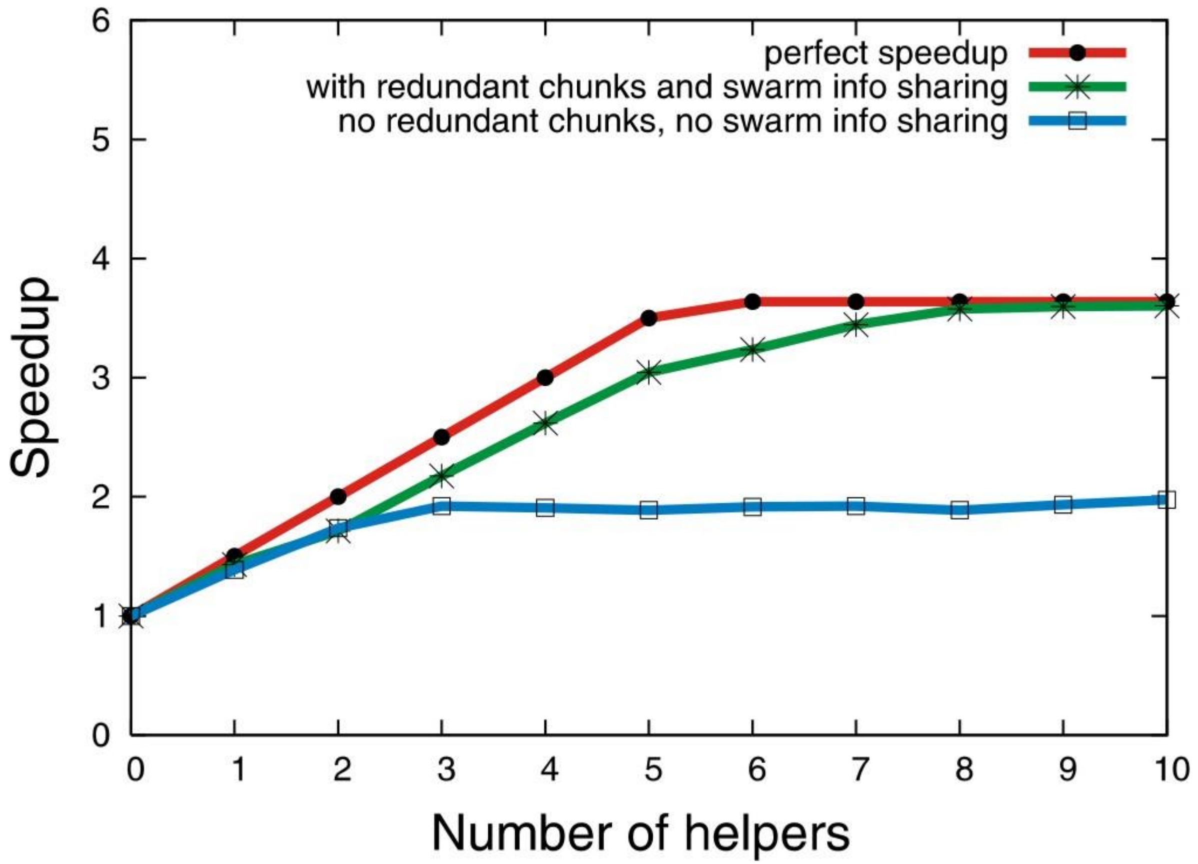
- Experiments performed in a real environment – collaborating peers connect to existing BitTorrent swarms
- Collaborating peers connected through ADSL links: 256kbps up / 1024kbps down
- Downloaded file size: 700 MB
- Swarm size: 100 leechers, 10 seeders

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

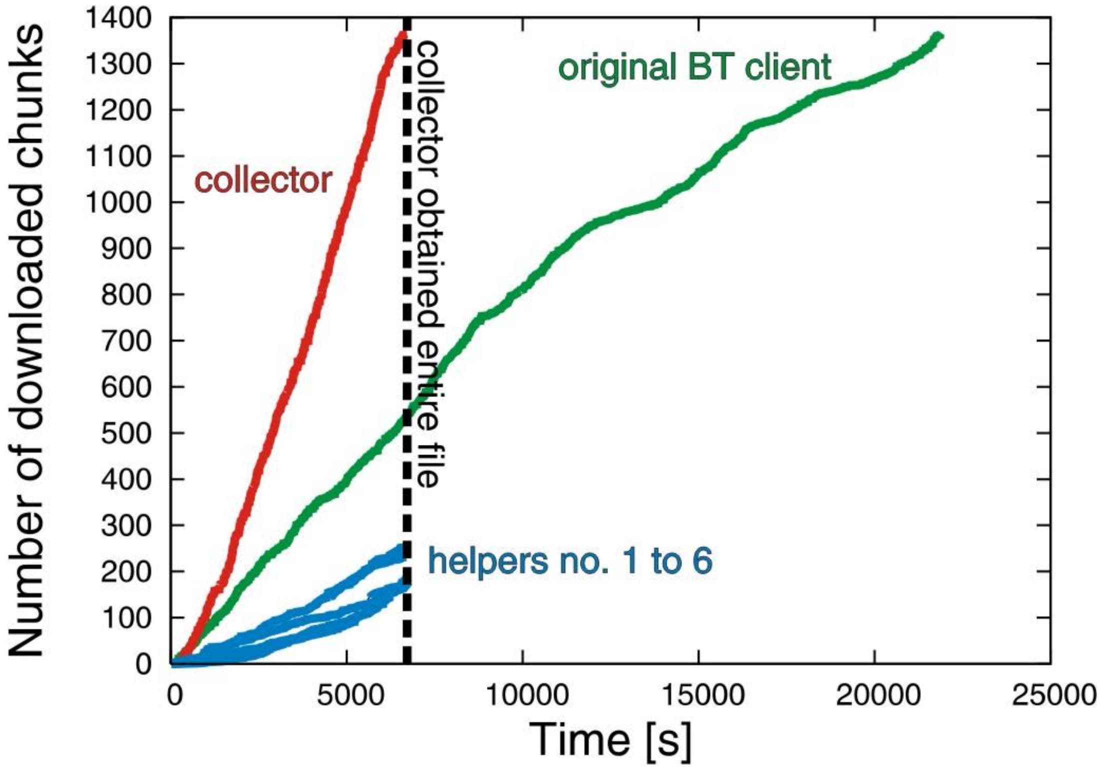
TU Delft

Speedup vs number of helpers



© 2017 Alexandru Iosup. All rights reserved.

Download progress

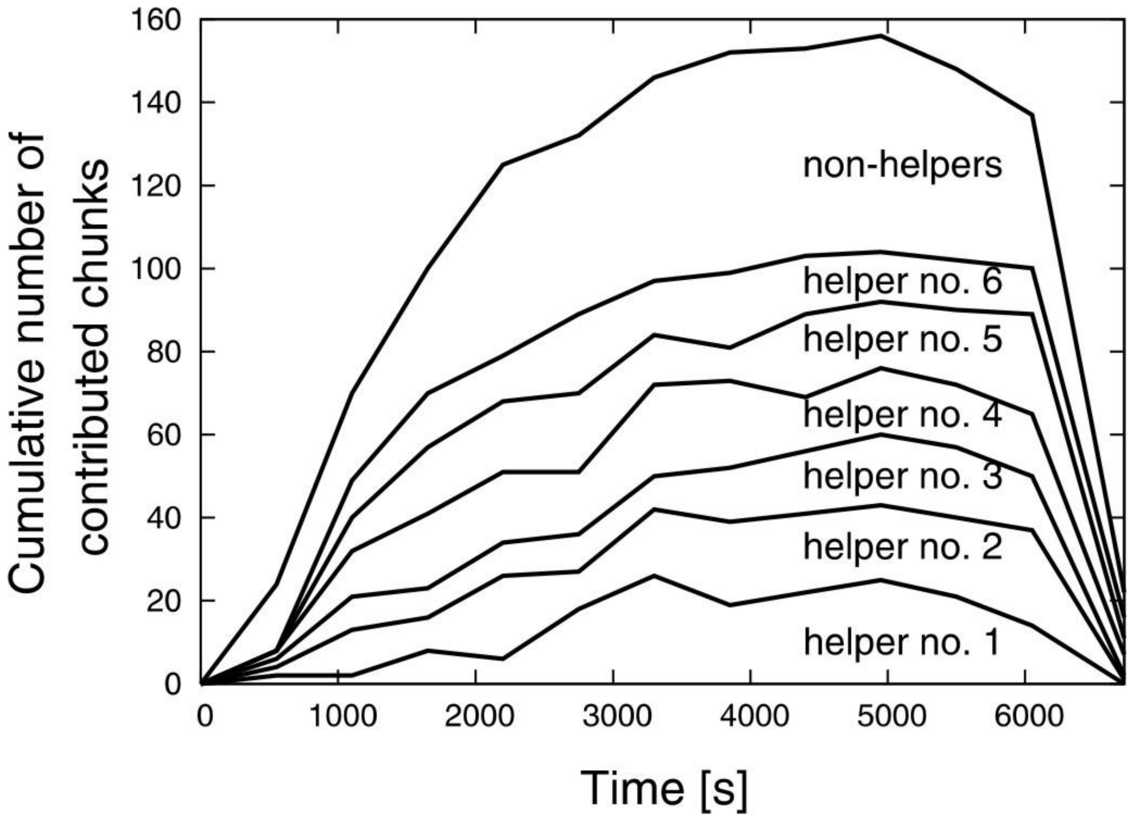


VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

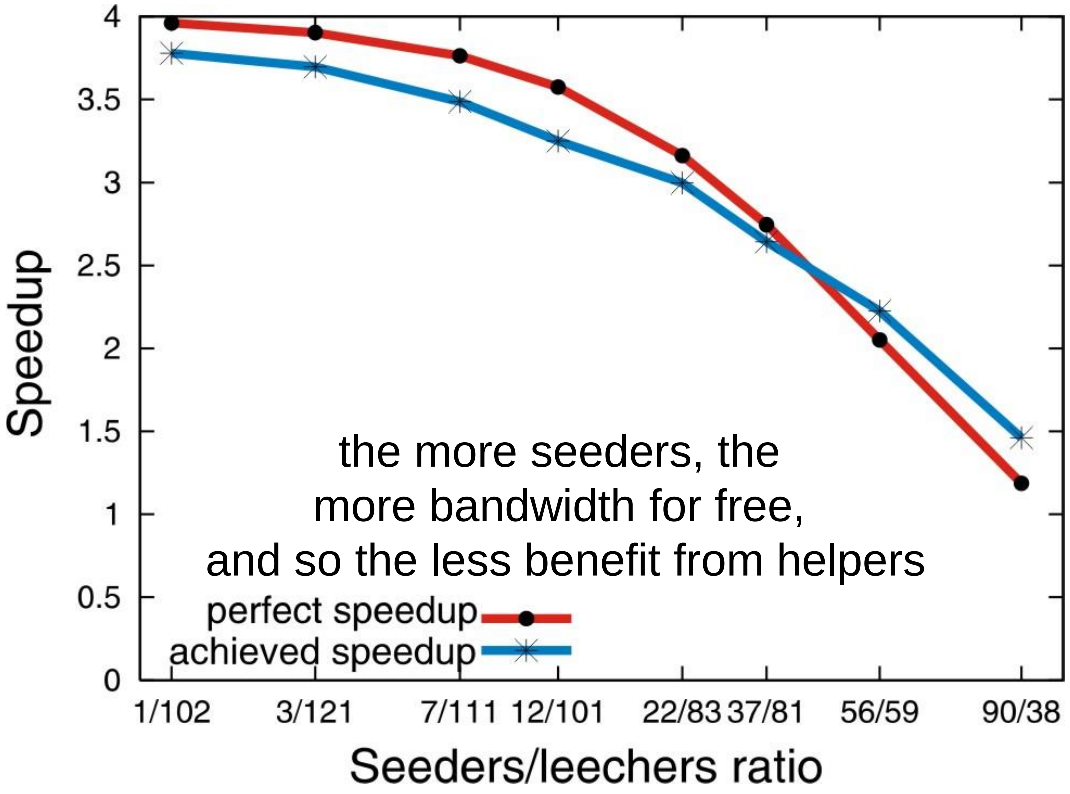
TU Delft

Helper contributions over time



© 2017 Alexandru Iosup. All rights reserved.

Speedup vs. seeders/leechers ratio



© 2017 Alexandru Iosup. All rights reserved.

Opencraft

Towards Scalable Minecraft-like Environments

Jesse Donkervliet
Jerom van der Sar
Alexandru Iosup

Contact: opencraft@atlarge-research.com
www.atlarge-research.com/opencraft



Jesse Donkervliet
Jerom van der Sar
Alexandru Iosup

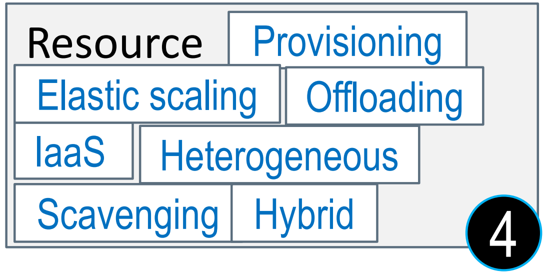
Meerkat

Dynamic Conit-based Scalability Techniques for Minecraft-like Environments

(Jun 2017)

(unpublished, so please do not record or share)

4. Resource Management and Scheduling



- Systems
  - Cycle Scavenging in Koala
  - Mirror Offloading in OpenTTD
- Design, Implementation, Deployment, and Testing of …
  - Elastic mechanisms and policies
  - IaaS provisioning and allocation policies
  - Cycle scavenging mechanisms and policies
  - Heterogeneous and hybrid resource management
  - Offloading architectures, mechanisms, and policies

© 2017 Alexandru Iosup. All rights reserved.

David Villegas
FIU/IBM

Athanasios
Antoniou

Alexandru
Iosup

Dick
Epema

IaaS Provisioning and Allocation

Design of new policies and real-world experiments to compare with alternatives

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012.

Provisioning and Allocation Policies*

* For User-Level Scheduling

- Provisioning

Policy	Class	Trigger	Adaptive
Startup	Static	–	–
OnDemand	Dynamic	QueueSize	No
ExecTime	Dynamic	Exec.Time	Yes
ExecAvg	Dynamic	Exec.Time	Yes
ExecKN	Dynamic	Exec.Time	Yes
QueueWait	Dynamic	Wait Time	Yes

- Allocation

Policy	Queue-based	Known job durations
FCFS	Yes	No
FCFS-NW	No	No
SJF	Yes	Yes

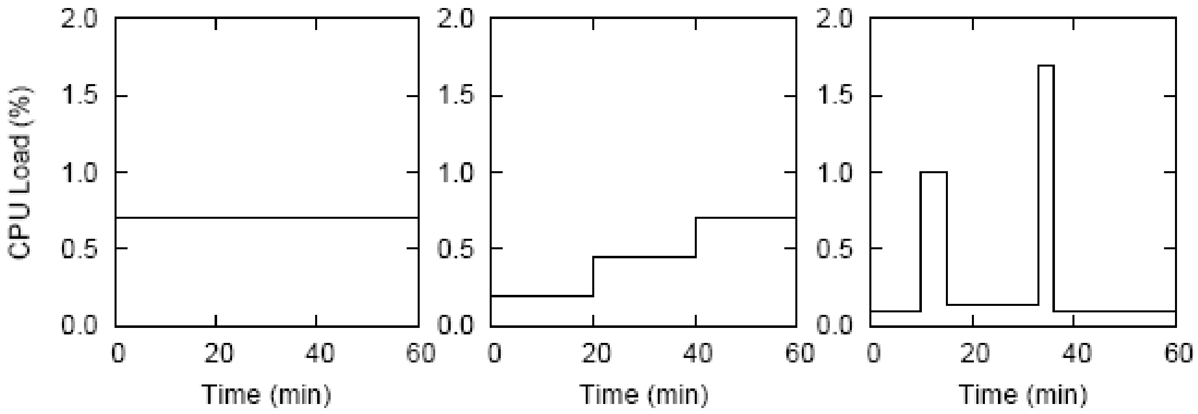
- Also looked at combined Provisioning + Allocation policies

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012.

Experimental Setup (1)

- Environments
  - DAS4,
    Florida International University (FIU)
  - Amazon EC2
- Workloads
  - Bottleneck
  - Arrival pattern

Workload Unit	CPU	Memory	I/O	Appears in
WU1	X			WL1
WU2		X		WL2,WL4
WU3			X	WL3,WL4



Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid2012 + PDS Tech.Rep.2011-009

Experimental Setup (2)

- **Performance Metrics**
  - Traditional: Makespan, Job Slowdown
  - Workload Speedup One (SU1)
  - Workload Slowdown Infinite (SUinf)

$$SU_1(W) = \frac{MS(W)}{\sum_{i \in W} t_R(i)}$$

$$SU_\infty(W) = \frac{MS(W)}{\max_{i \in W} t_R(i)}$$

- **Cost Metrics**
  - Actual Cost (Ca)
  - Charged Cost (Cc)

$$C_a(W) = \sum_{i \in leased\ VMs} t_{stop}(i) - t_{start}(i)$$

$$C_c(W) = \sum_{i \in leased\ VMs} \lceil t_{stop}(i) - t_{start}(i) \rceil$$

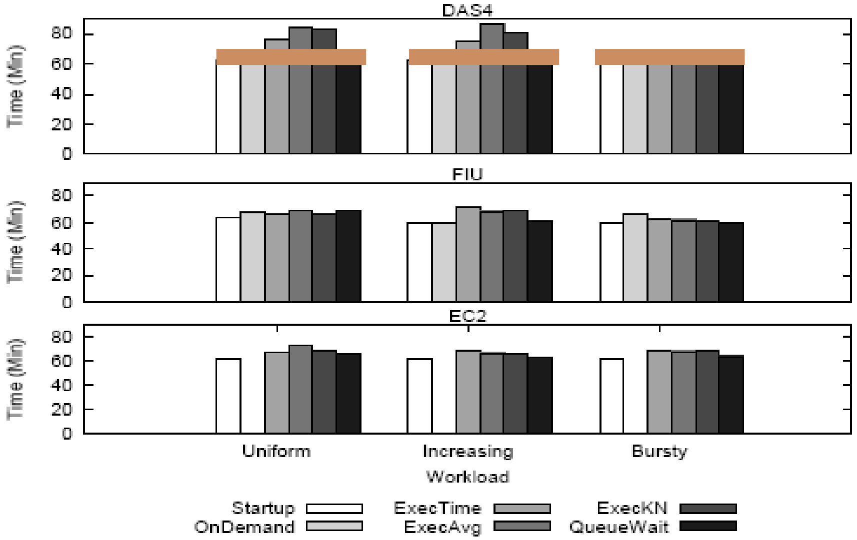
- **Compound Metrics**
  - Cost Efficiency (Ceff)
  - Utility

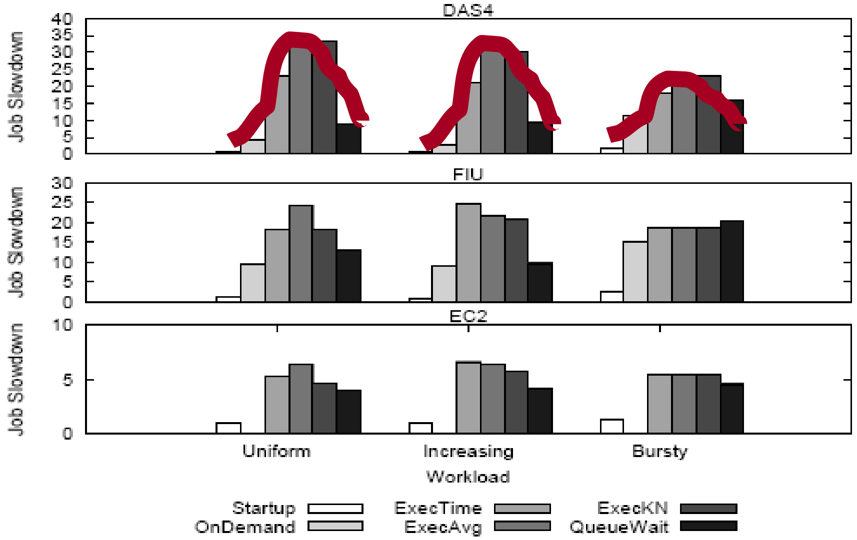
$$C_{eff}(W) = \frac{C_c(W)}{C_a(W)}$$

$$U(W) = \frac{SU_1(W)}{C_c(W)}$$

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Performance Metrics





- Makespan very similar
- Very different job slowdown

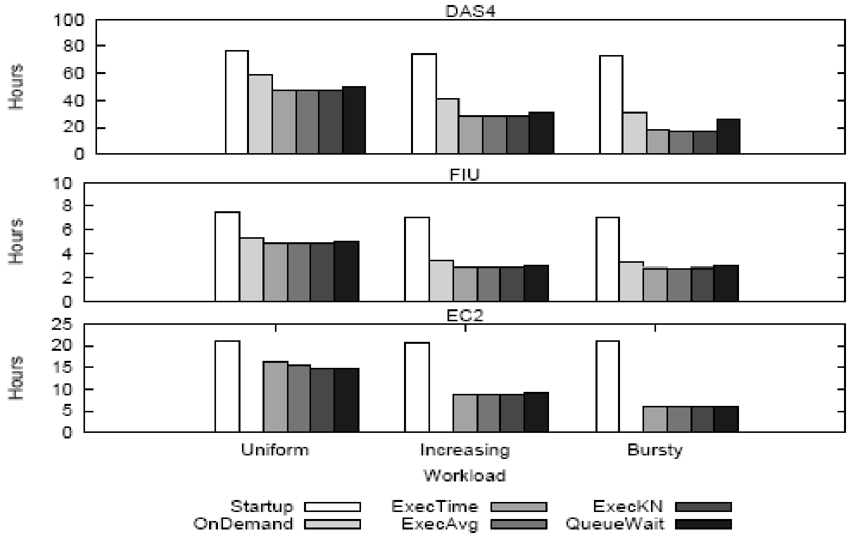
VU VRIJE UNIVERSITEIT AMSTERDAM

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

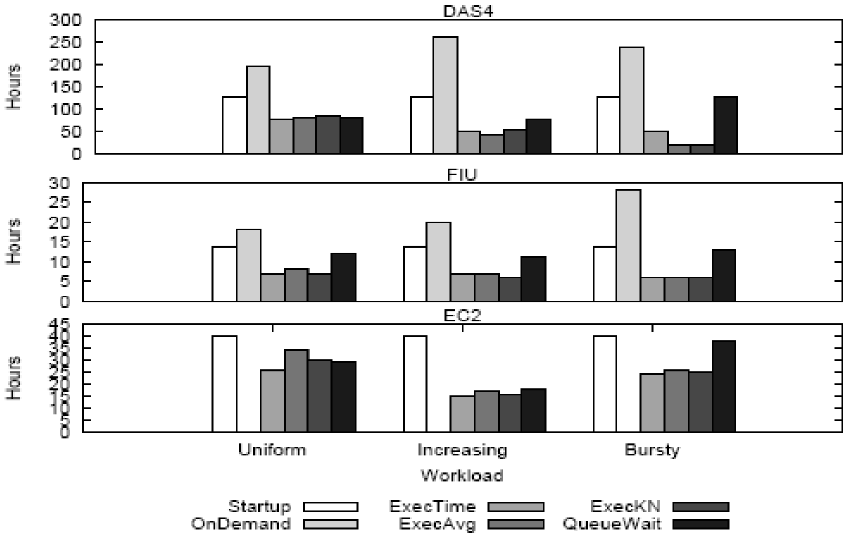
Delft

Cost Metrics

Actual Cost



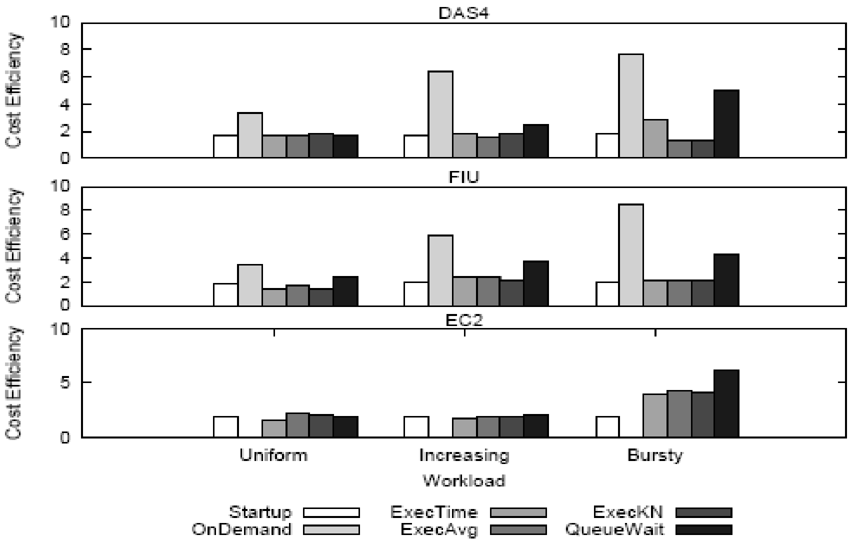
Charged Cost

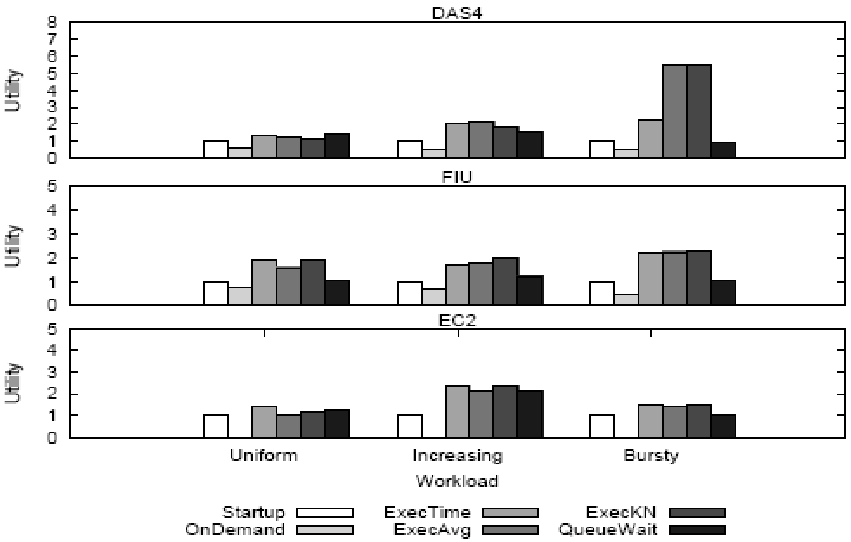


- Very different results between actual and charged
  - Cloud charging function an important selection criterion
- All policies better than Startup in actual cost
- Policies much better/worse than Startup in charged cost

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

Compound Metrics





- Trade-off Utility-Cost still needs investigation
- **Performance or Cost, not both:**
  the policies we have studied improve one, but not both

Villegas, Antoniou, Sadjadi, Iosup. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds, CCGrid 2012

VRIJE UNIVERSITEIT AMSTERDAM

Delft

Omer Ozan Sönmez

Alexandru Iosup

Dick Epema

Cycle Scavenging in Koala

Scheduling Strategies for Cycle Scavenging in Multicluster Grid Systems

Sonmez, Grundeken, Mohamed, Iosup, and Epema. Scheduling Strategies for Cycle Scavenging in Multicluster Grid Systems, CCGRID 2009.

KOALA: a co-allocating grid scheduler

Mohamed and Epema. KOALA: a co-allocating grid scheduler. CCPE 20(16): 1851-1876 (2008)

Original goals:

1. processor co-allocation - parallel applications.
2. data co-allocation - job affinity based on data locations.
3. load sharing - in the absence of co-allocation.

while being transparent for local schedulers

Additional goals:

4. Research vehicle for grid and cloud research.
5. Support for (other) popular application types.

Written in **Java, middleware independent** (initially Globus-based).

Has been deployed on the DAS2 - DAS4 (soon on DAS-5) since 2005.

© 2017 Alexandru Iosup. All rights reserved.

KOALA: the runners

The KOALA **runners** are **adaptation modules** for different application types:

- Set up communi
- Launch applicat
- Scheduling polic

Current runners

- **CSRunner**:
- **IRunner**:
- **Mrunner**:
- **OMRunner**: for **co-allocated parallel** OpenMPI applications
- **Wrunner**: for **co-allocated workflows**
- **MR-runner**: for **MapReduce** applications

Conclusion:

Very beneficial to have a deployed research vehicle (DAS4 + KOALA) for:

- driving research
- doing experimentation
- visibility

Sonmez, Mohamed, and Epema. On the Benefit of Processor Coallocation in Multicluster Grid Systems. IEEE TPDS 21(6): 778-789 (2010)

Cycle Scavenging in Koala (1/3): System Requirements

1. **Unobtrusiveness**

   Minimal delay for (higher priority) local and grid jobs

2. **Fairness**

   Multiple cycle scavenging applications running concurrently should be assigned comparable CPU-Time

3. **Dynamic Resource Allocation**

   Cycle scavenging applications have to Grow/Shrink at runtime

4. **Efficiency**

   As much use of dynamic resources as possible

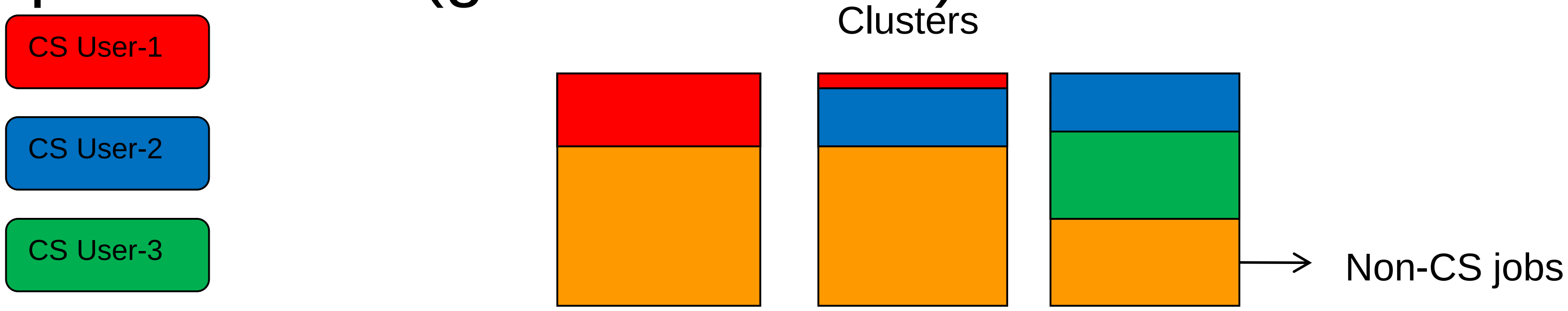
5. **Robustness and Fault Tolerance**

   Long-running, complex system: problems will occur, and must be dealt with

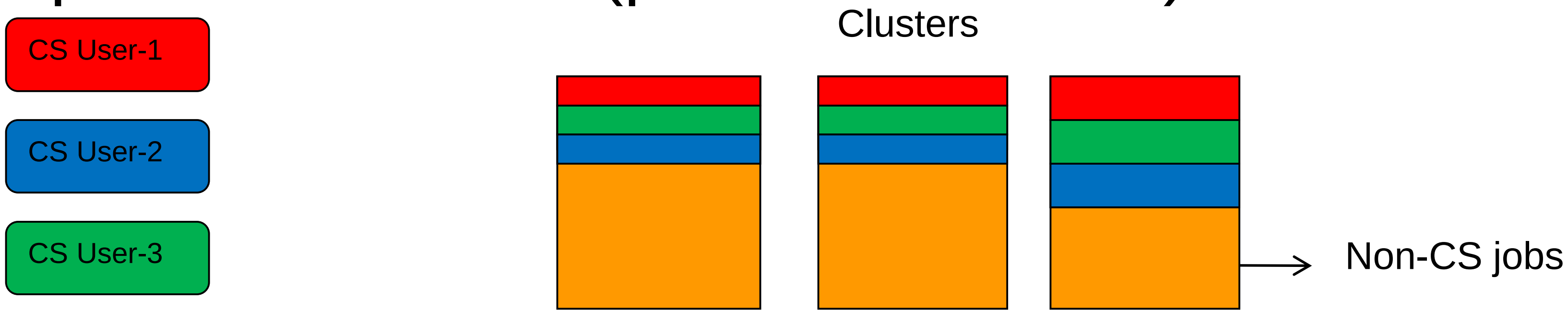
O.O. Sonmez, B. Grundeken, H.H. Mohamed, A. Iosup, and D.H.J. Epema, “Scheduling Strategies for Cycle Scavenging in Multicluster Grid Systems," *9th IEEE/ACM Int'l Symposium on Cluster Computing and the Grid (CCGRID09)*, May 2009.

Cycle Scavenging in Koala (2): Policies

1. Equipartition-All (grid-wide basis)



2. Equipartition-PerSite (per-cluster basis)

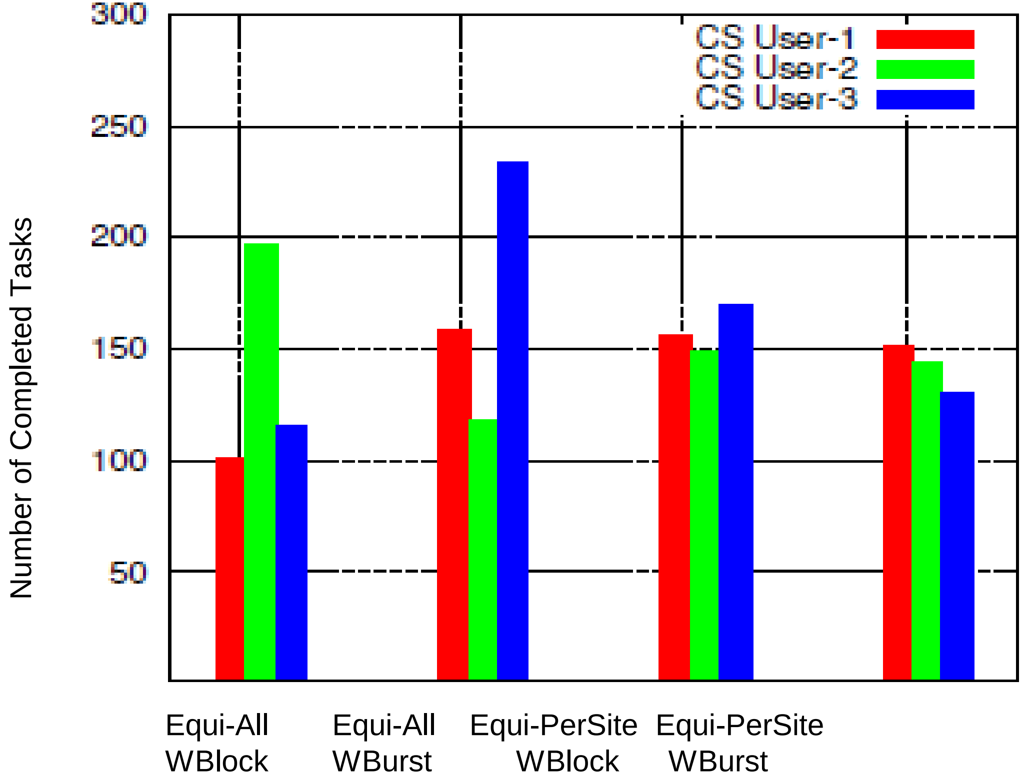


VU VRIJE UNIVERSITEIT AMSTERDAM

O.O. Sonmez, B. Grundeken, H.H. Mohamed, A. Iosup, and D.H.J. Epema, "Scheduling Strategies for Cycle Scavenging in Multicluster Grid Systems," *9th IEEE/ACM Int'l Symposium on Cluster Computing and the Grid (CCGRID09)*, May 2009.

TU Delft

Cycle Scavenging in Koala (3): Experimental Results



Equi-PerSite is fair and superior to Equi-All

VRIJE UNIVERSITEIT AMSTERDAM

O.O. Sonmez, B. Grundeken, H.H. Mohamed, A. Iosup, and D.H.J. Epema, “Scheduling Strategies for Cycle Scavenging in Multicluster Grid Systems," *9th IEEE/ACM Int'l Symposium on Cluster Computing and the Grid (CCGRID09)*, May 2009.

TU Delft

Alexandru
Iosup

Otto
Visser

Wishnu
Prasetya

Minghai
Jiang

Mirror

A Mirroring Architecture for Sophisticated Mobile Games using Computation-Offloading

(Sep 2017)

(unpublished, so please do not record or share)

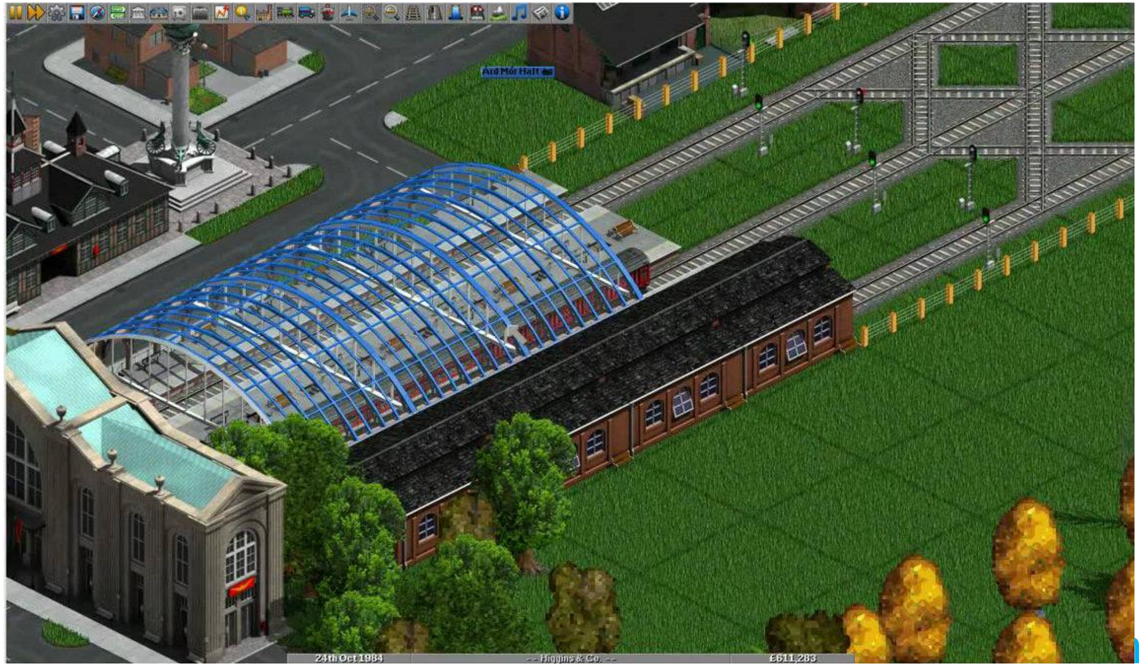
Bringing a Classic to the 21st Century

Chris Sawyer's Transport Tycoon — TTD@Win95 — OpenTTD — OpenTTD@large

1994 | 1995 | 1996 | 1997 | 2003 | 2007 | 2011 | 2014 | 2017

Transport Tycoon Deluxe: climate, better signals — Jeff Drexler´s TTDPatch++, gfx++ — OpenTTD+ AIs — Android — OpenTTD +Mirror

OpenTTD +Social Extensions



dru Iosup. All rights reserved.

OpenTTD: Open-Source Life to Transport Tycoon Deluxe ~300k players

- Replaced
  - GFX, SFX, Music
  - Non-cheating AI
  - AI VM + API (Squirrel~Lua)
- Added or improved
  - DLC: mods/maps/AIs
  - Pathfinding, train signal system, vehicle handling
  - Multiplayer
  - Too many to mention

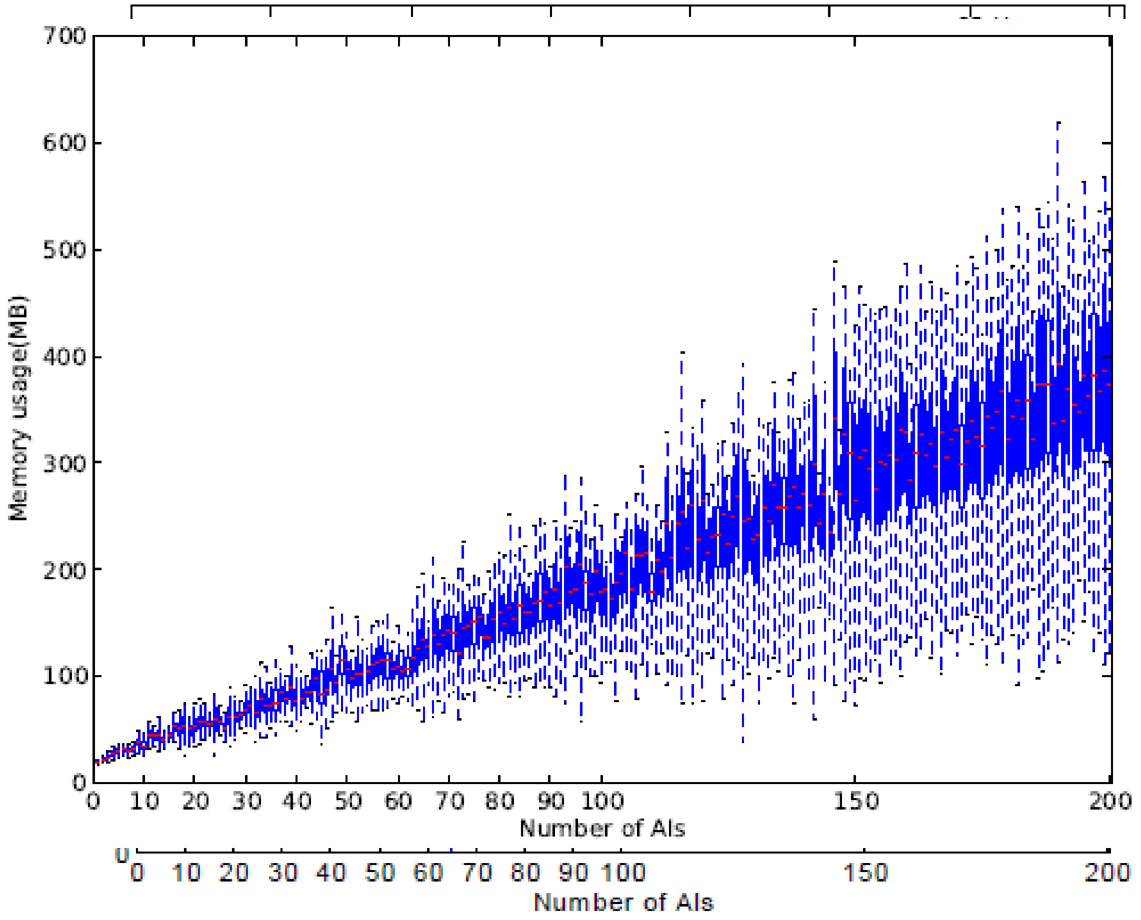
- Tech limitations
  - Max. 15 players (255 if cooperating, rare)
  - Max. map size $2k^2$
  - Scalable tech?
- Design limitations
  - Limited variety
  - No social
  - Scalable design?

© 2017 Alexandru Iosup. All rights reserved.

OpenTTD: Some Tech Limitations

- Network instability
- CPU overload
- Memory instability

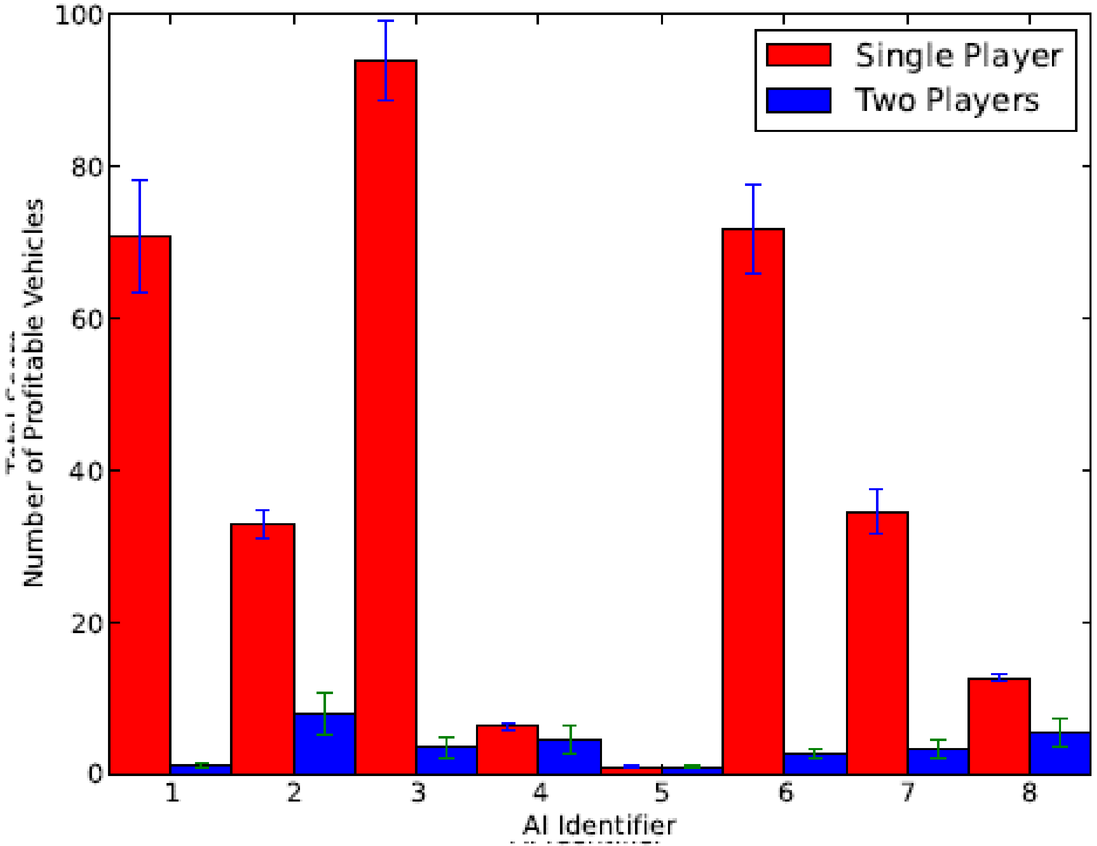
- So … 15 players in one game



Shen, Visser, Iosup: RTSenv: An experimental environment for real-time strategy games. NETGAMES 2011

OpenTTD: Some Design Limitations

- # profitable vehicles
- Complex to configure (e.g., AI selection)
- Free-riding AIs so far unbeatable (dominant strategy)
  - Our leading AI Rondje om de Kerk does this



Shen, Visser, Iosup: RTSenv: An experimental environment for real-time strategy games. NETGAMES 2011

OpenTTD@large: Massivizing OpenTTD

- Tech
  - Automatic scaling of server capacity
  - Single-map scalability enhancements
  - Gaming analytics engine
- Design
  - Unlimited map size
  - Unlimited amount of players
  - Support both casual and hardcore gamers
  - Add social aspects (like guilds and achievements)

http://squarefaction.ru/files/game/715/gallery/97213dfa302b09582f482c2138475632.png

http://bfewaw.com/showthread.php?t=272066

Need co-scalability of game platform and design!

© 2017 Alexandru Iosup. All rights reserved.

OpenTTD@large: Game Modes (for unlimited map size, # players)

- Quick game
  - Think of a 15 minute lunch break game
- Normal game
  - A few hours; much like current OpenTTD
- Challenge mode
  - Accomplish a certain feat, to unlock technology
- Unlimited (new)
  - Unlimited size or players, only unlocked technology and your own little square on the map

© 2017 Alexandru Iosup. All rights reserved.

OpenTTD@large: One Social Aspect
The Neighbor Interaction [1/2]

- **A new way to interact with others in OpenTTD**
- Scenario: A map can have wood, but no sawmills. Need exchange mechanism to keep economy running.
- Mechanism elements:
  - Players can build “trade centers” at the map edges
  - Players can suggest “international” trades (e.g.: oil at
  - The neighbouring map player(s) accept (or not)
  - Price and volume are negotiable
  - Play with currency exchange rate if needed

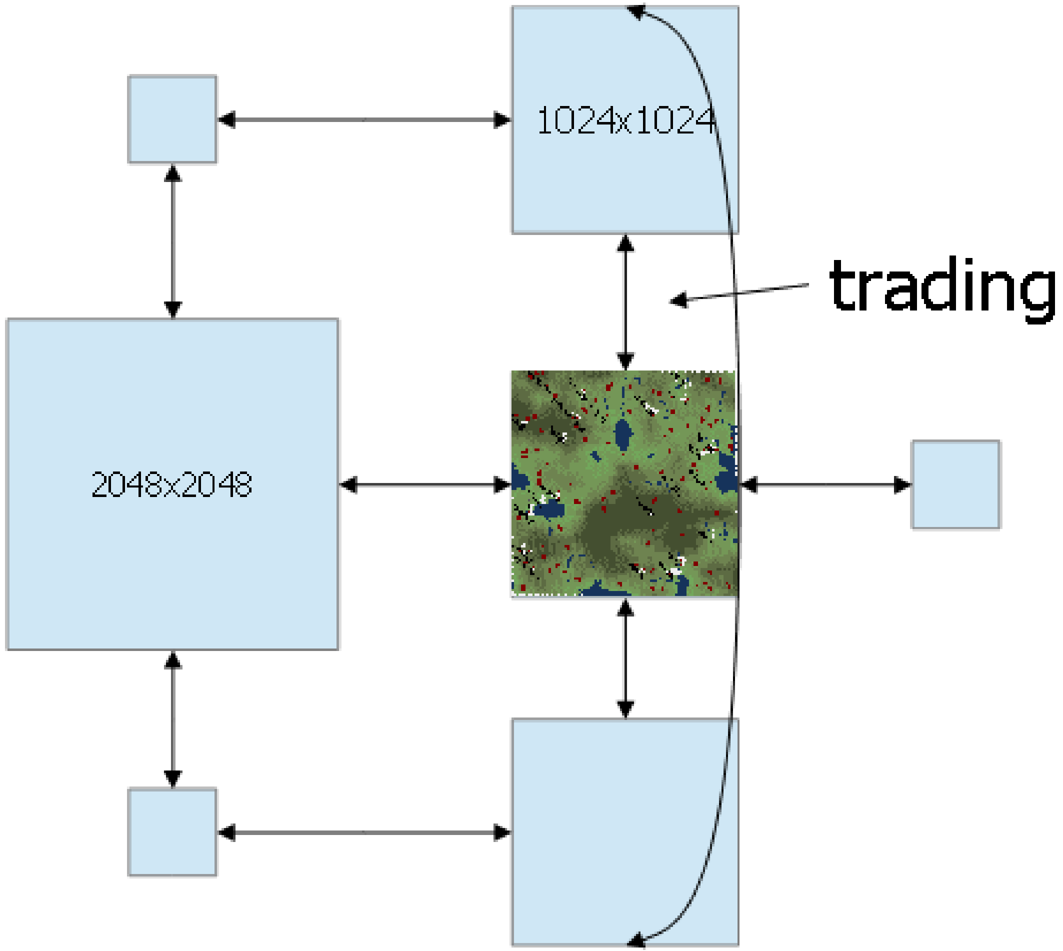
VRIJE UNIVERSITEIT AMSTERDAM

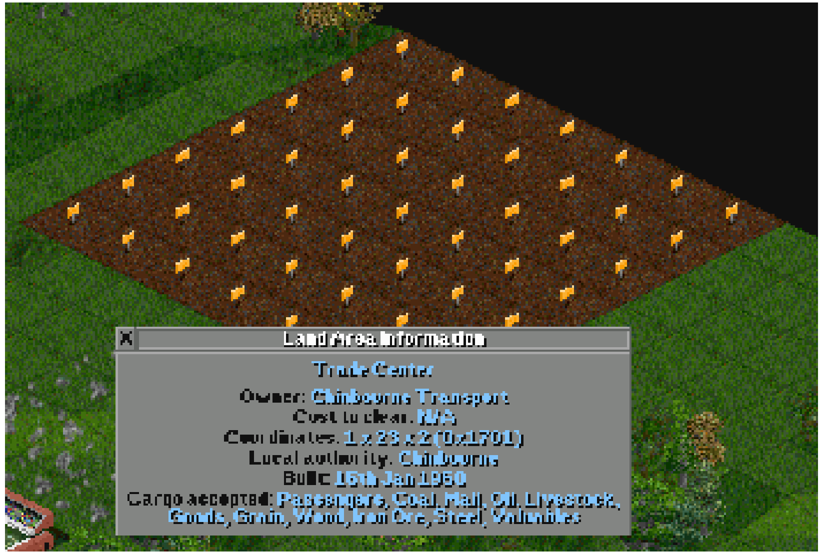
© 2017 Alexandru Iosup. All rights reserved.

TUDelft

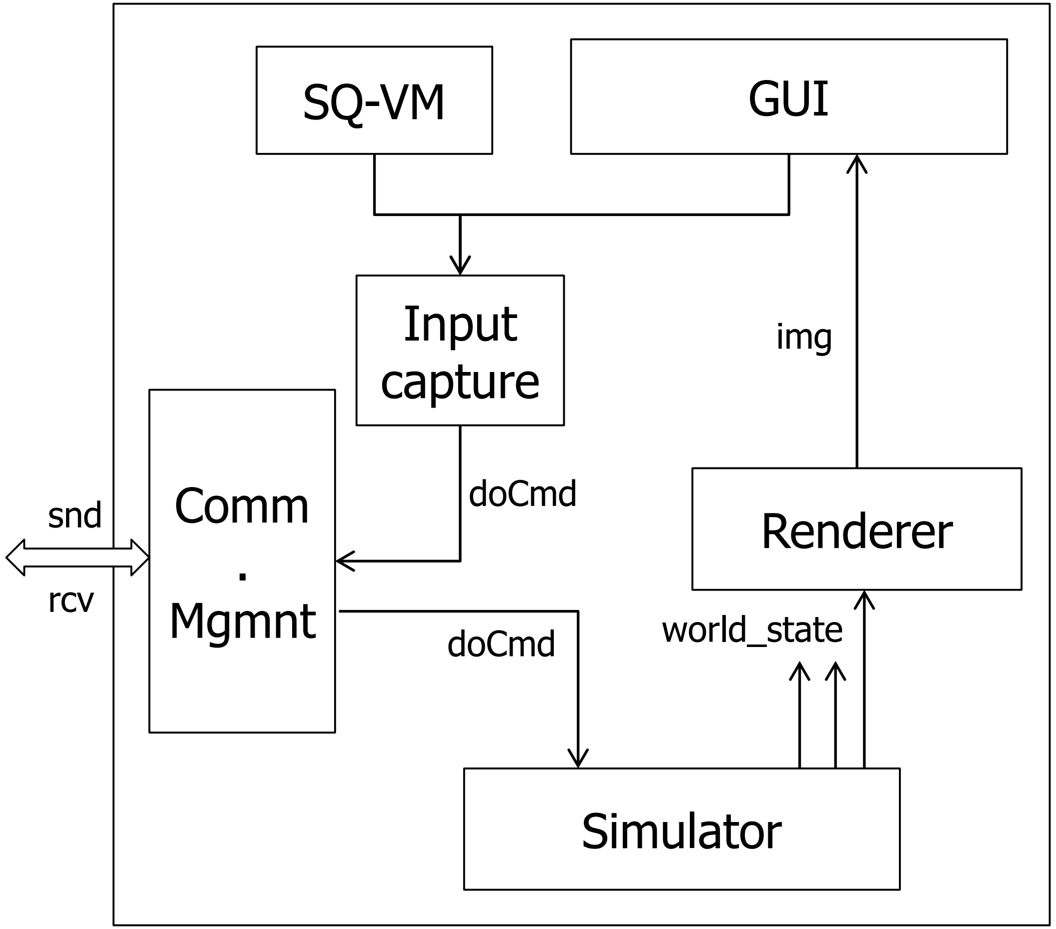
OpenTTD@large: One Social Aspect
The Neighbor Interaction [2/2]

- Players can build “trade centers” at the map edges





An Offloading Use Case: The OpenTTD Client

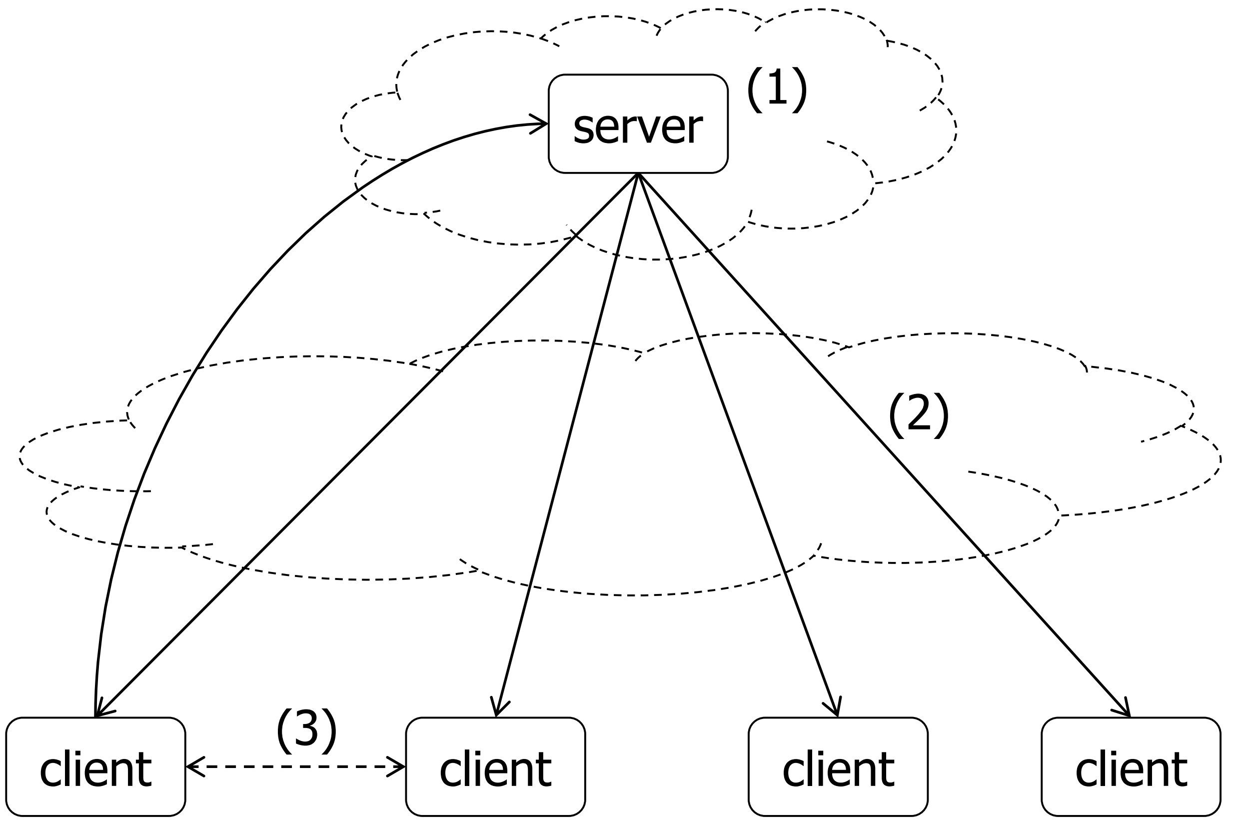


Game Parameters:

- map size
- number of players
- number of cities
- number of resources
- animations on/off

© 2017 Alexandru Iosup. All rights reserved.

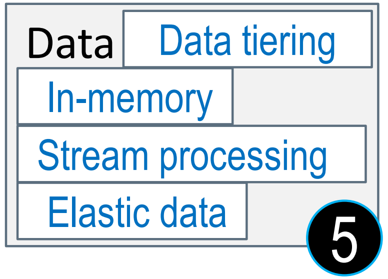
Many Cloud Offloading Alternatives



Cases to investigate:
1. server in cloud
2. server behind cloud
3. clients in same LAN
4. hybrid

VU VRIJE UNIVERSITEIT AMSTERDAM
© 2017 Alexandru Iosup. All rights reserved.
TU Delft

5. Data Management and Scheduling



- Systems
  - Fawkes
  - MemFS (MemEFS, MemEEFS)
  - HyGraph
  - JoyGraph
- Design, Implementation, Deployment, and Testing of
  - Elastic data processing architectures, mechanisms, and policies
  - In-memory architectures, mechanisms, and policies
  - Stream processing of graphs with data-partition management
  - Distributed, heterogeneous and hybrid, graph processing

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Bogdan
Ghiț

Nezih
Yigĭtbası

Alexandru
Iosup

Dick
Epema

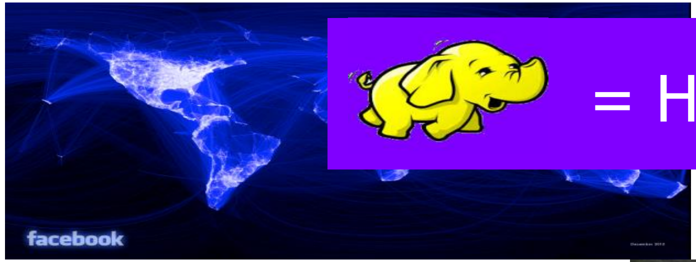
Fawkes

Balanced Resource Allocations Across Multiple Dynamic MapReduce Clusters

Ghit, Yigitbasi, Iosup, Epema. Balanced resource allocations across multiple dynamic MapReduce clusters. SIGMETRICS 2014: 329-341

The “Big Cake” Challenge In the Datacenter

Online Social Networks



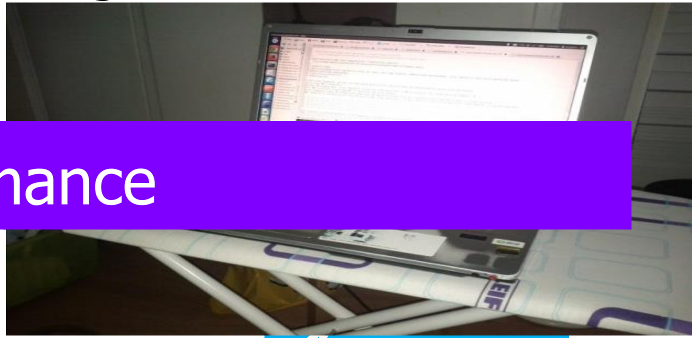
Financial Analysts

= Hadoop / MapReduce framework

Need multi-tenant, self-aware schedulers and resource managers

Universe Explorers

Big Data Enthusiast



Multiple frameworks = Isolation, especially performance

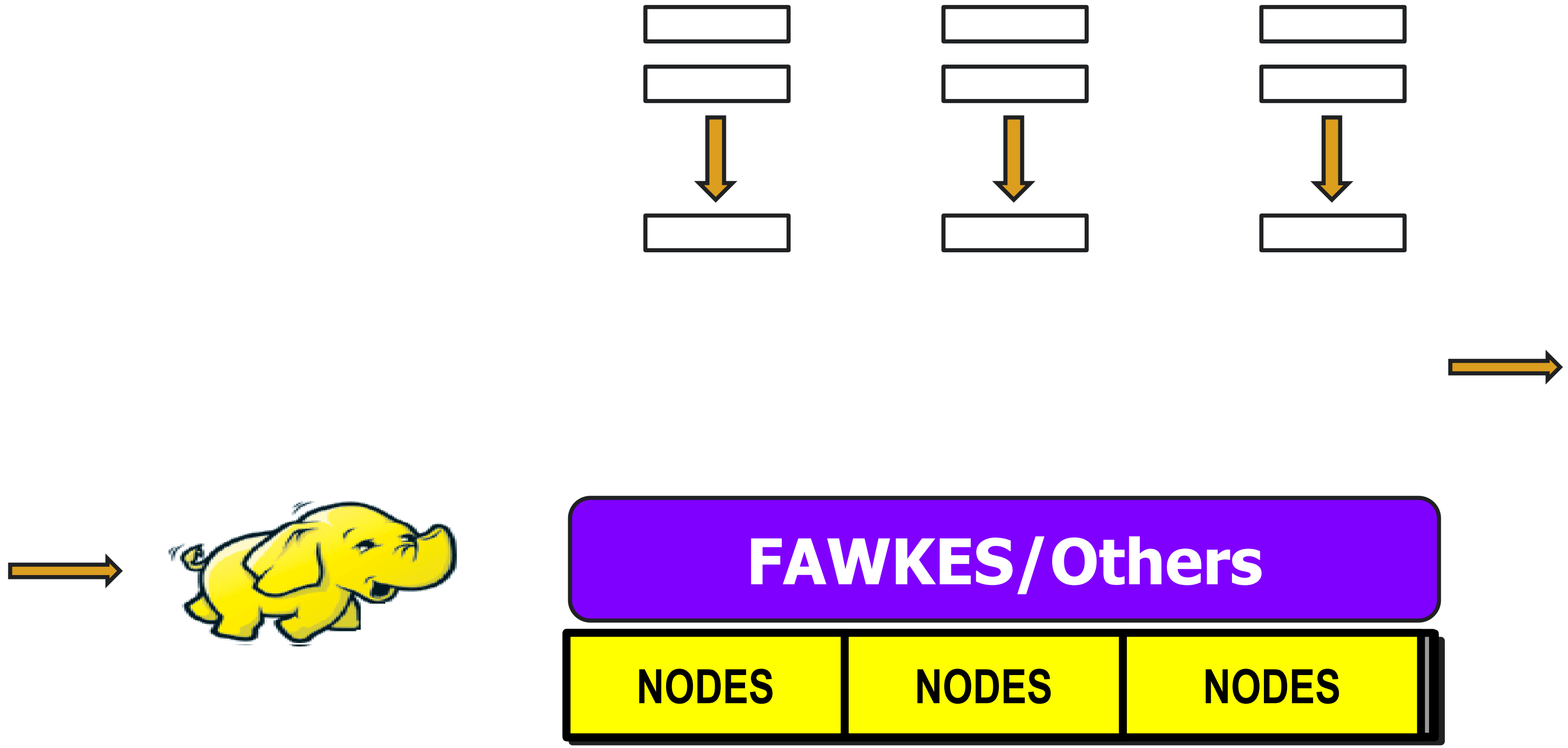
VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Dynamic Big Data Processing

Fawkes = Elastic MapReduce



FAWKES

Job submissions

Frameworks

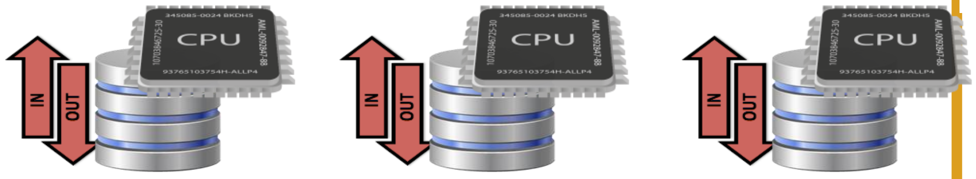
Resource manager

Infrastructure

B. Ghit et al. Balanced Resource Allocations Across Multiple Dynamic MapReduce Clusters. SIGMETRICS 2014

Elasticity for MapReduce Frameworrks

Core nodes



INPUT/OUTPUT DATA

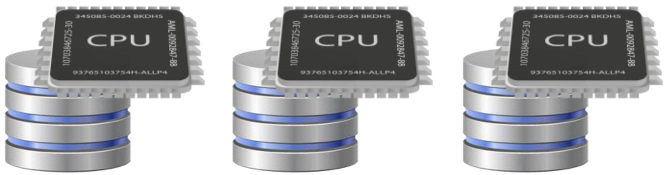
- Classical deployment
- Uniform data distribution
- **No removal**

Transient nodes (TR)

NO DATA

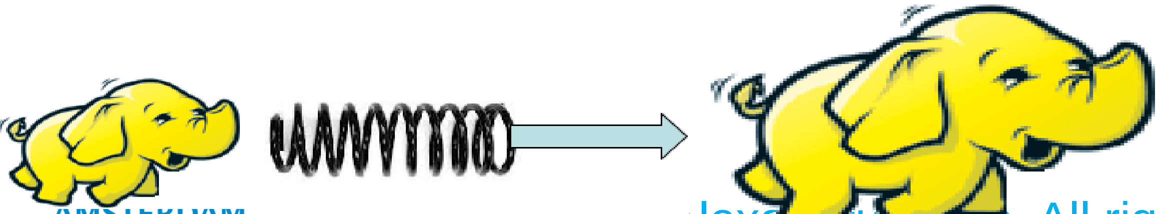
- No local storage
- R/W from/to core nodes
- **Instant removal**

Trans-core nodes (TC)



OUTPUT DATA

- Local storage, no input
- Only R from core nodes
- **Delayed removal**

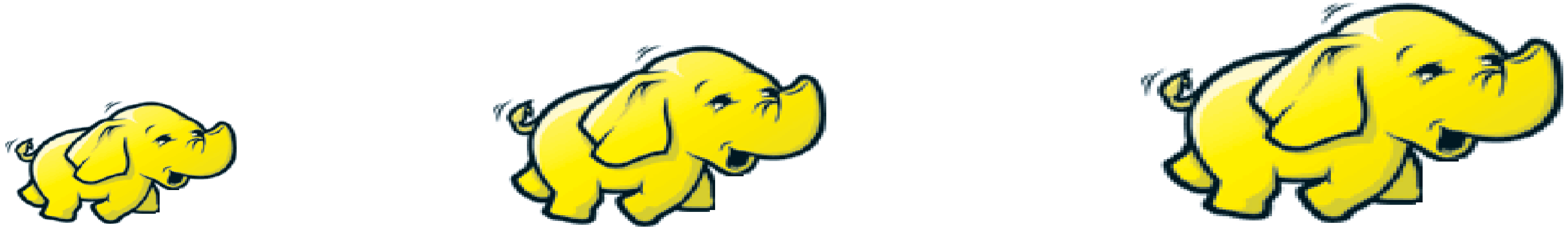


lexa ru sup. All righ

Fawkes in a Nutshell [1/2]

Because workloads may be time-varying:

- Poor resource utilization
- Imbalanced service levels



$w_1 < w_2 < w_3$

1. Fair framework size:

$$s_i = \frac{w_i}{w_1 + w_2 + w_3}, \quad i = 1,2,3$$

VU VRIJE UNIVERSITEIT AMSTERDAM

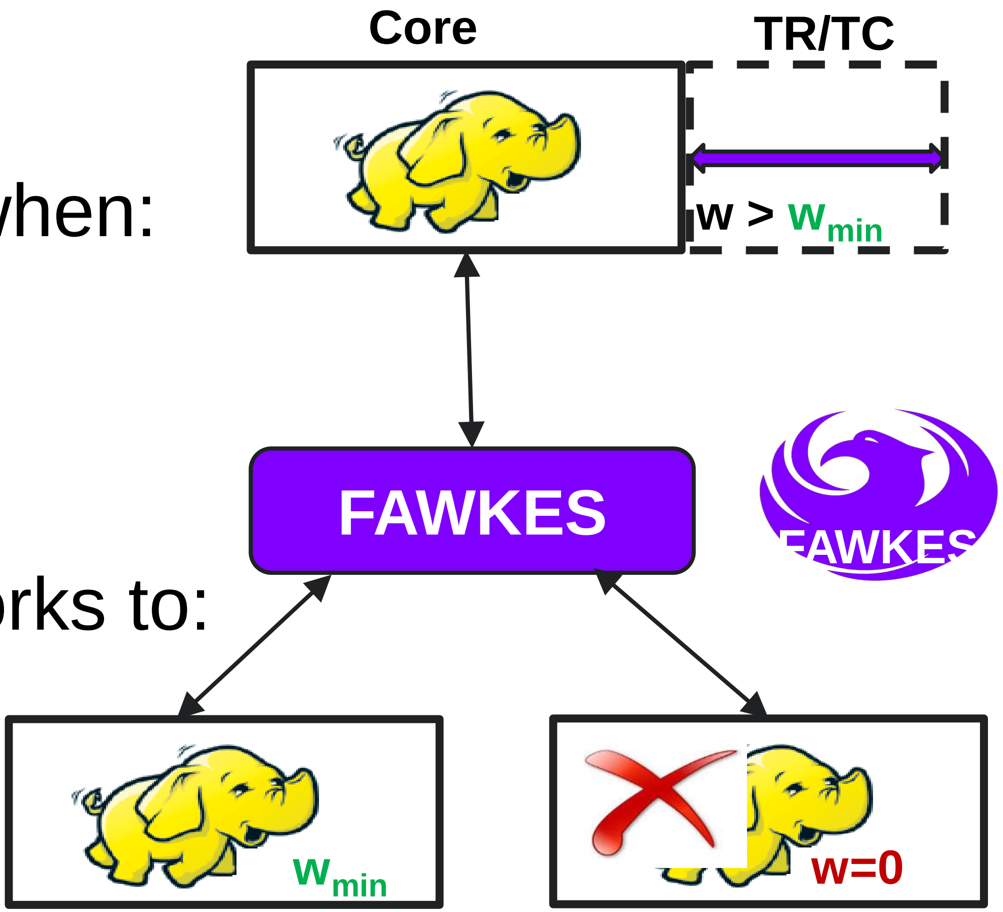
© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Fawkes in a Nutshell [2/2]

2. Updates dynamic weights when:
  - New frameworks arrive
  - Framework states change

3. Shrinks and grows frameworks to:
  - Allocate new frameworks
  - Give fair shares to existing frameworks
  - Eliminate unused frameworks

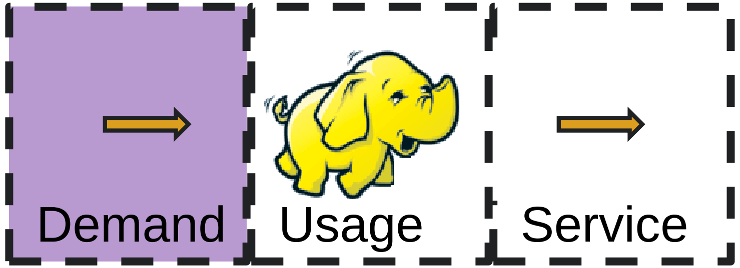


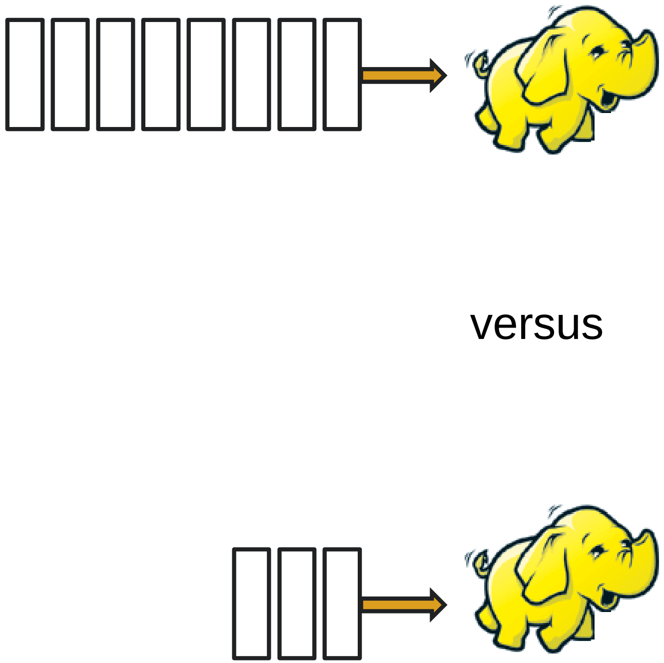
VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

How to differentiate frameworks (1/3)



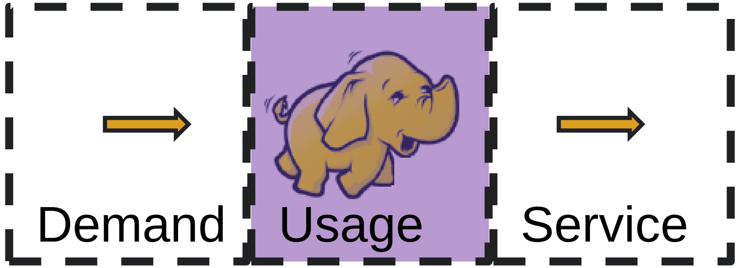


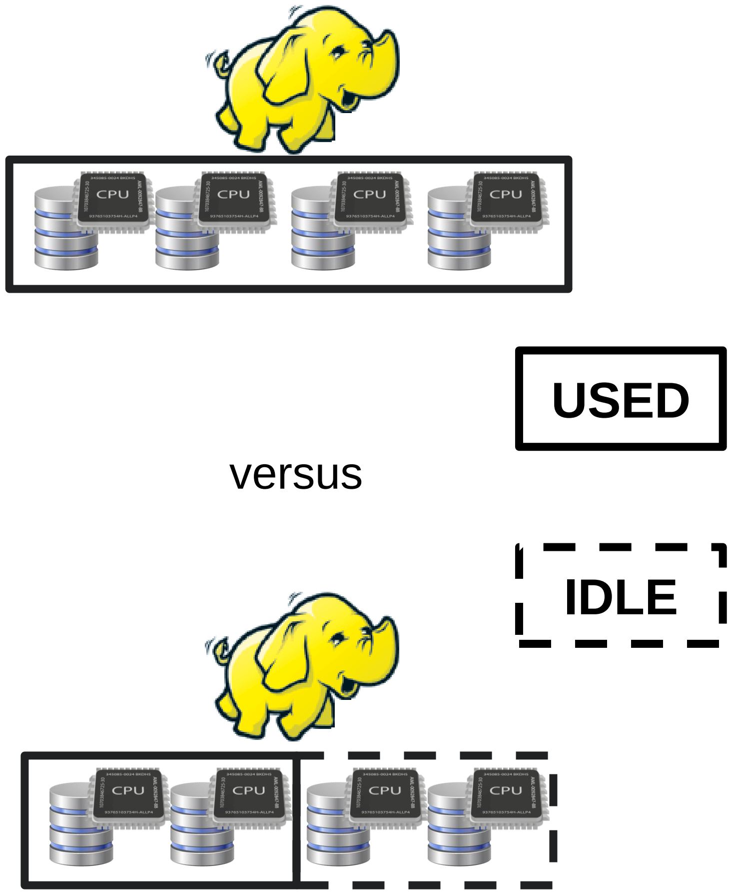
By demand – 3 policies:

- Job Demand (JD)
- Data Demand (DD)
- Task Demand (TD)

© 2017 Alexandru Iosup. All rights reserved.

How to differentiate frameworks (2/3)



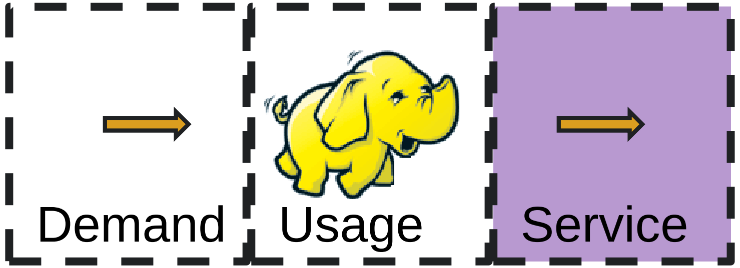


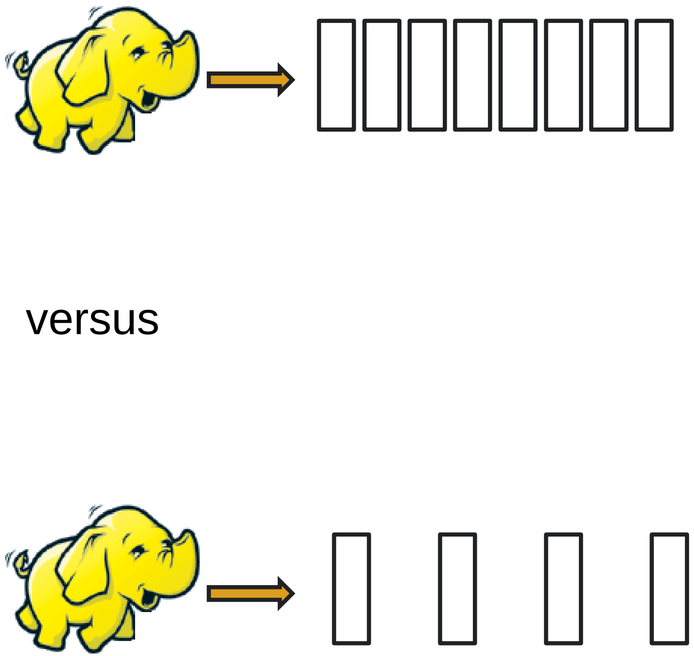
By usage – 3 policies:

- Processor Usage (PU)
- Disk Usage (DU)
- Resource Usage (RU)

© 2017 Alexandru Iosup. All rights reserved.

How to differentiate frameworks (3/3)





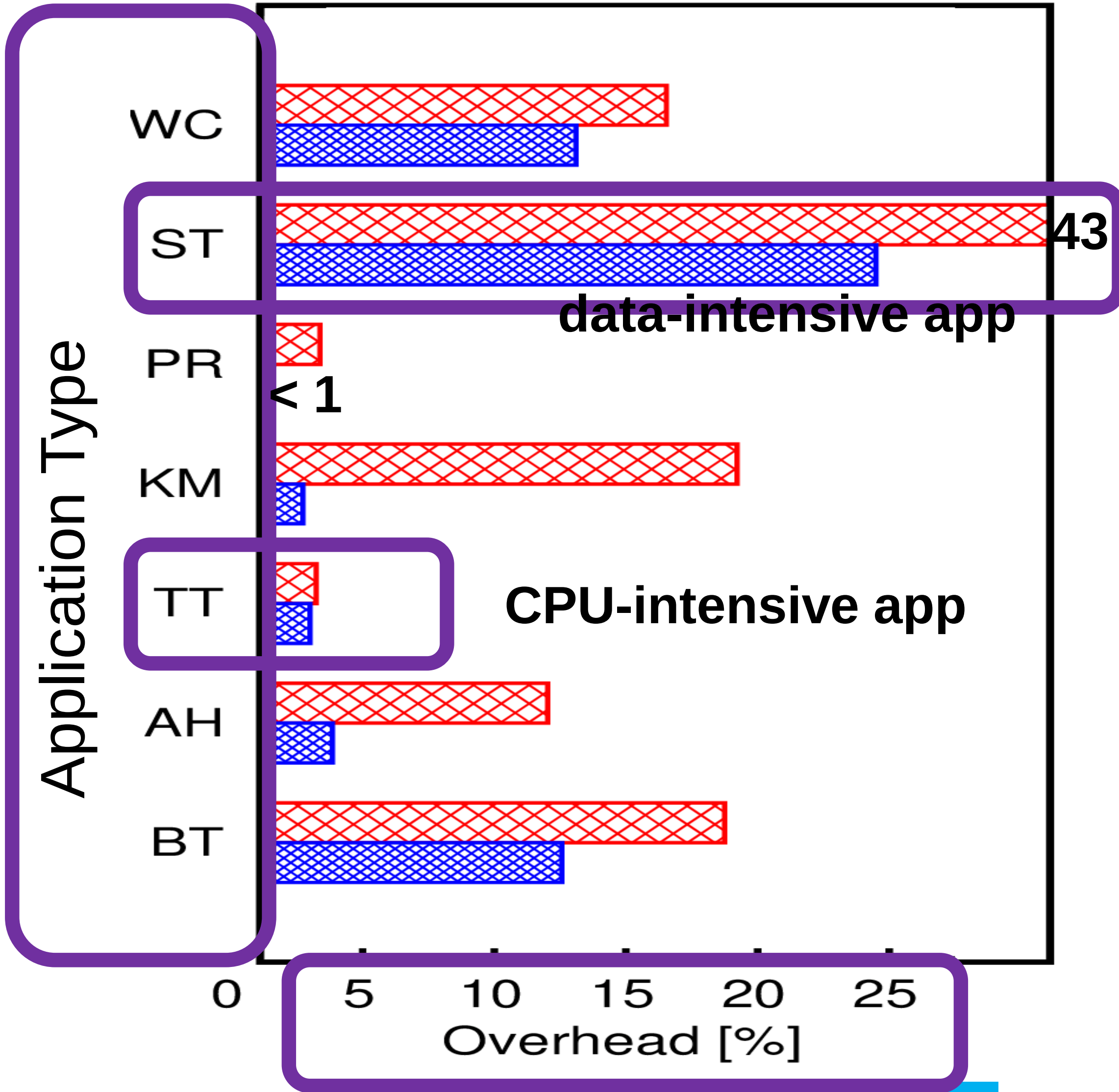
By service – 3 policies:

- Job Slowdown (JS)
- Job Throughput (JT)
- Task Throughput (TT)

© 2017 Alexandru Iosup. All rights reserved.

Performance of dynamic, elastic MapReduce

10 core +10xTR
10 core +10xTC
vs.
20 core nodes (baseline)



TR - good for compute-intensive workloads.

TC - needed for disk-intensive workloads.

Dynamic MapReduce:
< 25% overhead

Fawkes also reduces imbalance

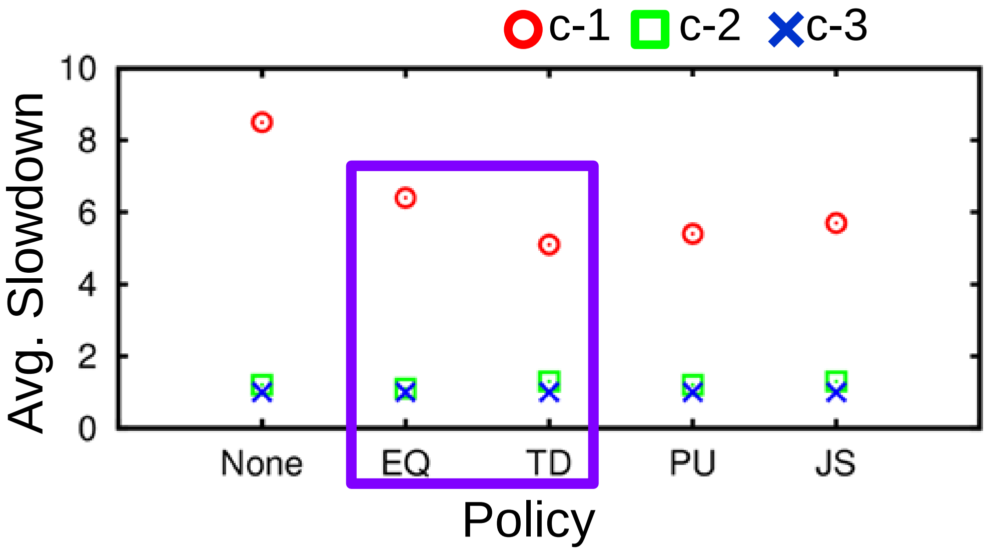
VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TUDelft

Performance of FAWKES

Nodes	45
Frameworks	3
Min. shares	10
Datasets	300 GB
Jobs submitted	900



None – Minimum shares
EQ – EQual shares
TD – Task Demand
PU – Processor Usage
JS – Job Slowdown

C-1: heavy-tailed workload – 1 to 100 GB
C-2/3: short interactive jobs

Up to 20% lower slowdown.
Small impact on the interactive workloads.

© 2017 Alexandru Iosup. All rights reserved.

Massivizing Distributed Systems

MSc topics available

Scheduling
Bags-Of-Tasks
Workflow
Mixed-Workload
Portfolio

Dependability
Failure Analysis*
Space-/Time-Correlation
Availability-On-Demand

New World
Workload Modeling
Interaction Graphs
Business-Critical
Online Gaming

Ecosystem Navigation
Performance Variability
Grid*, Cloud, Big Data
Benchmarking
Longitudinal Studies

Scalability/Elasticity
Delegated Matchmaking*
POGGI*
Area-Of-Simulation
BTWorld*
Auto-Scalers

Socially Aware Techniques
Collaborative Downloads*
Groups in Online Gaming
Toxicity Detection*

Software Artifacts
Graphalytics, etc.

Data Artifacts
A Distributed Systems Memex*

Fundamental Problems
Our Contribution So Far (* Award-winning)

Existing Graph-Processing Systems: *Either* Distributed *or* Heterogeneous

- Distributed CPU-based systems cannot use additional computational power of accelerators

Oracle Labs PGX

APACHE GIRAPH

GraphLab

GraphX

YARN

- GPU-enabled systems are (mostly) single-machine systems, cannot handle large-s

NetSysLab

medusa-gpu
Medusa: Simplified Graph Processing on GPUs

mapgraph Beta
Massively Parallel Graph processing on GPUs

TOTEM

Gunrock
High-performance Graph Primitives on GPU

VertexAPI2

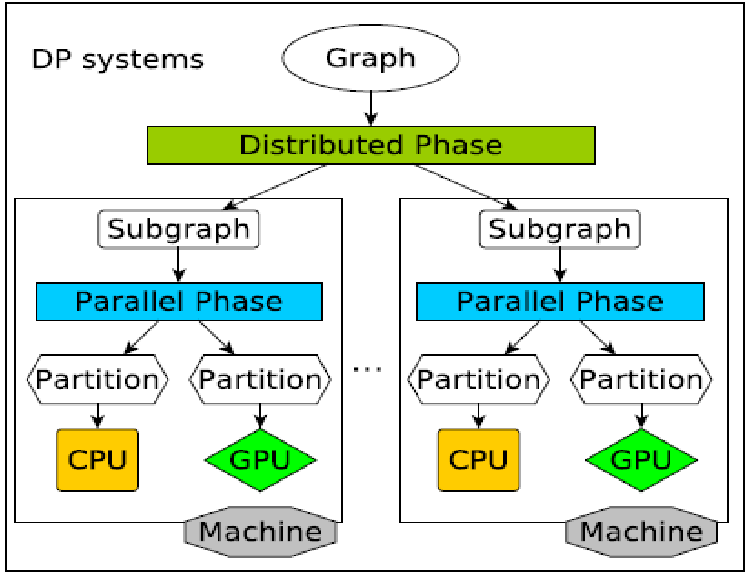
VU VRIJE UNIVERSITEIT AMSTERDAM

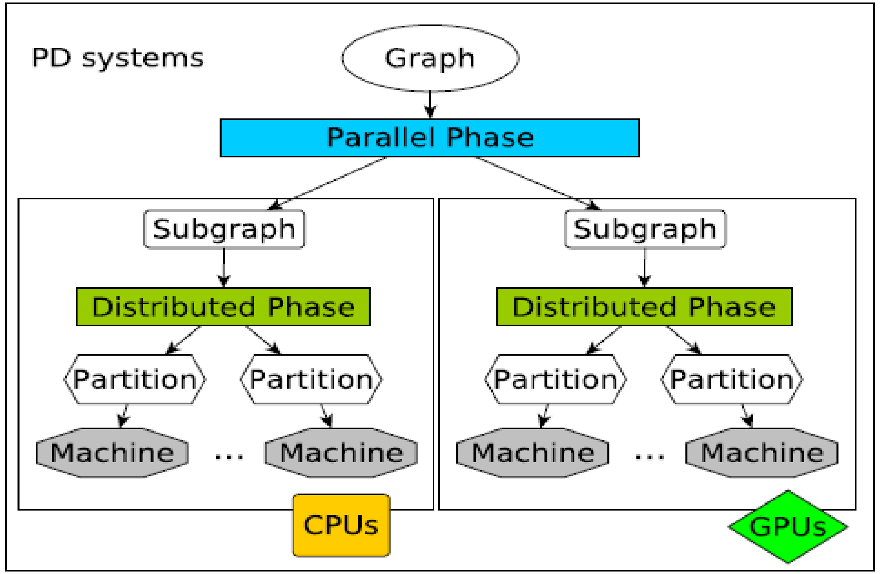
Y. Guo, A. L. Varbanescu, D. Epema, and A. Iosup, "Design and Experimental Evaluation of Distributed Heterogeneous Graph-Processing Systems," *CCGrid, 2016.*

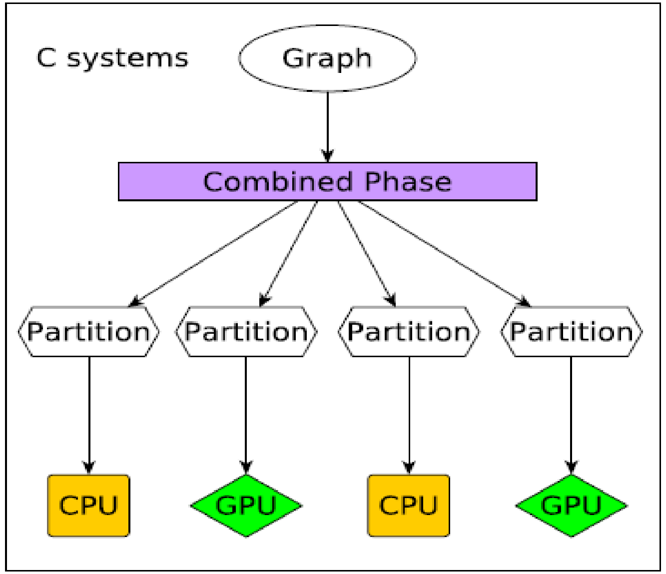
© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Our approach: 3 Families of Distributed *and* Heterogeneous (CPU+GPU) Graph-Processing Systems







Distributed-then-Parallel (DP) Systems

Parallel-then-Distributed (PD) Systems

(Combined Par.-and-Distributed (C) Systems

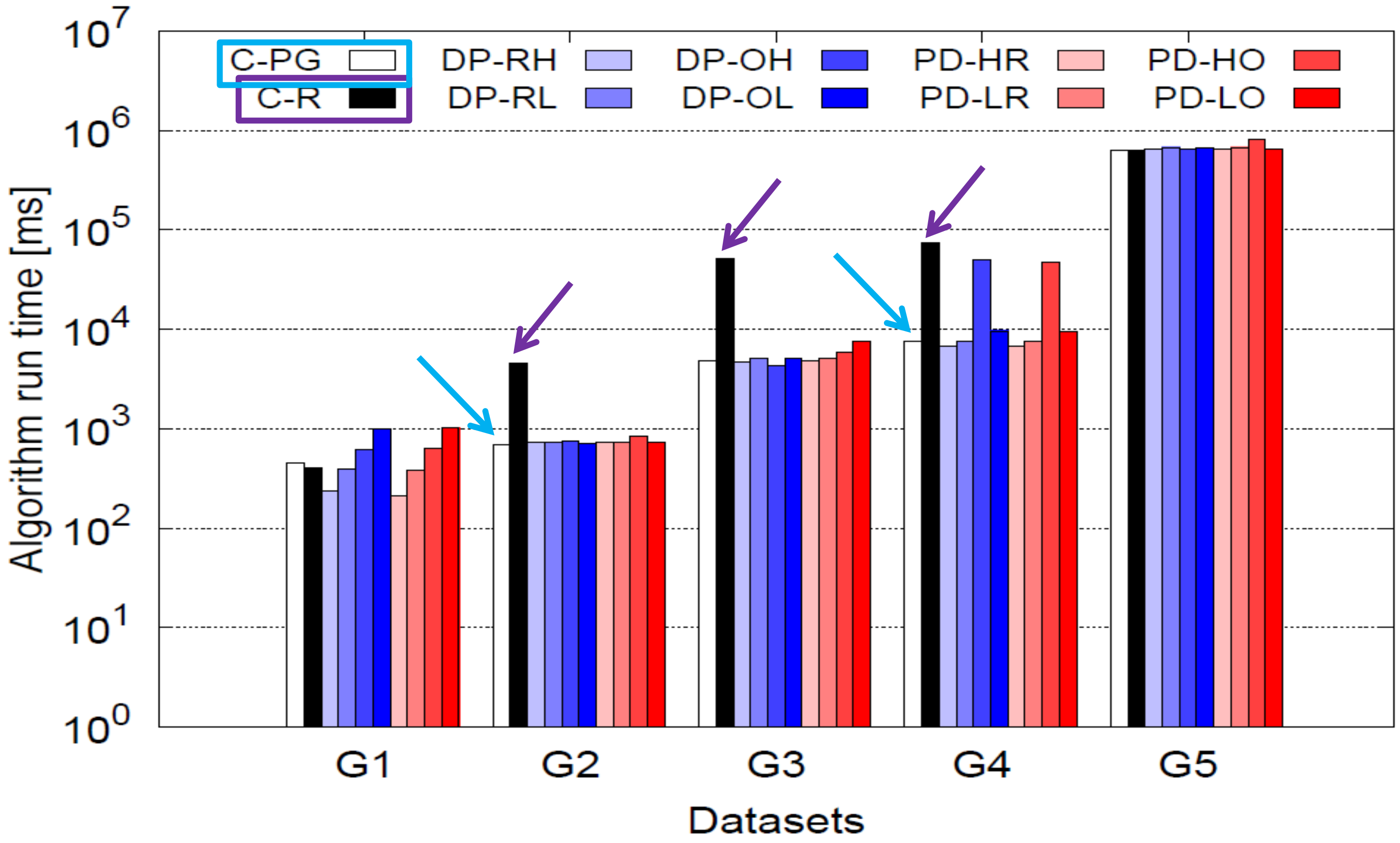
Guo et al., *CCGrid, 2016.*

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

3 Families Explored: 2 Lessons Learned



- PageRank, 4 machines
- Also tried BFS and WCC

1. There is no overall winner, but C-R is in general the worst.
2. Our new PG policy for Combined systems shows good performance.

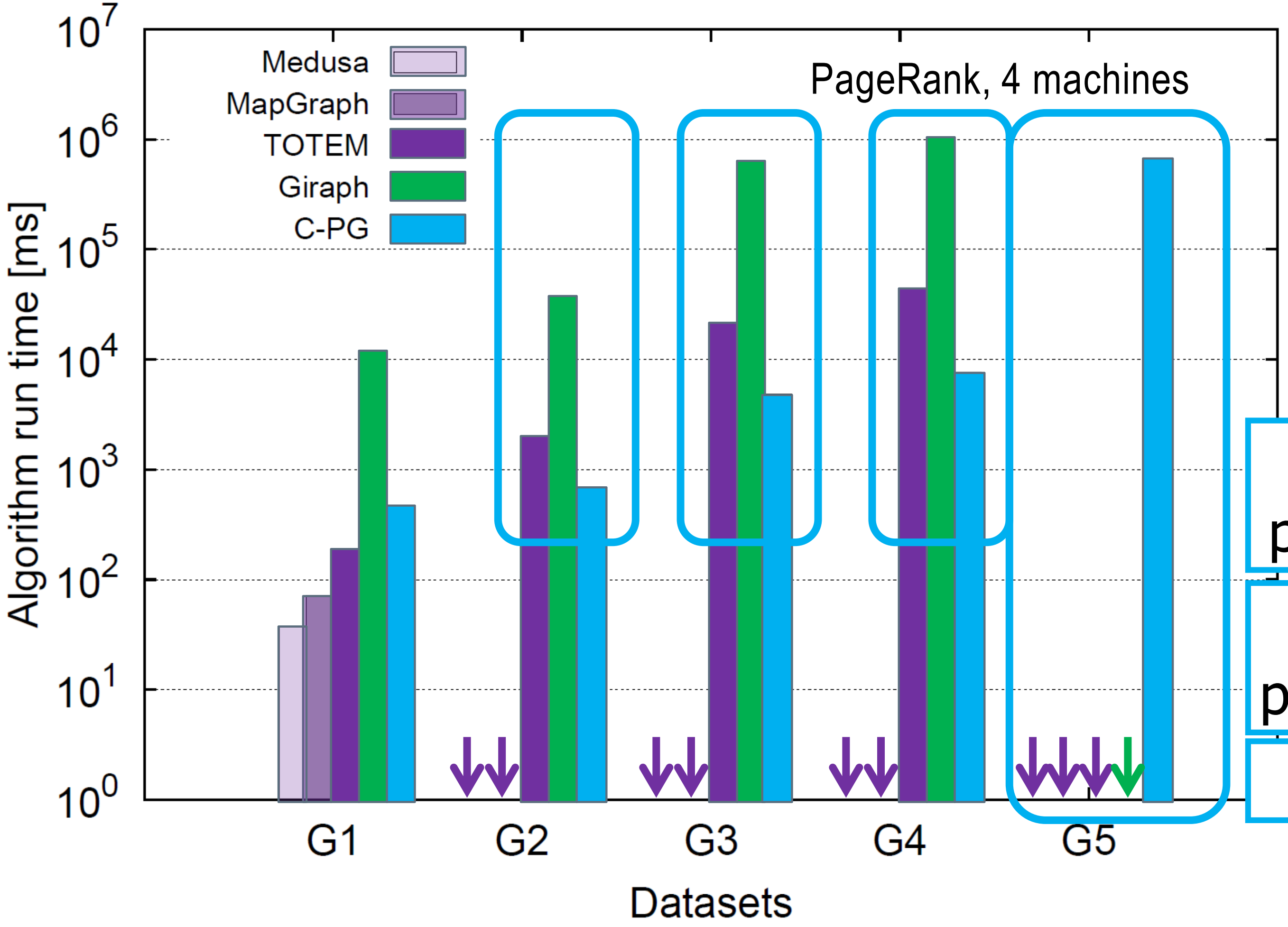
Guo et al., *CCGrid, 2016.*

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Promising Results for Distributed *and* Heterogeneous Graph-Processing Systems



Distributed *and* Heterogeneous

Distributed only

Heterogeneous only

Missing bar = ➔ = system failure

C-PG system has good performance iff. large data

TOTEM has good performance iff. in-memory

C-PG system scales

Guo et al., *CCGrid, 2016.*

UNIVERSITEIT AMSTERDAM

TU Delft

© 2017 Alexandru Iosup. All rights reserved.

Stijn
Heldens

Ana Lucia
Vârbănescu

Alexandru
Iosup

Is there a case for heterogeneous computing in graph processing?

HyGraph

Dynamic Load Balancing for High-Performance Graph Processing on Hybrid CPU-GPU Platforms

Heldens, Varbanescu, Iosup. Dynamic Load Balancing for High-Performance Graph Processing on Hybrid CPU-GPU Platforms. IA3@SC 2016: 62-65

So how about Totem?

- The only heterogeneous graph processing system
  - Single node CPU+multi-GPU
  - Communication optimization
- What's "wrong"/missing ?
  - Static partitioning only
  - BSP model
  - It's not distributed
    - We fixed that, 2014—2015*

*Yong Guo et. al, "Design and Experimental Evaluation of Distributed Heterogeneous Graph-Processing Systems", CCGrid 2015

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Challenges for heterogeneous GP

- Granularity mismatch
  - The CPU requires coarse granularity (i.e., larger jobs),
  - The GPU requires fine granularity (i.e., many tiny jobs).
- Scheduling & load-balancing
  - Jobs need to be assigned to the CPU and/or the GPU.
- CPU-GPU Expensive Communication
  - CPU and GPU need to communicate to synchronize

VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

An alternative: HyGraph*

* S.Heldens et al, “HyGraph: Fast Graph Processing on Hybrid CPU-GPU Platforms by Adaptive Load-Balancing” (SC16 WS)

- Simple vertex-centric API
  - Code is generated for CPU (OpenMP) and GPU (CUDA)
- Data is replicated on all devices
  - Largest graph in our experiments: 0.24GB of memory
- The graph is split into blocks** (groups of vertices)
  - CPU: one block per thread
  - GPU: one block per SM

** Similar to shards in G-shards in CuSha and matrix rows GraphMat

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

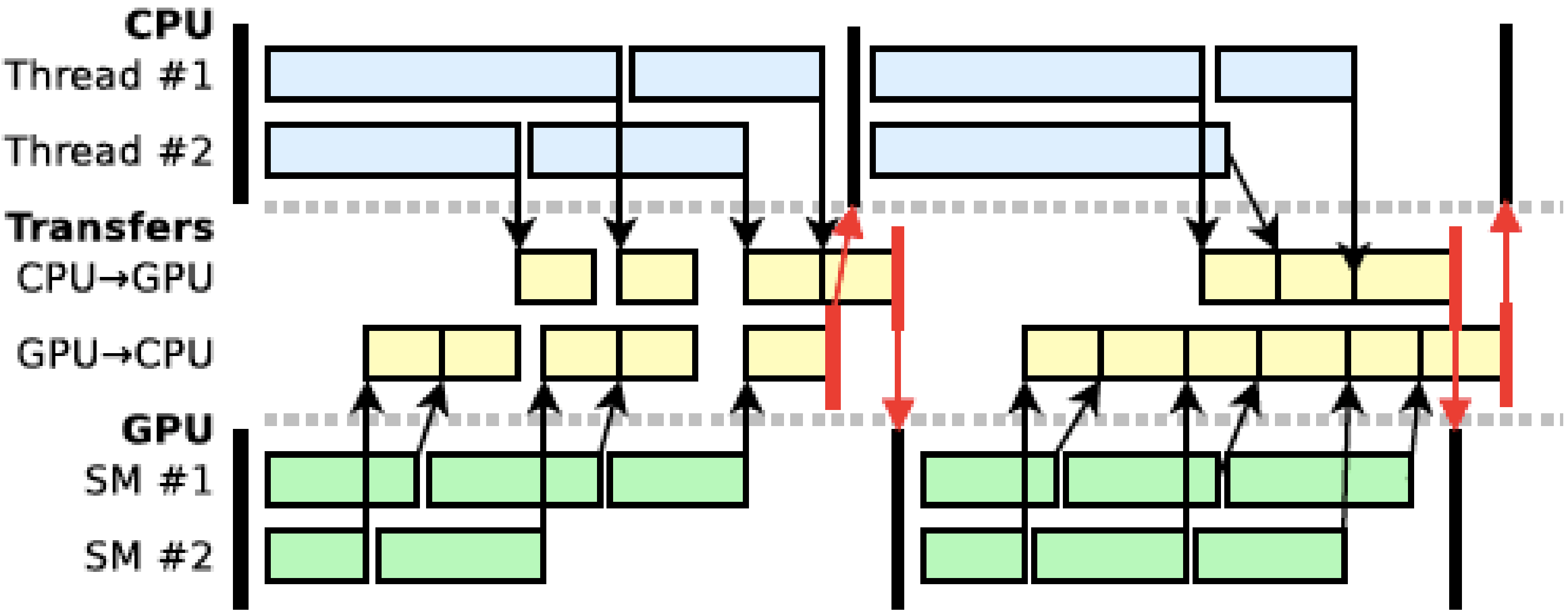
HyGraph key points

- Pre-processing
  - Reorganizes the graph in a block-based structure
- Granularity
  - Different block sizes for CPU and GPU
- Scheduling
  - Cooperation between CPU and GPU only at block-level
- Communication-computation overlap
  - As soon as a block is finished, results are sent
    - We use CUDA streams and multi-job kernels

© 2017 Alexandru Iosup. All rights reserved.

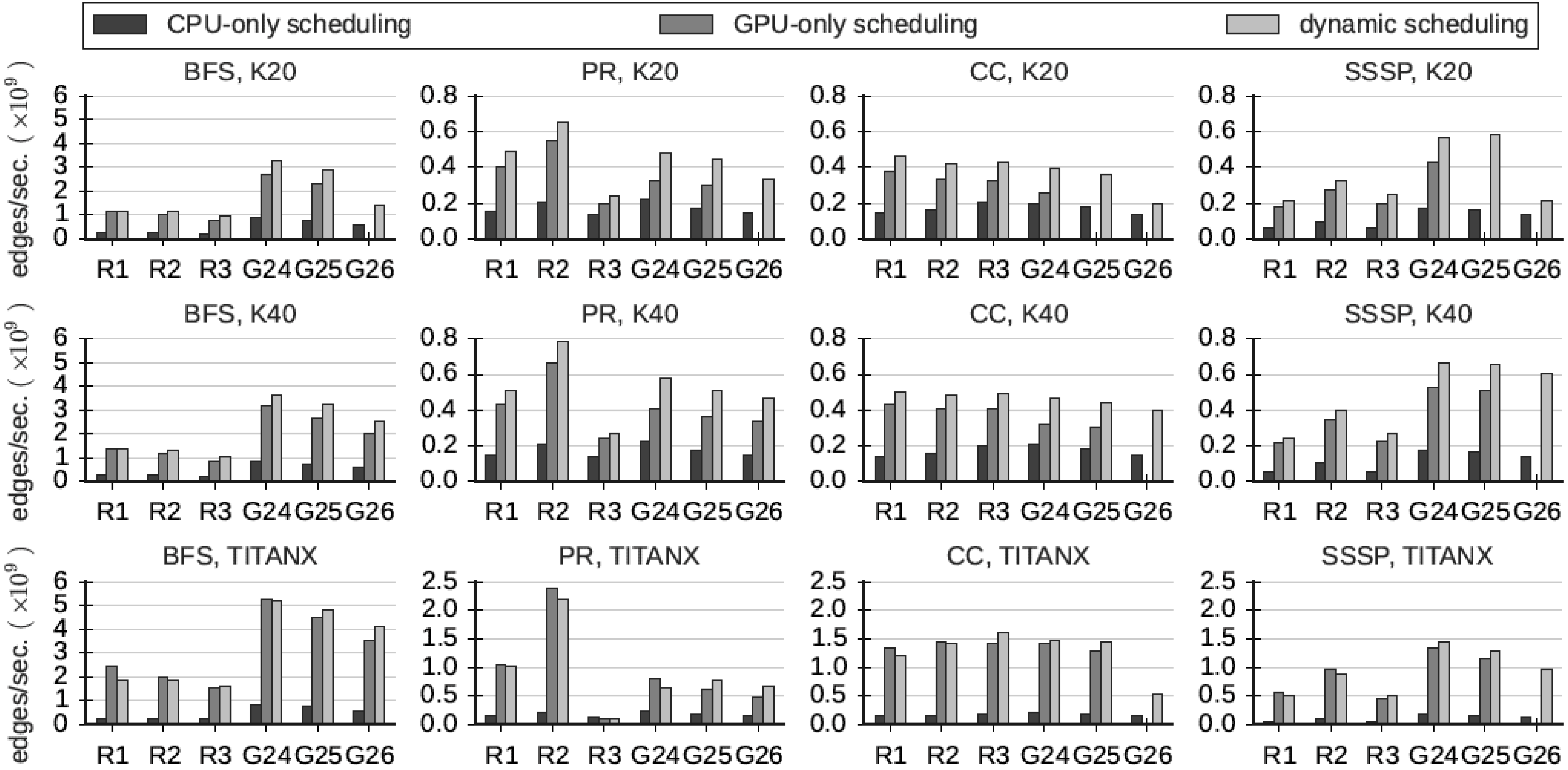
HyGraph CPU+GPU processing

- Jobs dispatched on CPU and GPU



© 2017 Alexandru Iosup. All rights reserved.

HyGraph results: performance

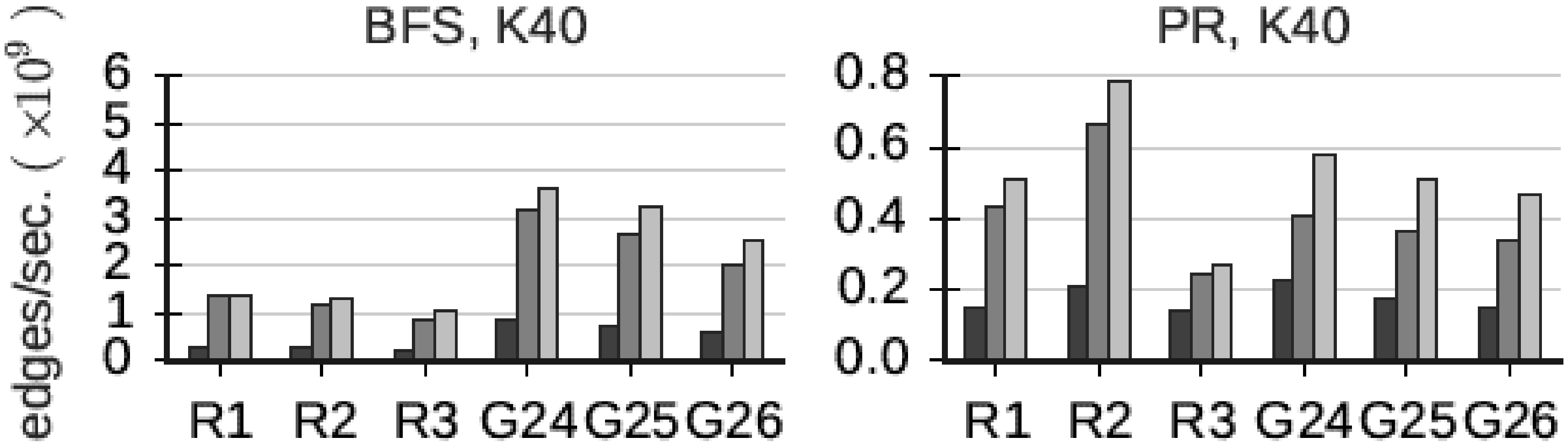


UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TUDelft

HyGraph results: performance



The GPU outperforms the CPU.
The hybrid performance improvement is between 3% and 37.3%

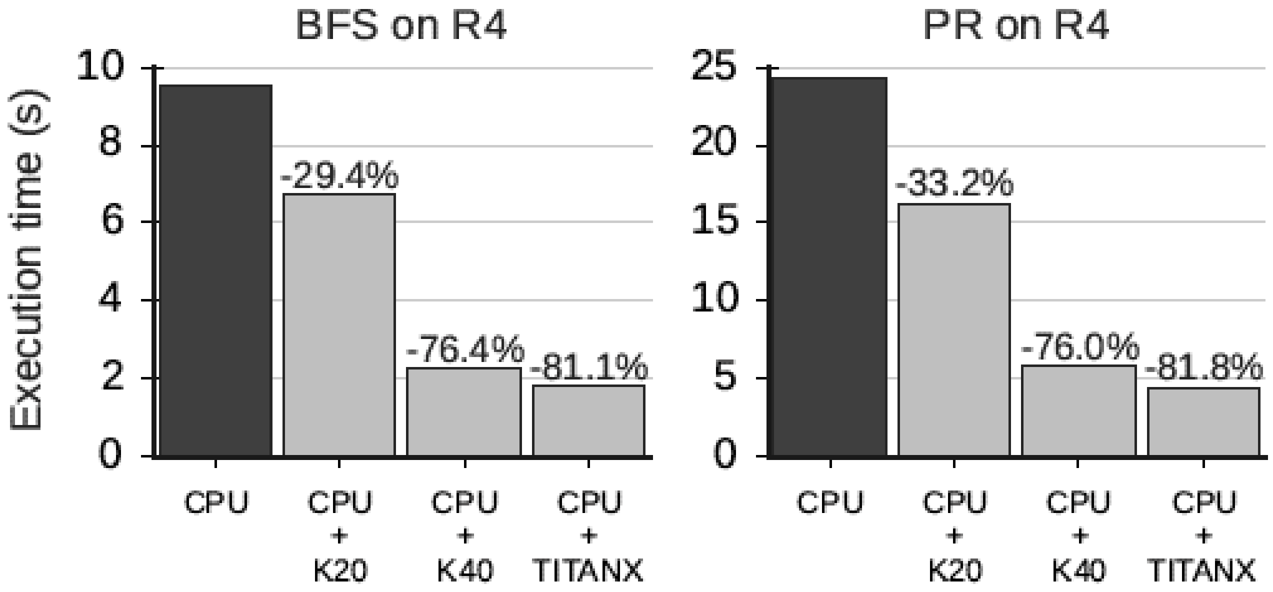
Dynamic scheduling adds little overhead, and outperforms static partitioning.

VU

TUDelft

HyGraph results: size

- 1.8B edges graph
  - K20 : 32.7% , K40 : 79%, TITANX : 84.3%2



VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Lessons learned

- Hybrid graph processing possible
  - HyGraph provides this “for free”
  - Reasonable impact in performance (5-37%)
  - Significant impact as “extra-buffer” for GPU memory
- Performance gain and simplicity of design due to GPU improvements
- Graph ordering and block-size tuning are essential for performance
- Static partitioning is too general to fit iterative graph processing

VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Sietse
Au

Alexandru
Uțã

Alexey
Ilyushkin

Alexandru
Iosup

Is there a case for elastic computing in graph processing?

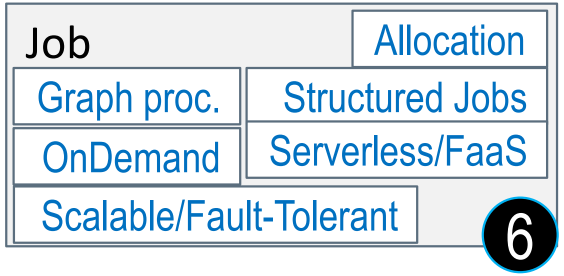
JoyGraph

An Elastic, Distributed, Easily Programmable System for Graph Processing

(Jun 2017)

(unpublished, so please do not record or share)

6. Workload/Job Orchestration and Scheduling



- On-Demand
  - Availability-on-Demand
- Scalable and Fault-tolerant
  - Area of Simulation
- Support for workflows and other structured jobs
- Serverless/FaaS execution

© 2017 Alexandru Iosup. All rights reserved.

Siqi
Shen

Alexandru
Iosup

Dick
Epema

Availability-on-Demand

Easy to specify, auto-tuning availability mechanism for datacenters

Shen, Iosup, Israel, Cirne, Raz, Epema. An Availability-on-Demand Mechanism for Datacenters. CCGRID 2015: 495-504

Massivizing Distributed Systems

Scheduling
Bags-Of-Tasks
Workflow
Mixed-Workload
Portfolio

Dependability
Failure Analysis*
Space-/Time-Correlation
Availability-On-Demand

With Technion, Google

New World
Workload Modeling
Interaction Graphs
Business-Critical
Online Gaming

Ecosystem Navigation
Performance Variability
Grid*, Cloud, Big Data
Benchmarking
Longitudinal Studies

Scalability/Elasticity
Delegated Matchmaking*
POGGI*
Area-Of-Simulation
BTWorld*
Auto-Scalers

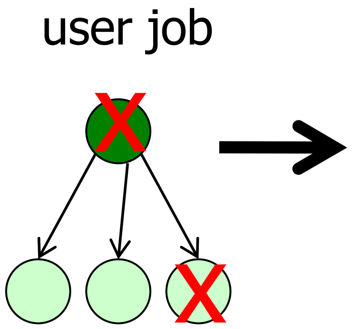
Socially Aware Techniques
Collaborative Downloads*
Groups in Online Gaming
Toxicity Detection*

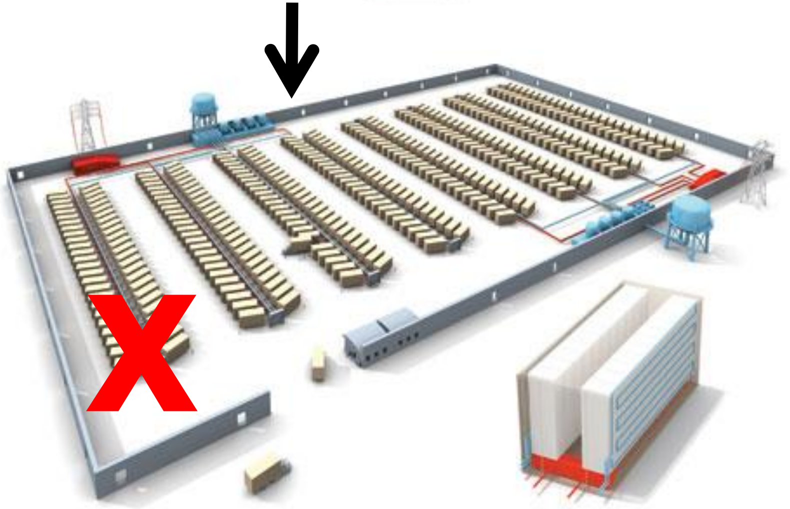
Software Artifacts
Graphalytics, etc.

Data Artifacts
A Distributed Systems Memex*

Fundamental Problems
My Contribution So Far (* Award-winning)

Addressing Failures in Datacenters of IaaS Clouds





Main idea: Create task replicas during periods of high required availability

Siqi Shen, Alexandru Iosup, Assaf Israel, Walfredo Cirne, Danny Raz, Dick H. J. Epema: An Availability-on-Demand Mechanism for Datacenters. CCGRID 2015: 495-504.

Google

Research question

How and when to use High Availability (HA) techniques effectively in datacenters, to counter resource failures?

More precisely, considering the time and space dimension of jobs consisting of multiple tasks,

RQ1: when, and for which tasks, to require HA?

RQ2: how to implement HA?

(RQ3: how can users with relatively low technical background specify HA requirements?)

System/job model/failure model

- Infrastructure as a Service, only CPU as a resource

- A job can consist of multiple tasks
  - master-slave (MS): slaves dependent on master
  - bag-of-task (BoT): no dependencies

- Fail-stop + recovery after a while
- Failing tasks are resubmitted to the system-level queue and are restarted from scratch

© 2017 Alexandru Iosup. All rights reserved.

Availability on Demand (AoD)

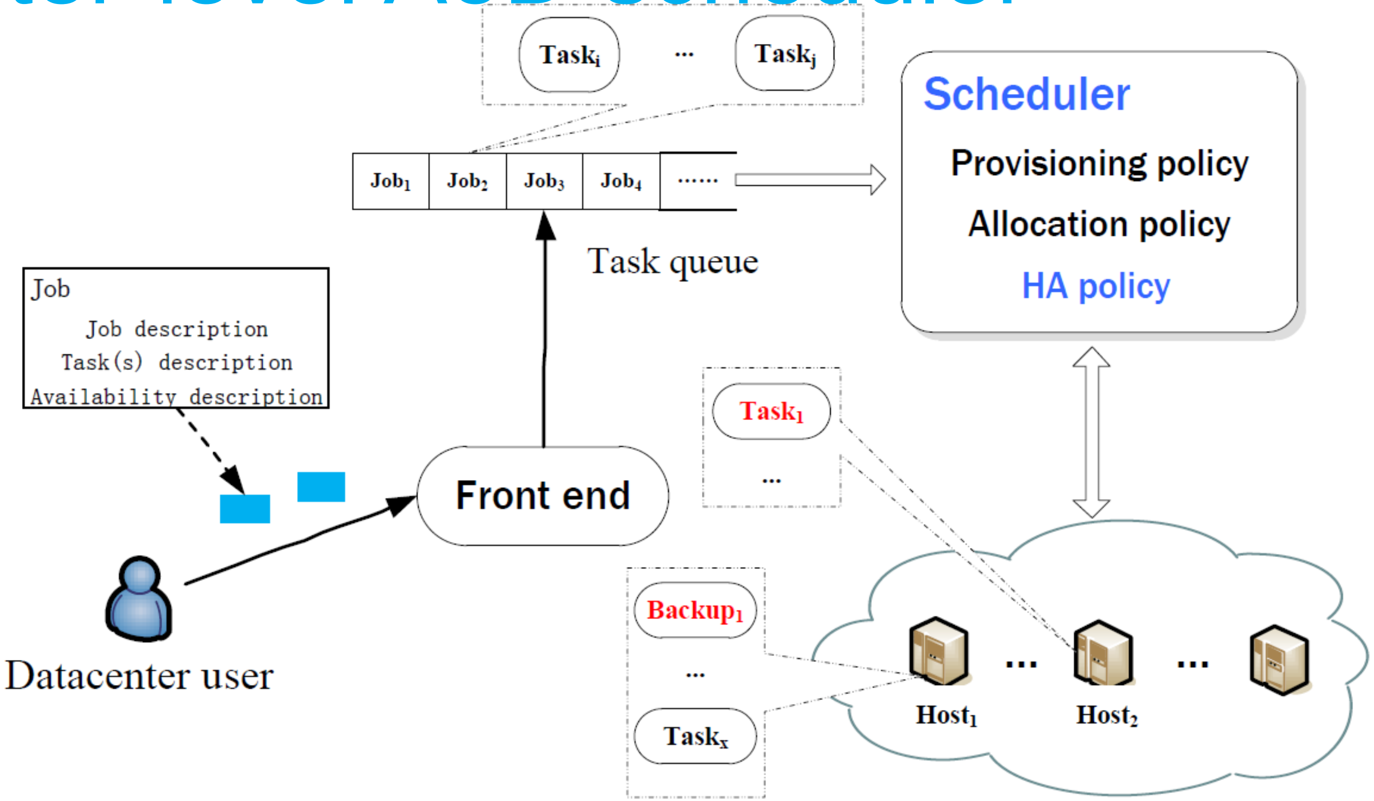
- API
  - **single call**, easy-to-use
  - specifies the dynamic requirements per service component
- **`SetAvailability`**`(id, availability, time period)`
  - "id" of the job or task
  - "availability" level: normal (NA) or high (HA))
  - "time period" is required availability duration
- For instance, for an MS application:
  - `SetAvailability(MasterId, HA, all)`
  - `SetAvailability(WorkerId, NA, all)`
- For an online game:
  - `SetAvailability(gamingAppId, HA, 9PM->1AM)`

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Datacenter-level AoD scheduler



- AoD+R HA policy: Create a **backup task** for every task that requires HA during the time it requires HA + policy to allocate backup tasks

© 2017 Alexandru Iosup. All rights reserved.

Policies used for comparison with AoD+R

- **None**
  - simply restart a task if it fails
- **Rnd**
  - with a fixed probability, add to each task an AA backup task that runs for the entire duration of the task
- **AoD-I**
  - variation of AoD+R which does not distinguish between master tasks and slave tasks (AoD-R: master always HA, AoD-I: master also NA periods)
- **Pred**
  - ideal policy which uses perfect prediction of failures (cannot work in practice, but gives an idea of optimum)

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Experimental Setup: Simulator and traces

- **Simulator**
    - event-based simulation
    - based on our own DGSim and Cloudsim
    - simulated system: 1,000 x 16-core machines
- **Input**
    - real-world workload traces
    - realistic failure generation (based on our previous work)

Trace Type	Trace name	#jobs	Avg. runtime [s]	Avg. CPU	Trace source
Sci.comp.	KTH-SP2	28,489	8876	7.7	PWA [30]
Sci.comp.	DAS2	219,618	530	10.3	GWA [31]
Onl.Gam.	DLI	109,250	2232	1	GTA [32]

VU

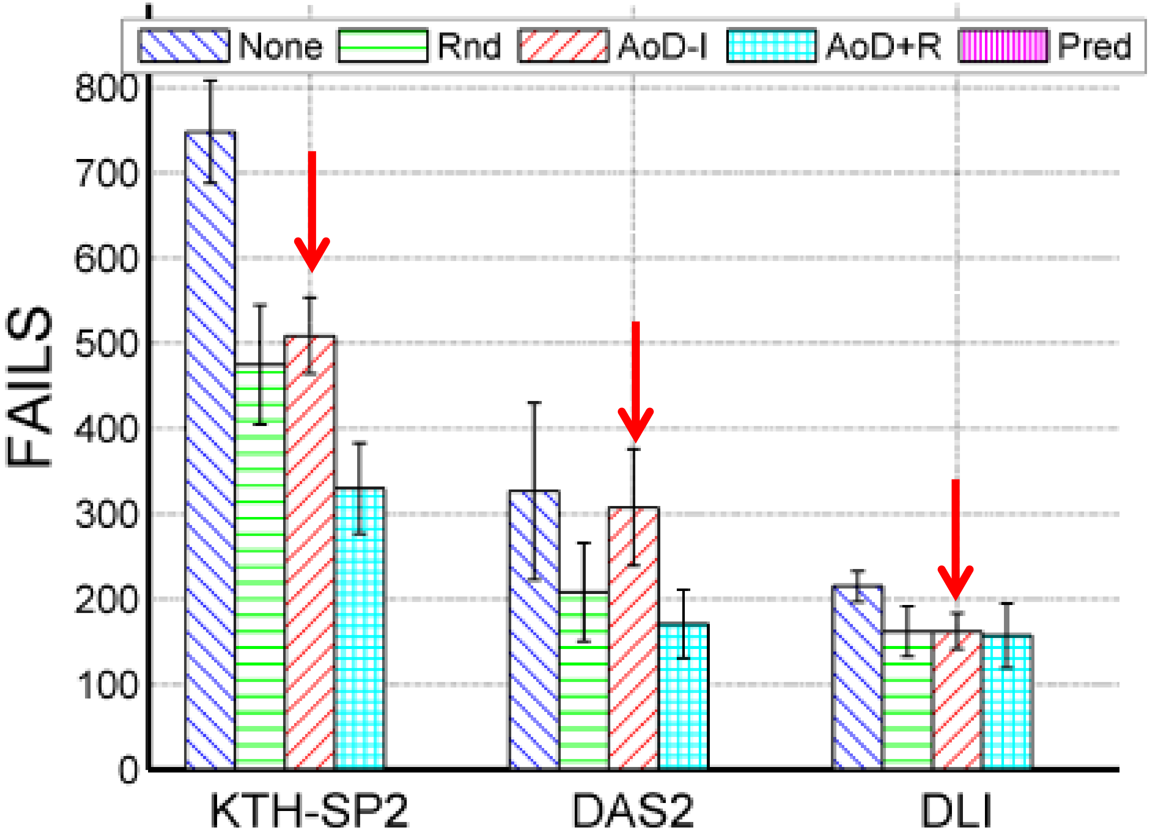
TUDelft

Experimental setup: Metrics

- **FAILS**:
  - total number of **failure events**
- **CRITS**:
  - number of **critical failure events** (i.e., during HA periods)
- **CPU hours**
  - measures the cost efficiency

© 2017 Alexandru Iosup. All rights reserved.

Experimental Results (1/3): Number of failure events (FAILS)



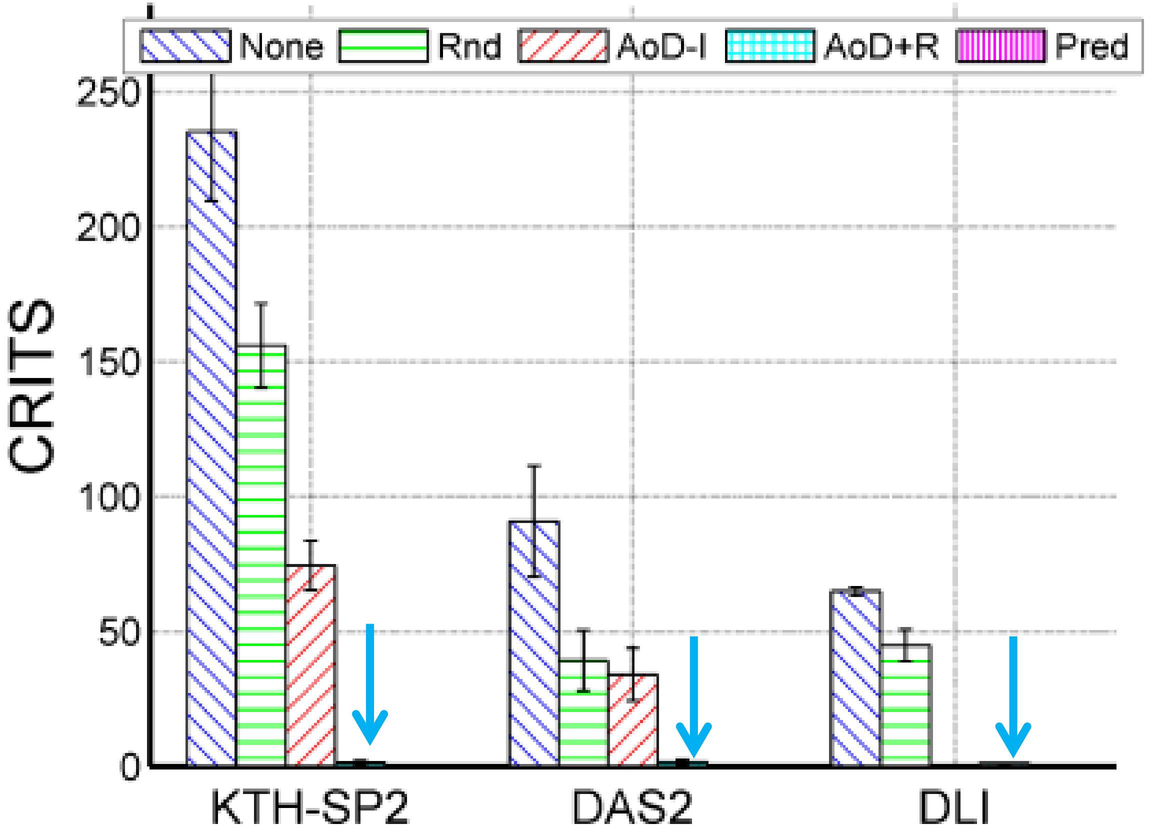
AoD-I: high FAILS, because the master can fail, which makes all other tasks to fail too

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Experimental Results (2/3): Number of critical failure events (CRITS)



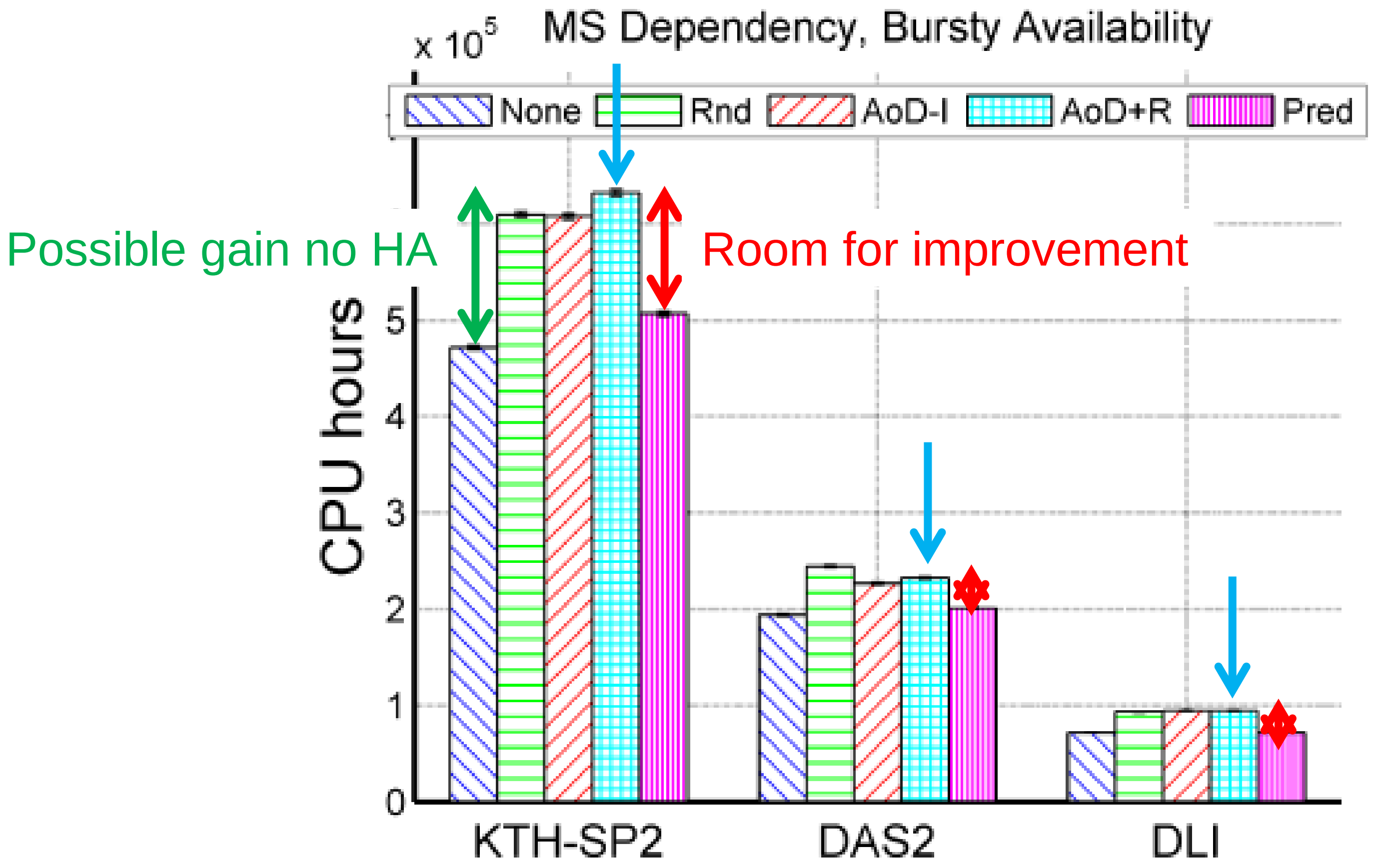
AoD+R: excellent CRITS performance

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Experimental Results (3/3): Used CPU hours



AoD+R policy consumes a reasonable amount of CPU hours, similar to other policies that use AA techniques

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Siqi
Shen

Alexandru
Iosup

Dick
Epema

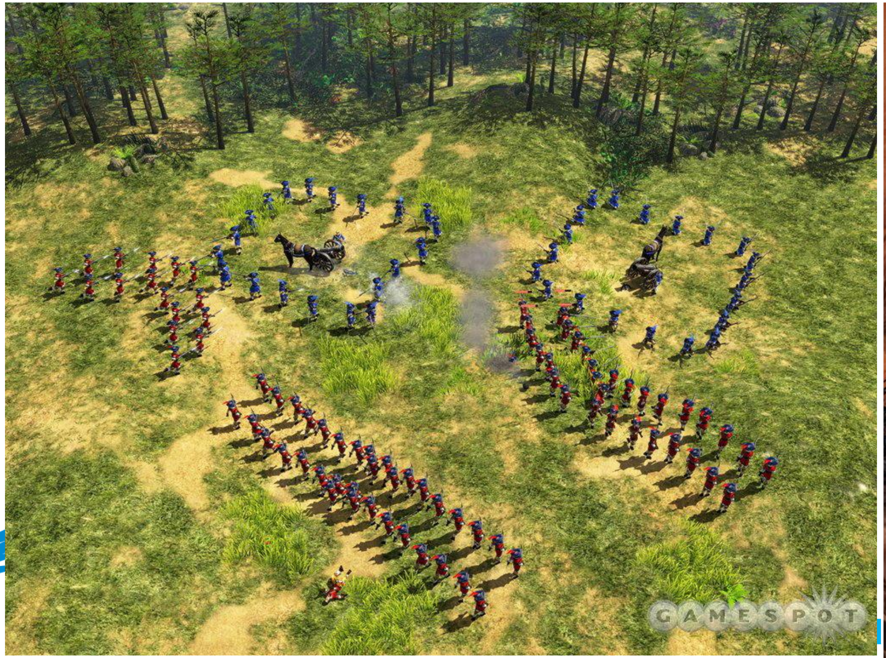
Area of Simulation

Mechanism and Architecture for Scalable Consistency Management in Multi-Avatar Virtual Environments

Shen, Hu, Iosup, Epema. Area of Simulation: Mechanism and Architecture for Multi-Avatar Virtual Environments. TOMCCAP 12(1): 8:1-8:24 (2015)

RTS Games

- Players control tens up to hundreds of units.
- Players need to take decisions in real-time.

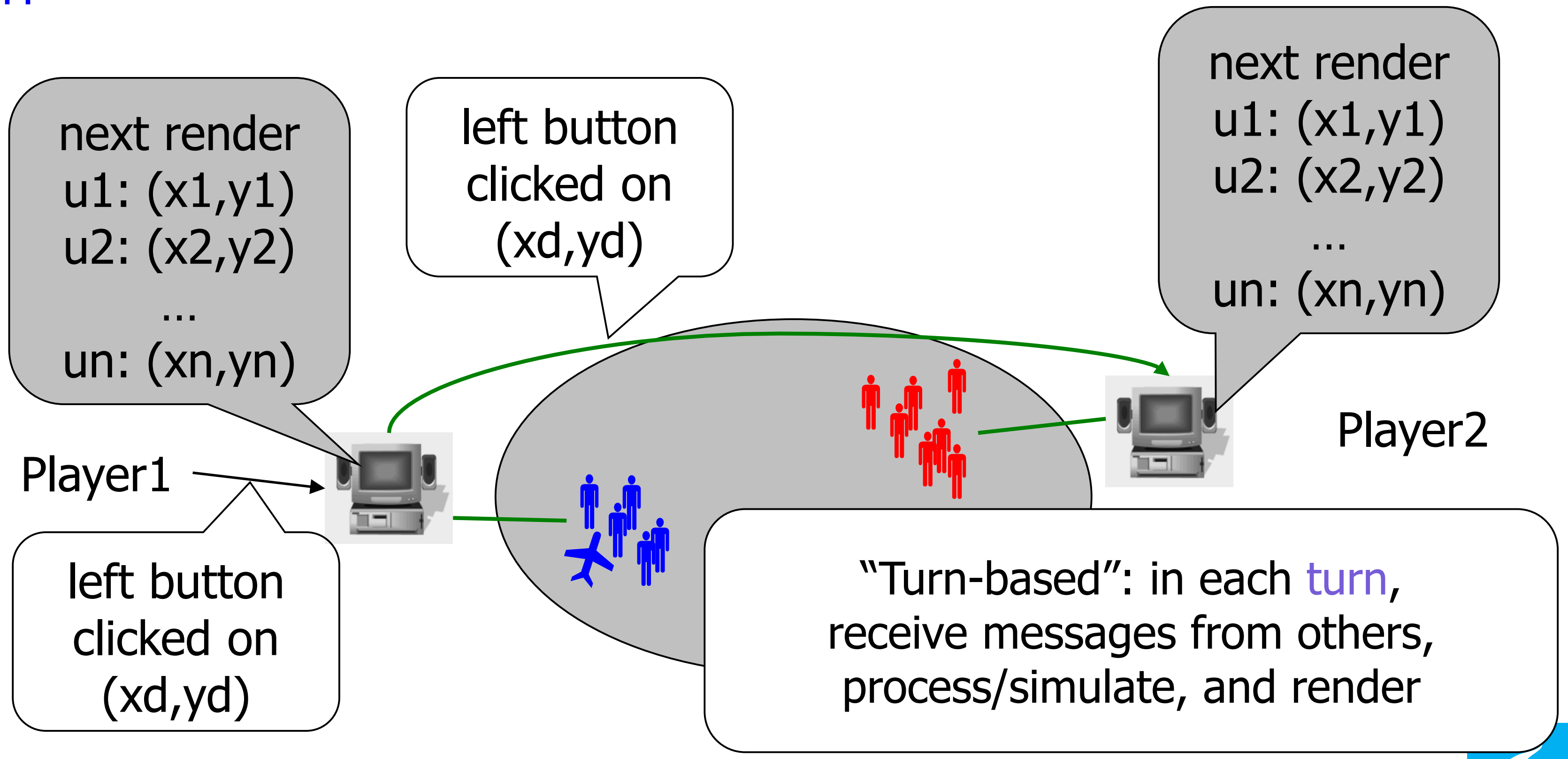


Consistency: 1.5k Archers on 28.8k Line [1/3]

Age of Empires [Bettner & Terrano GDC01]

Problem: Too many players/units to update at each click

➔ New Approach: Simultaneous simulations

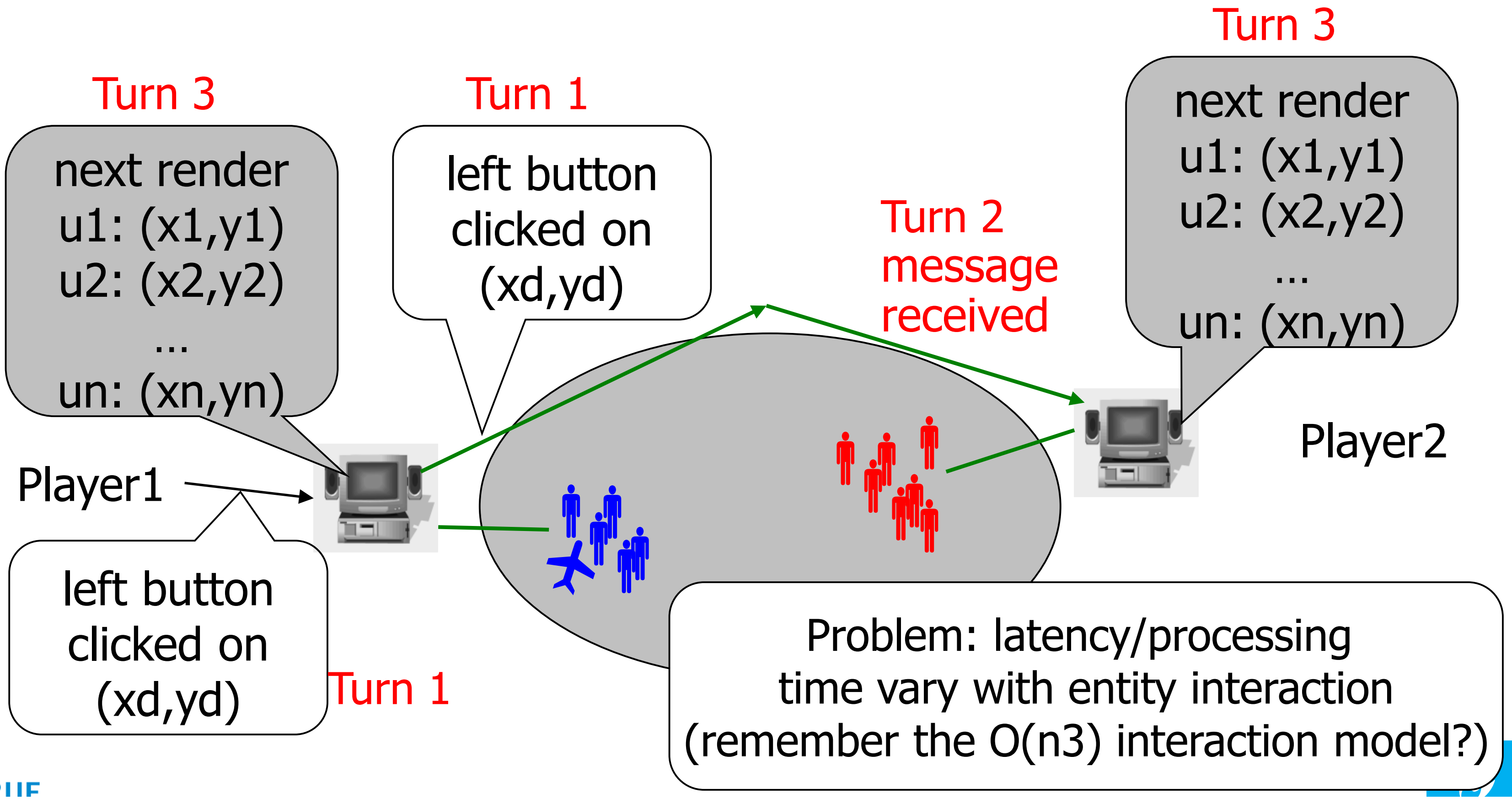


Source: http://www.gamasutra.com/view/feature/131503/1500_archers_on_a_288_network_.php?print=1.
Slides source: http://www.cs.duke.edu/courses/spring08/cps214/lectures/lecture18.ppt

Consistency: 1.5k Archers on 28.8k Line [2/3]

Problem: need very long turn to finish everything!

➔ **Approach: Pipelining of operations, have multi-turn tick**



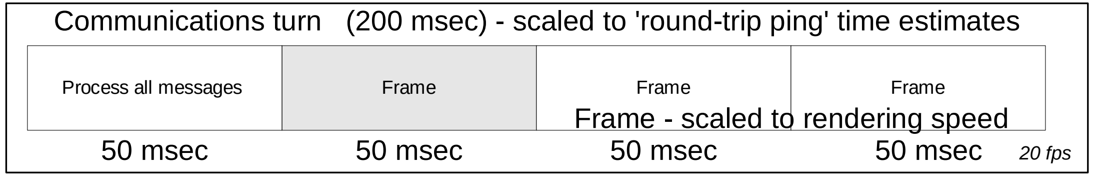
Source: http://www.gamasutra.com/view/feature/131503/1500_archers_on_a_288_network_.php?print=1.
Slides source: http://www.cs.duke.edu/courses/spring08/cps214/lectures/lecture18.ppt

Consistency: 1.5k Archers on 28.8k Line [3/3]

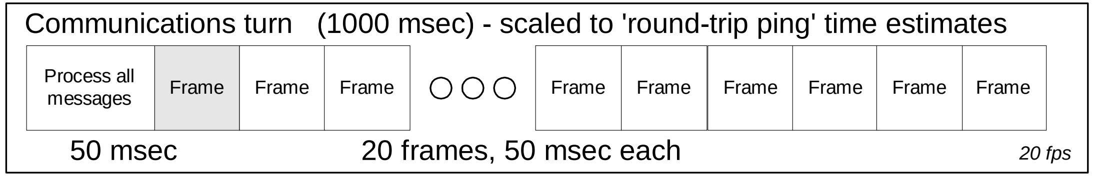
Approach: dynamic turn length

- Adjusts to real delays experienced by real players

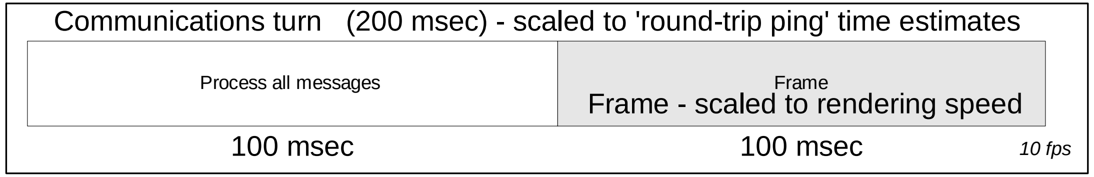
Regular Net/CPU
200 ms latency
50 ms proc/render



Slow Net/Reg. CPU
1000 ms latency
50 ms proc/render



Reg. Net/Slow CPU
200 ms latency
100 ms proc/render

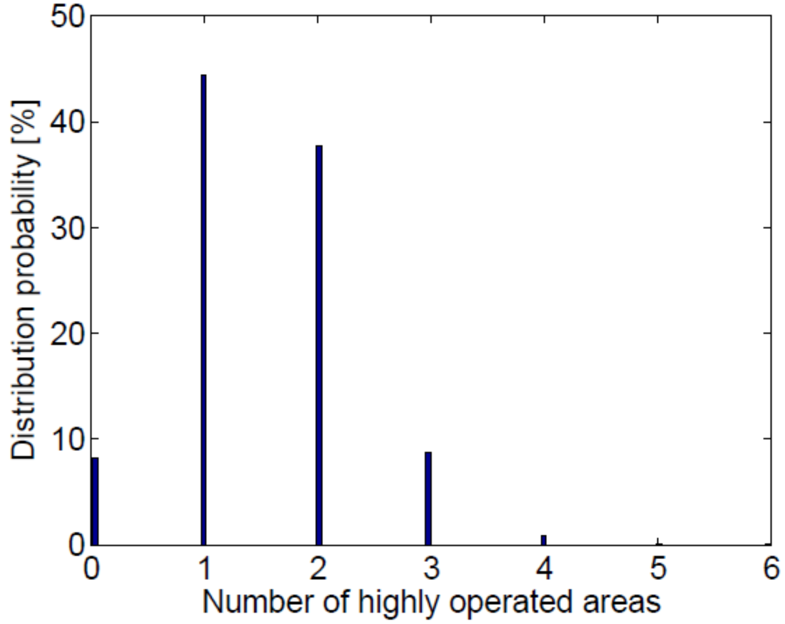


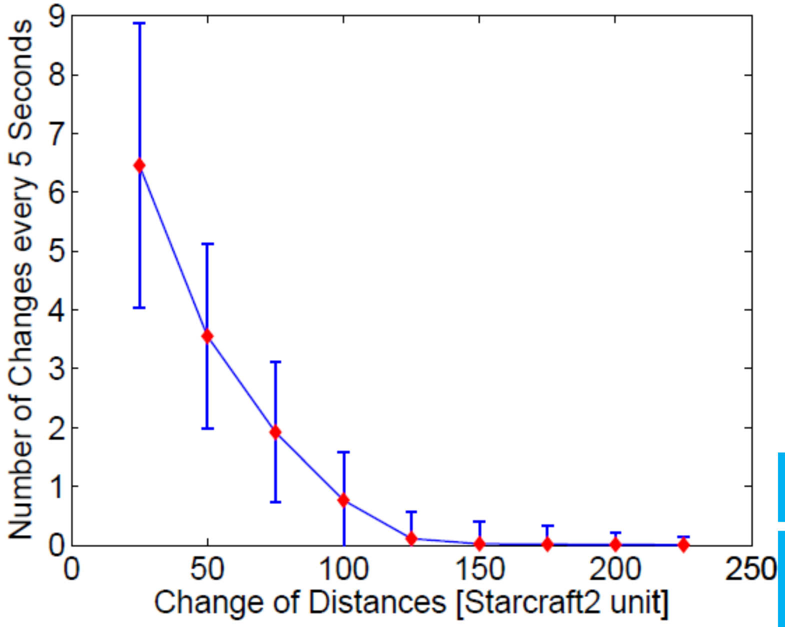
- Problem: slow turns. Could we use only Area of Interest?

Source: http://www.gamasutra.com/view/feature/131503/1500_archers_on_a_288_network_.php?print=1.
Slides source: http://www.cs.duke.edu/courses/spring08/cps214/lectures/lecture18.ppt

Traditional Area-of-Interest does not work

- Area of Interest (AoI) = traditional mechanism for RPG:
  only receive information around avatar, but…
- …In RTS, each player has tens of units under control, so too much data to be transferred
- … In RTS, we were the first to show that players change interest more often than in RPG and FPS games, so too high management overhead





Area of Simulation: Core Idea

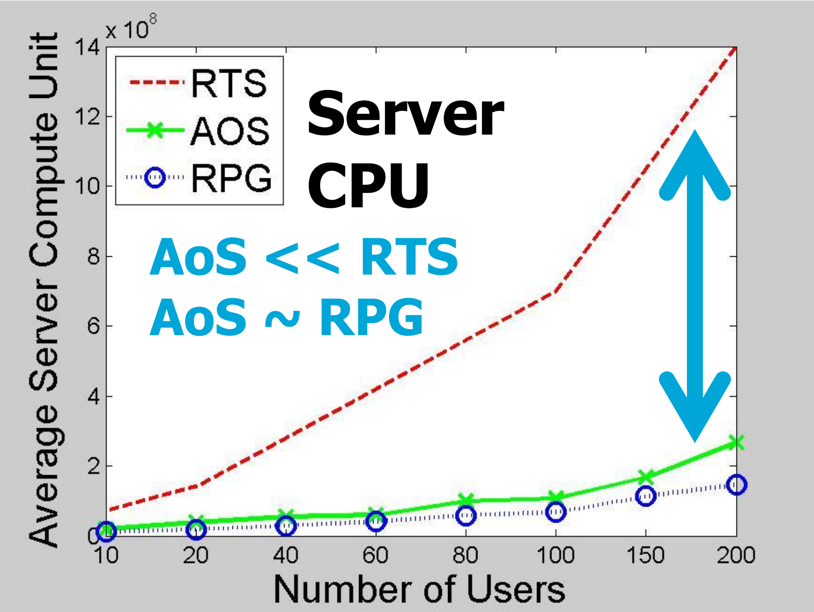
- Partition the game into multiple areas (rectangular)
- Each player pays attention to different areas + attention level
- Depending on attention level and machine performance, the player will receive different types of information (**commands** or **state**) about the game world
  - AoS: Area of Simulation = high-attention area, full simulation based on input commands (CPU-intensive)
  - AoU: Area of Update = low-attention area, receives state (Net)
  - NUA: No update area
- Each player can have multiple AoS and AoU

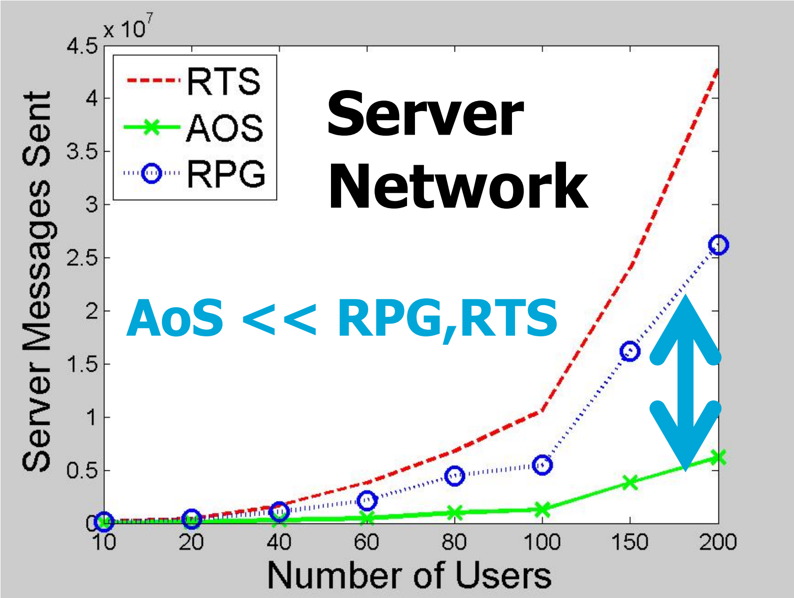
S. Shen, A. Iosup, D. H. J. Epema, and S.-Y. Hu. Area of Simulation: Mechanism and Architecture for Multi-Avatar Virtual Environments, ACM Trans. Multimedia Comput. Comm. App. 2015.

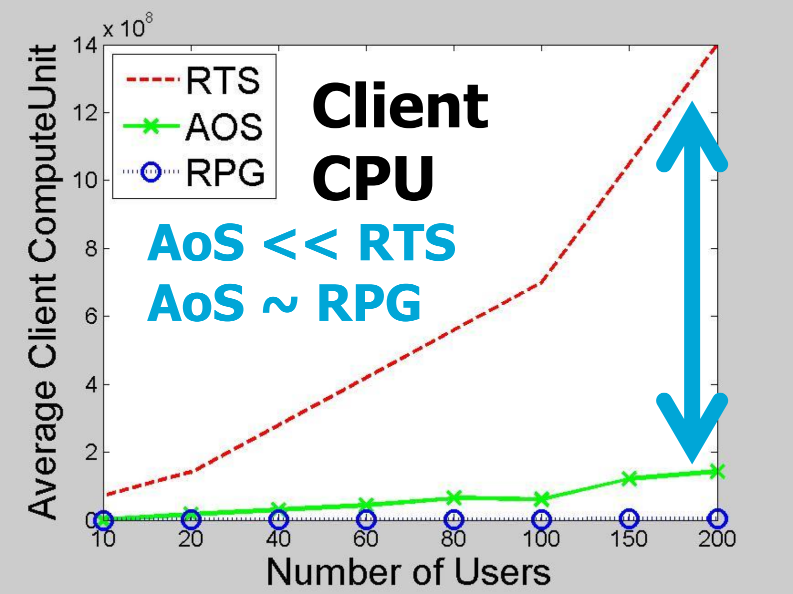
Experimental results

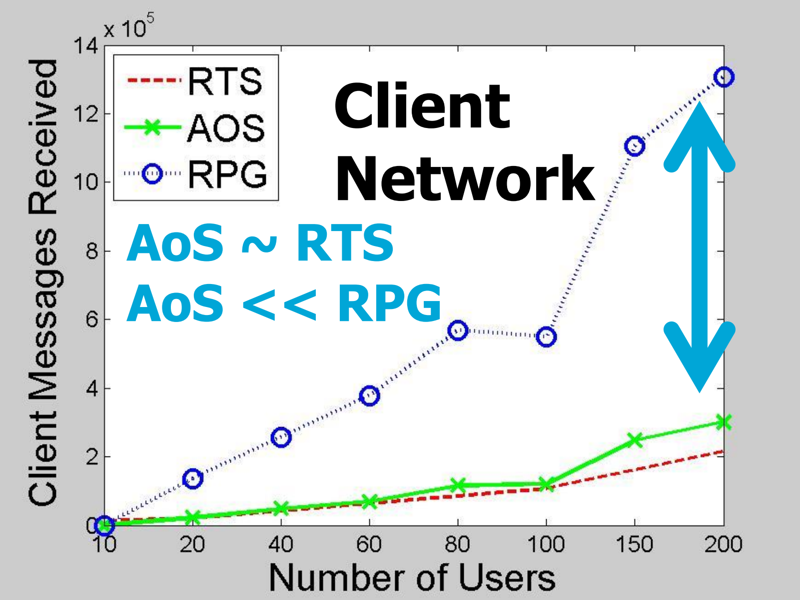
- Simulator and prototype RTS game
- Evaluate in two Cloud platforms: EC2 and Azure
- Prototype about 20k lines of C++ code
  - Based on an open source game (~6k lines)
- Up to 200 players and **10,000 battle units**
  - **State-of-the-art unplayable at 1-2,000**
  - **Crashes not uncommon due to CPU and Network bottlenecks**
- ➔ Using our AoS-based method can lead to **lower CPU** consumption than pure event-based method (RTS) and **lower network consumption** than pure update-based method (RPG)

© 2017 Alexandru Iosup. All rights reserved.



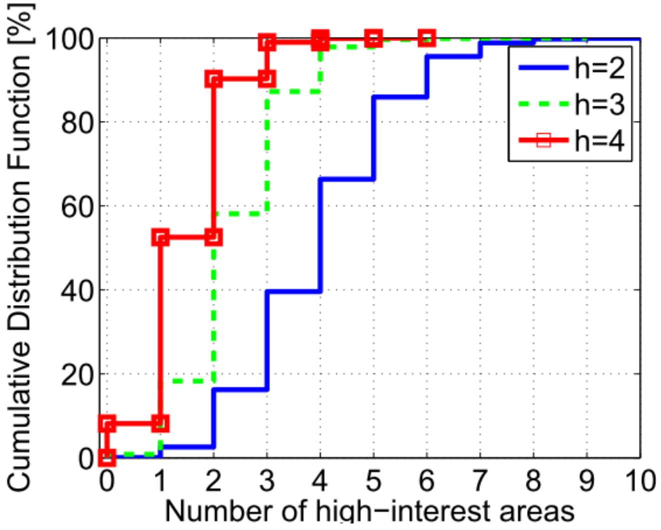






VU AMSTERDAM
TU Delft
© 2017 Alexandru Iosup. All rights reserved.

Area of Simulation Take-Home Message



h=high:normal interest

- **Area of Simulation is needed**
  - N (practice) vs. 1 (assumed) Areas of Interest
- Simulator and real-world prototype RTS game
  - Prototype about 20k lines of C++ code
  - Evaluated in two cloud platforms, Amazon EC2 and Microsoft Azure
- Our AoS-based method leads under most circumstances to
  - Higher scalability Up to 200 players and **10,000 battle units**

    vs. state-of-the-art: unplayable at 1,000-2,000 battle units + crashes above 5,000+
  - **Lower CPU** consumption than pure event-based method (RTS) and **lower network** consumption than pure update-based method (RPG)

VU VRIJE UNIVERSITEIT AMSTERDAM

S. Shen, A. Iosup, D. H. J. Epema, and S.-Y. Hu. Area of Simulation: Mechanism and Architecture for Multi-Avatar Virtual Environments, ACM Trans. Multimedia Comput. Comm. App. 2015.

Delft

Erwin
van Eyk

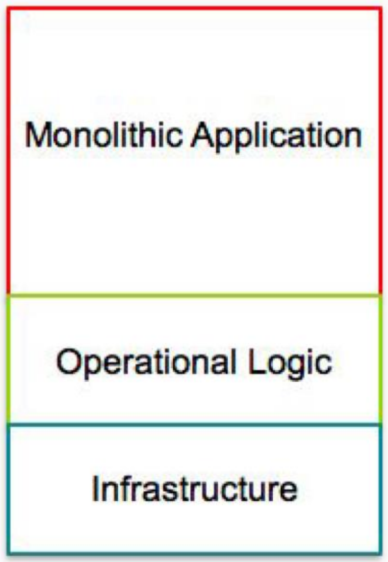
Alexandru
Iosup

Serverless / FaaS Execution

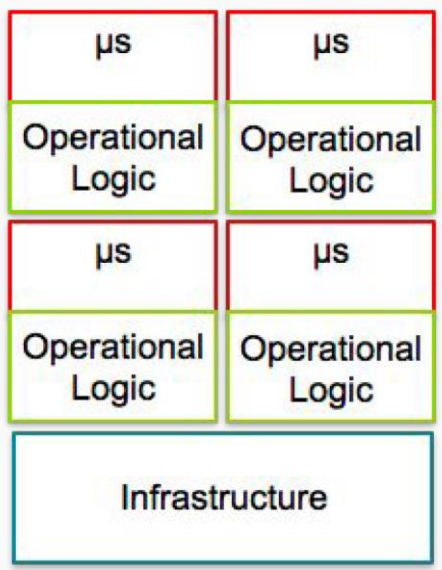
Vision and Architecture for Serverless Execution in Cloud Environments

Erwin van Eyk, Simon Seif (SAP), Markus Thoemmes (IBM Germany), Alexandru Iosup. The SPEC Cloud Group's Research Vision on FaaS and Serverless Architectures. Submitted to Workshop on Serverless Computing (WoSC'17), held in conjunction with Middleware'17.

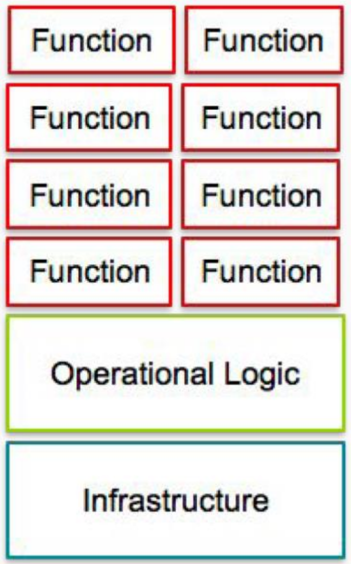
From Monoliths to Microservices to FaaS



- Difficult to Scale
- Inflexible
- Infrequent
- Complex deployment
- Tightly coupled stack



- Scalable
- Frequent
- Flexible
- Complexity: from application logic to operational logic.
- Need for DevOps



- Scalable
- Frequent
- Fexible
- Explicit separation of Business Logic vs. Operational Logic.
- Minimal layer coupling, unit of deployment

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserve

Why Research Microservice and FaaS Deployments?

- Growing industry-driven adoption.
- Current approaches are still wasteful.
- Far more logic delegation to the (cloud) infrastructure.
- New technologies, old issues:
  - Orchestration and scheduling
  - Versioning
  - Testing, benchmarking, etc.

© 2017 Alexandru Iosup. All rights reserved.

FaaS + Workflows

- Promise
  - Offload communication complexity to the platform
  - For the platform: operational efficiency ("knowledge = power")
  - Encourages composition and reuse of functions
  - Other performance improvements
- Use-cases (low-level)
  - Authenticate before function call
  - Data mapping before or/after function call
  - Fallback functions
- Use-cases (high-level)
  - ETL and data wrangling
  - CI/CD workloads
  - Business Processes as a Service

© 2017 Alexandru Iosup. All rights reserved.

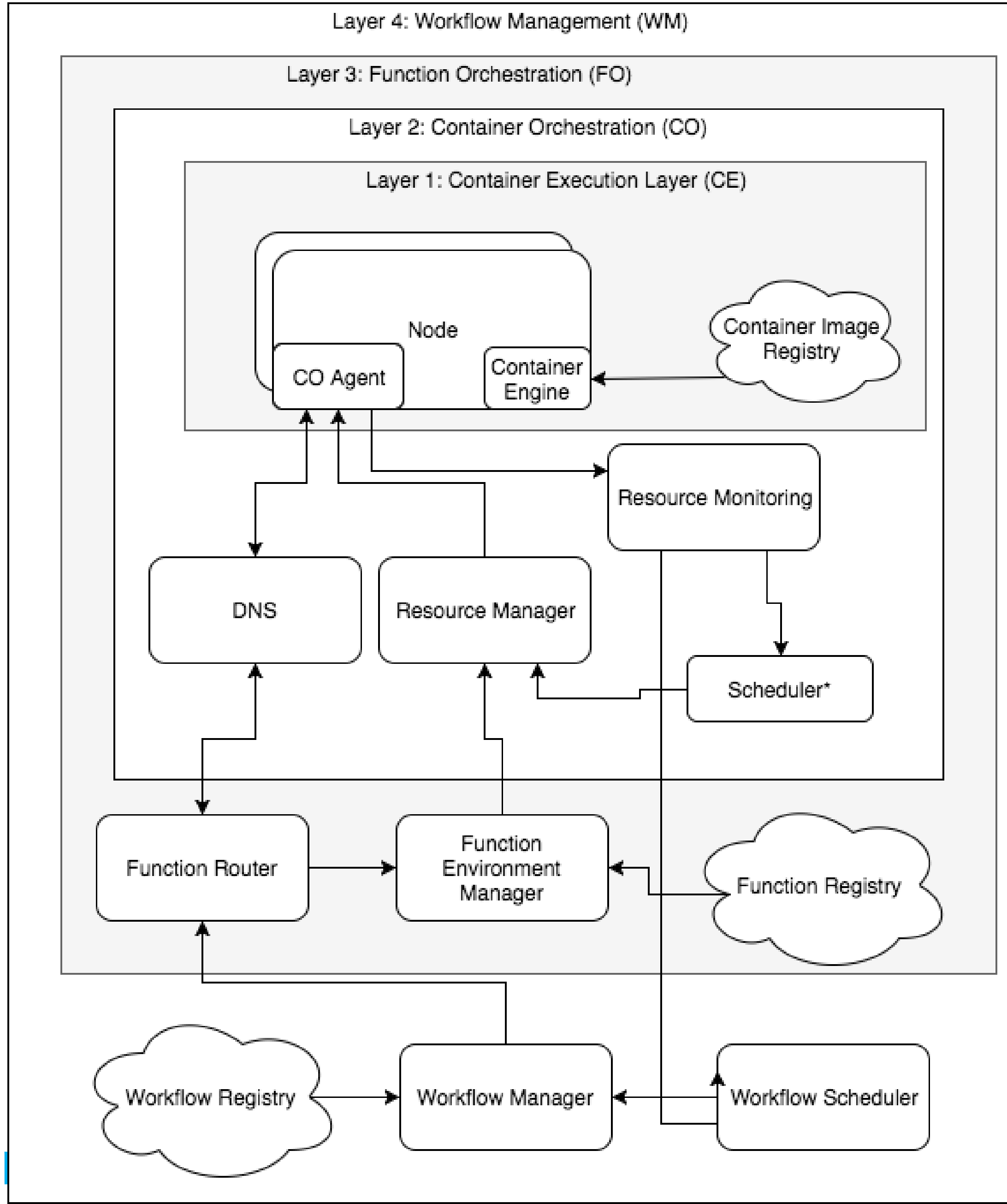
State-of-the-Art in Workflow Management

- Scientific Workflows
  - Capable resource, job, and data management, but
  - Coarse granularity
  - Pegasus (2007—ongoing), Taverna, Kepler II
- Data Processing Workflows
  - Somewhat capable resource and job management
  - Capable data management
  - Typically coarse granularity
  - Hadoop (2011—ongoing)
  - Luigi (2012), Airflow (2014)
- Cloud Workflows
  - Ports of the other workflows
  - Basic resource/job/data mgmt.
  - Fine-grained
  - AWS Step Functions (2016), OpenWhisk Sequences (2017), Azure Logic Apps (2017)

© 2017 Alexandru Iosup. All rights reserved.

Reference Architecture for FaaS Management

1. Container Execution Layer (CE)
   - Resource management on 1 node
2. Container Orchestration (CO)
   - Management system for VMs/containers
3. Function Orchestration (FO)
   - Management system for functions
4. Workflow Management (WM)



VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup

Workflow Management Architecture in Fission.io

Designed by Erwin van Eyk during internship at Platform9, in collaboration w/ Platform9 team and Alexandru Iosup.

1 Core Function / 2 AI Server

- Exposes all actions through API

3 Event Store / 4 Projector

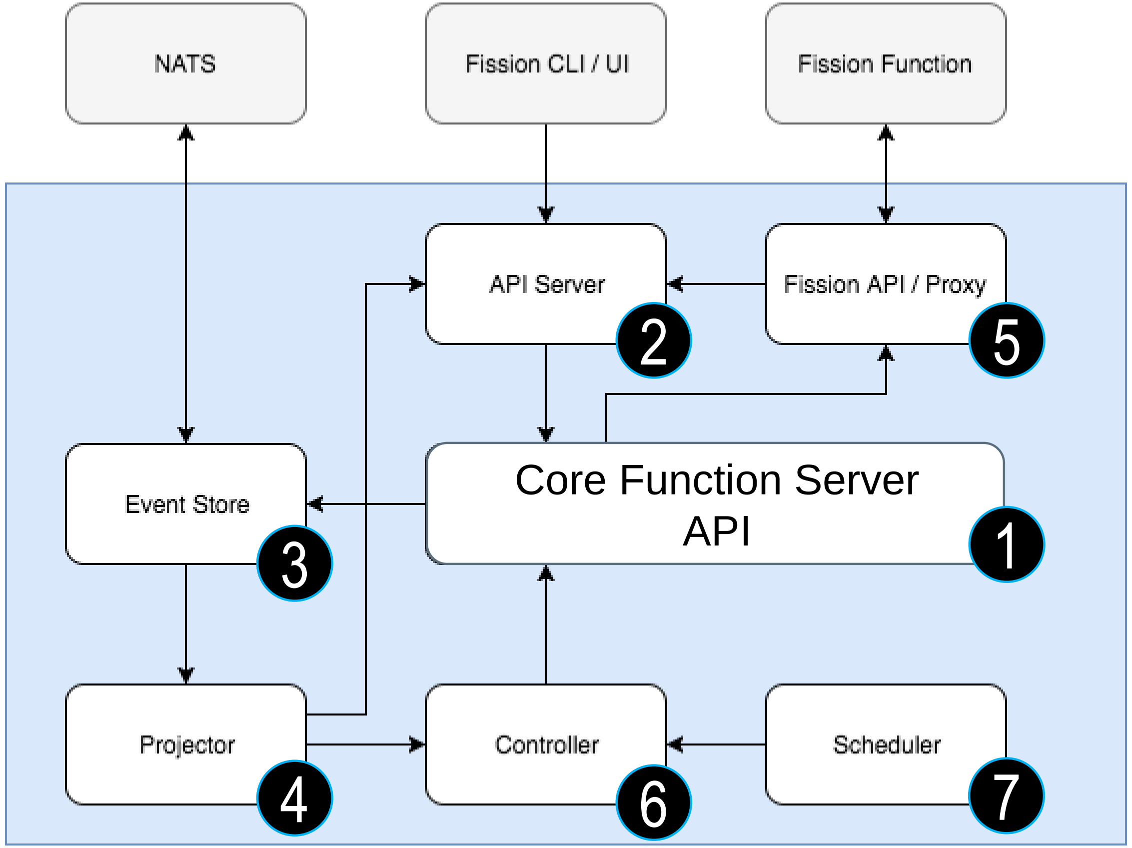
- Events update the workflow
- Store has Pub/Sub functionality
- Projector builds current state

5 Fission Proxy

- API access to Fission FaaS

6 Controller / 7 Scheduler

- Workflow manager



https://github.com/fission/fission-workflows/blob/master/Docs/architecture.md

© 2017 Alexandru Iosup. All rights reserved.

7. Meta-Management and Meta-Scheduling

- Portfolio scheduling
  - For workloads of bags-of-tasks
  - For Big Data workloads
  - For Gaming workloads
  - For DC workloads
- Self-Awareness
  - Topology identification
  - VM placement w topological constraints
  - TAGS-based scheduling w unknown task durations
- Auto-scaling / -tiering
  - Policy design
  - For workloads of workflows
  - For Gaming workloads
  - For DC workloads
- Re-configuration
  - Queue-architecture re-config
  - Delegated MatchMaking
  - Koala-C

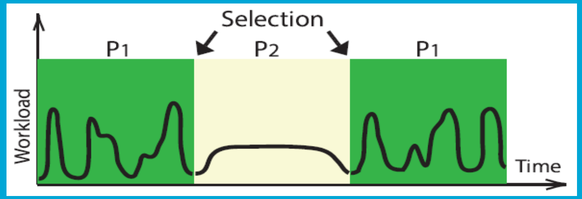
© 2017 Alexandru Iosup. All rights reserved.

Vincent
van Beek

Tim
Hegeman

Jesse
Donkervliet

Alexandru
Iosup



Portfolio Scheduling for DCs

Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters

van Beek, Donkervliet, Hegeman, Hugtenburg, Iosup. Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters. IEEE Computer 48(7): 46-54 (2015)

Deng, Song, Ren, Iosup. Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC 2013: 55:1-55:12

Massivizing Distributed Systems

Scheduling
Bags-Of-Tasks
Workflow
Mixed-Workload
Portfolio

1st time in DCs

Dependability
Failure Analysis*
Space-/Time-Correlation
Availability-On-Demand

New World
Workload Modeling
Interaction Graphs
Business-Critical
Online Gaming

Ecosystem Navigation
Performance Variability
Grid*, Cloud, Big Data
Benchmarking
Longitudinal Studies

Scalability/Elasticity
Delegated Matchmaking*
POGGI*
Area-Of-Simulation
BTWorld*
Auto-Scalers

Socially Aware Techniques
Collaborative Downloads*
Groups in Online Gaming
Toxicity Detection*

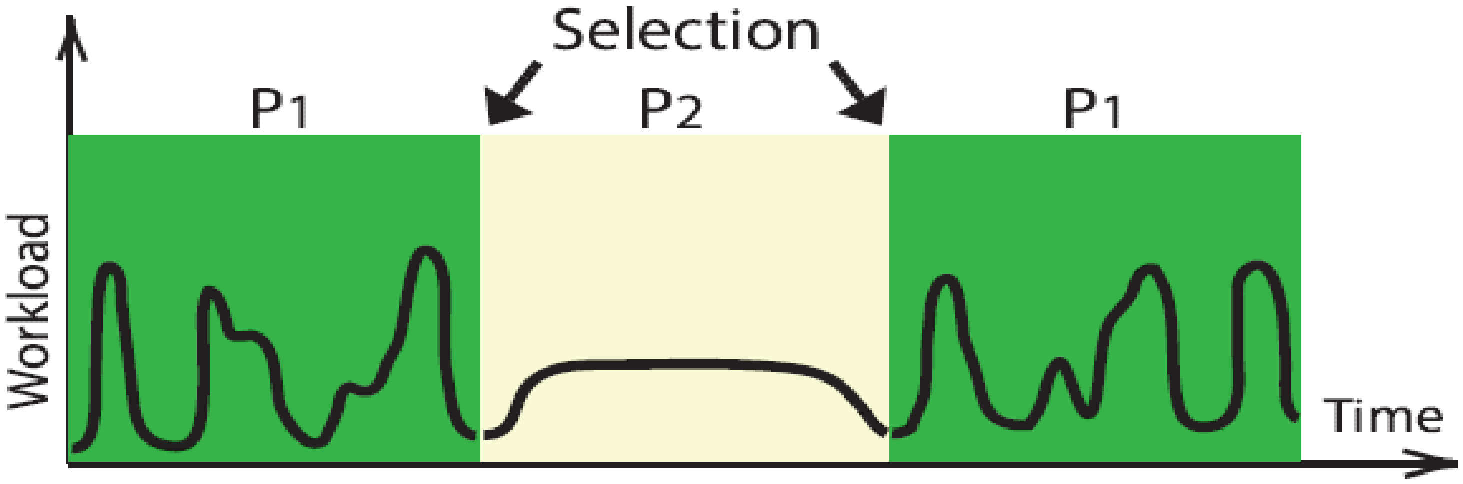
Software Artifacts
Graphalytics, etc.

Data Artifacts
A Distributed Systems Memex*

Fundamental Problems
My Contribution So Far (* Award-winning)

Portfolio Scheduling, In A Nutshell

- Datacenters cannot work without one or even several schedulers
- Instead of ephemeral, risky schedulers, we propose to



1. Create a set of schedulers (resource provisioning and allocation policies)
2. Select active scheduler online, apply for the next period, analyze results

(Repeat)

K. Deng et al. Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC|13

Portfolio Scheduling

Portfolio Scheduling for Computer Systems

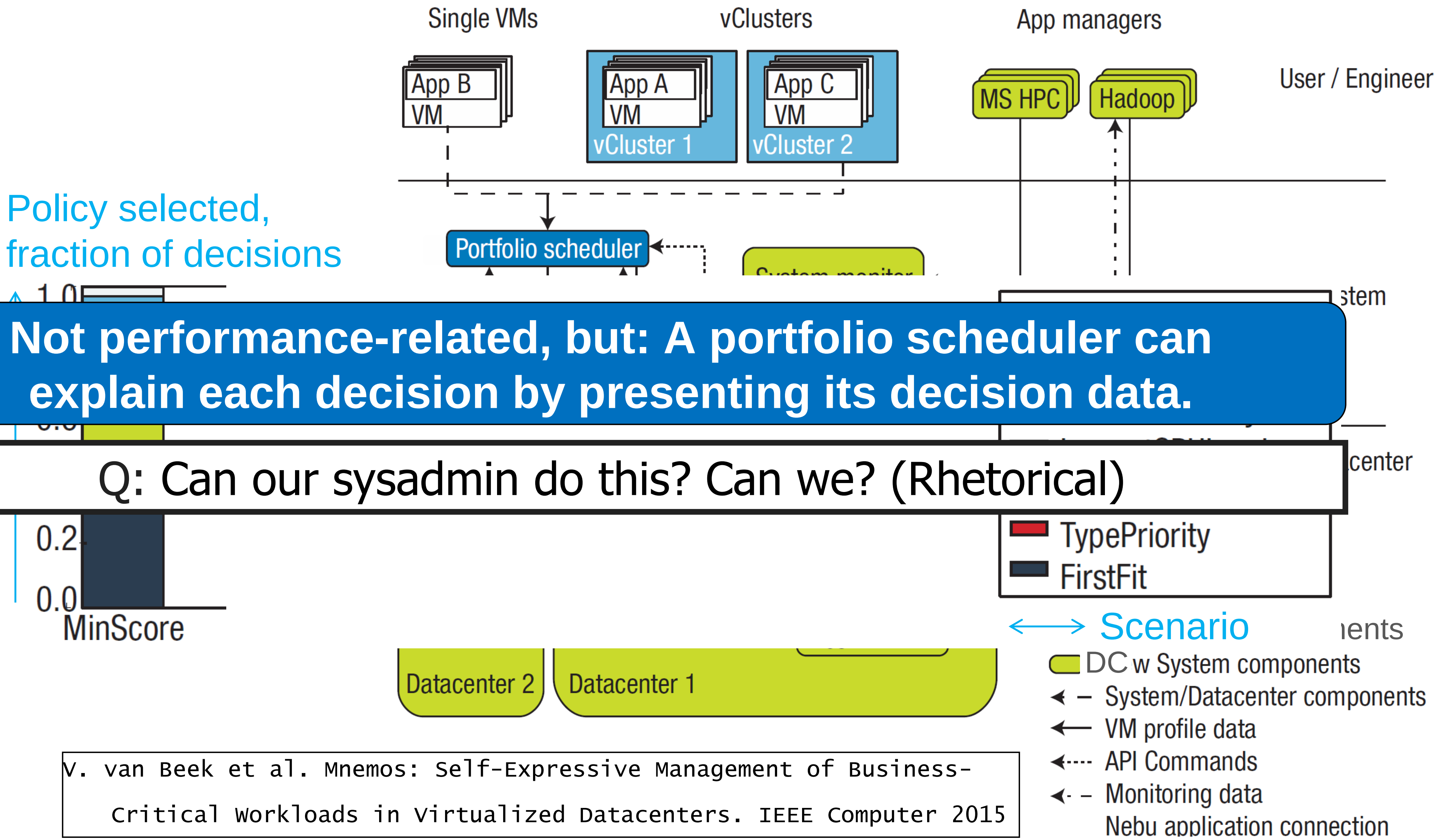
Portfolio Creation
- Configure schedulers
- 10s-100s+ schedulers

Scheduler Selection + Explanation
- Define new metrics, risk
- Consider data in the process

Application of Selected Scheduler
- Monitor system for issues

Self-Reflection on Portfolio + Scheduler
- Reflect and Adapt portfolio

Portfolio Scheduling in Practice: Massive Datacenters



Not performance-related, but: A portfolio scheduler can explain each decision by presenting its decision data.

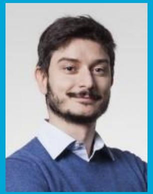
Q: Can our sysadmin do this? Can we? (Rhetorical)

V. van Beek et al. Mnemos: Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters. IEEE Computer 2015

Alexey Ilyushkin

Ahmed Ali-Eldin

Nikolas Herbst



Alessandro Papadopoulos

Bogdan Ghiț

Dick Epema

Alexandru Iosup

Auto-Scaling

Experimental Performance Evaluation of Autoscaling Policies for Complex Workflows

Best Paper Candidate

Ilyushkin, Ali-Eldin, Herbst, Papadopoulos, Ghit, Epema, Iosup. An Experimental Performance Evaluation of Autoscaling Policies for Complex Workflows. ICPE 2017

Massivizing Distributed Systems

Scheduling
Bags-Of-Tasks
Workflow
Mixed-Workload
Portfolio

Dependability
Failure Analysis*
Space-/Time-Correlation
Availability-On-Demand

New World
Workload Modeling
Interaction Graphs
Business-Critical
Online Gaming

Ecosystem Navigation
Performance Variability
Grid*, Cloud, Big Data
Benchmarking
Longitudinal Studies

Scalability/Elasticity
Delegated Matchmaking*
POGGI*
Area-Of-Simulation
BTWorld*
Auto-Scalers
A

Socially Aware Techniques
Collaborative Downloads*
Groups in Online Gaming
Toxicity Detection*

Software Artifacts
Graphalytics, etc.

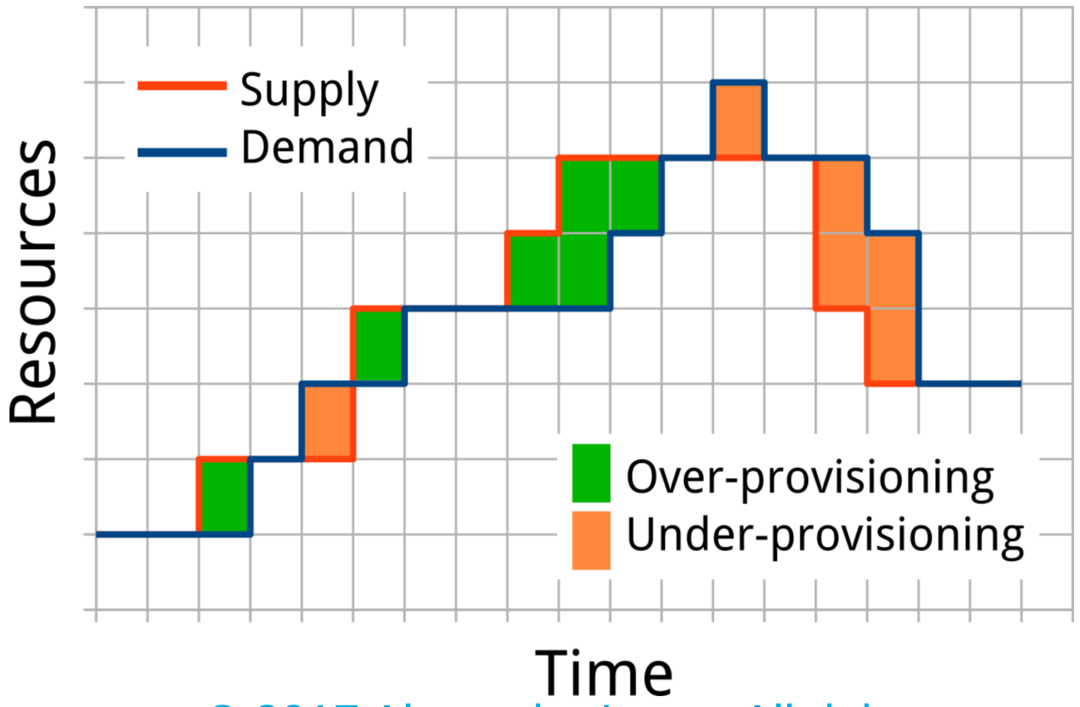
1st real-world comparative study on workflow scheduling

Fundamental Problems

My Contribution So Far (* Award-winning)

What is an Autoscaler?

An **autoscaler** *automatically* provisions and releases resources according to demand



VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Our Approach

A comprehensive method for evaluating and comparing autoscalers

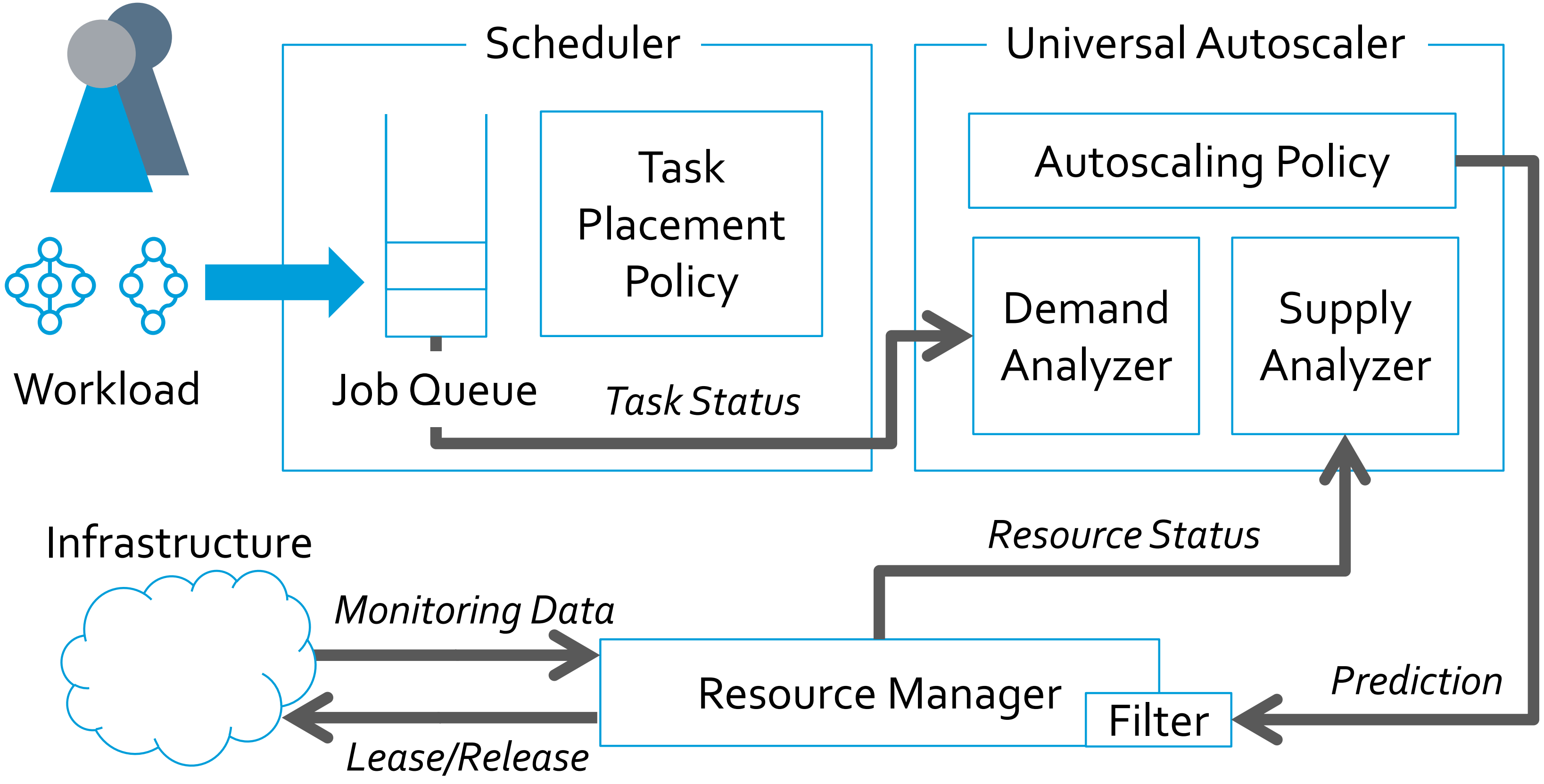
- A **model** for elastic cloud platform
- A set of relevant **metrics** for assessing autoscaler performance
- A set of general and workflow-specific **autoscalers**
- Three **comparison methods** for autoscalers
- **Real experiments** with up to 50 VMs in OpenNebula on DAS supercomputer

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Elastic Cloud Platform



VU VRIJE UNIVERSITEIT AMSTERDAM

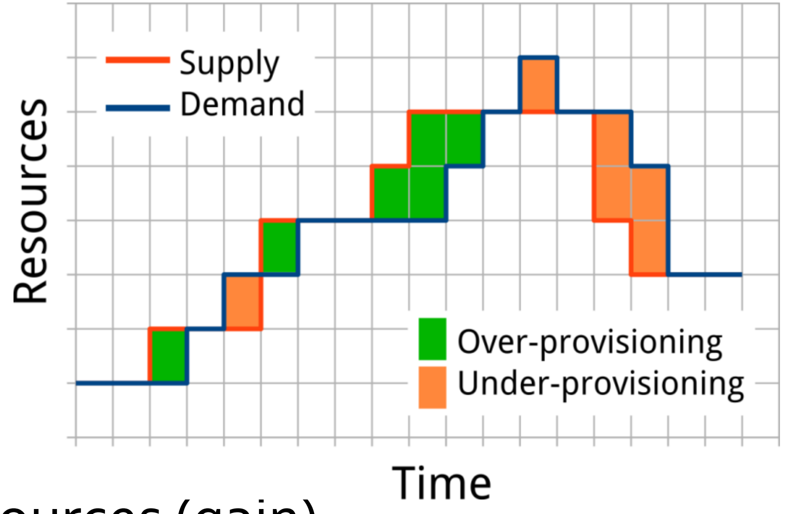
© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Performance Metrics

System-oriented elasticity metrics

- Accuracy (also normalized by actual demand)
- Wrong-Provisioning Timeshare
- Instability



User-oriented metrics

- Elastic Slowdown
- Average Number of Utilized Resources (gain)
- Average Throughput (tasks/h)

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Autoscaling Policies

Timeliness of the Information

	Long-term	**Current/Recent**
Server	Hist, Reg, ConPaaS	React, Adapt
Job	Plan	Token

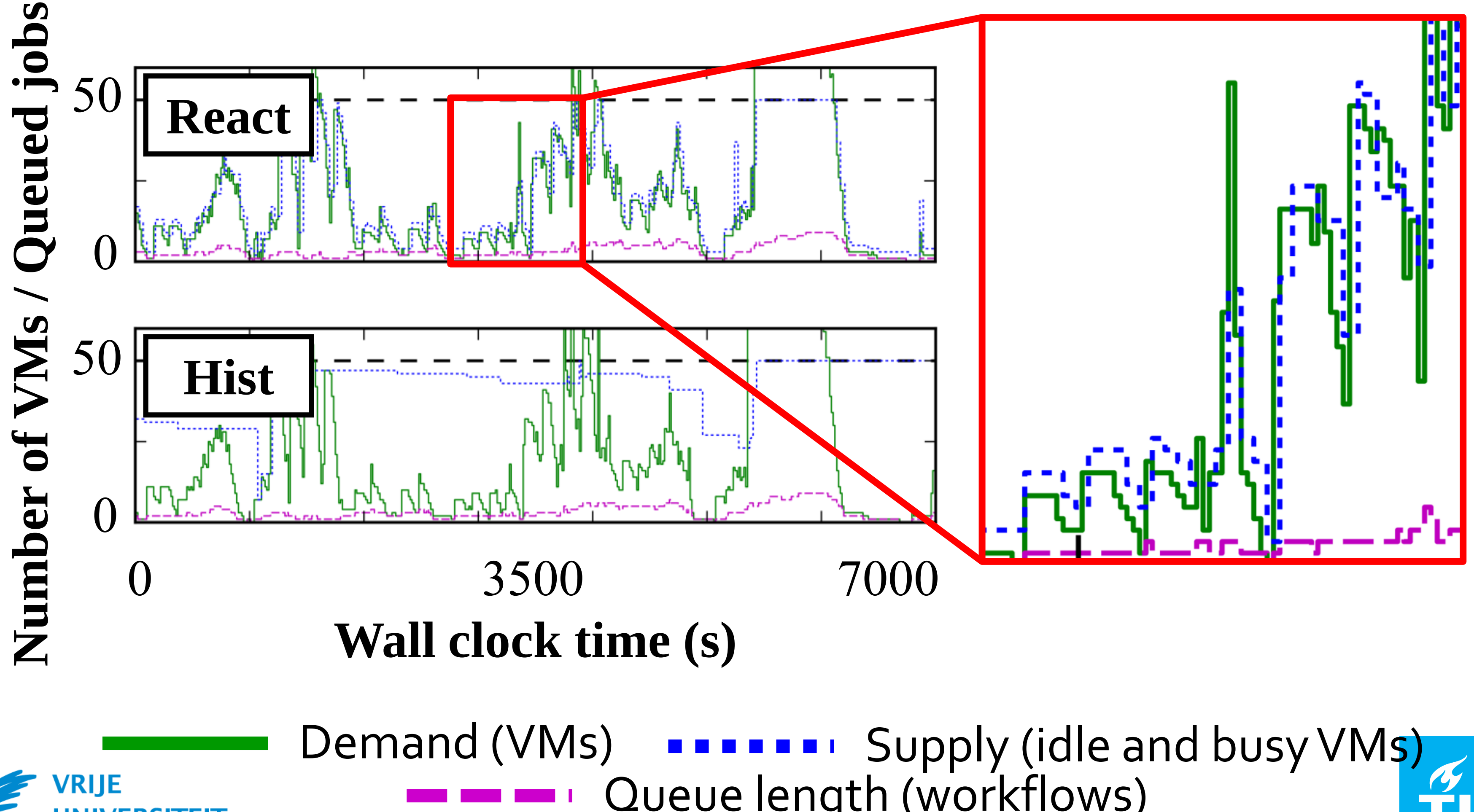
Information Source

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Experimental Results



© 2017 Alexandru Iosup. All rights reserved.

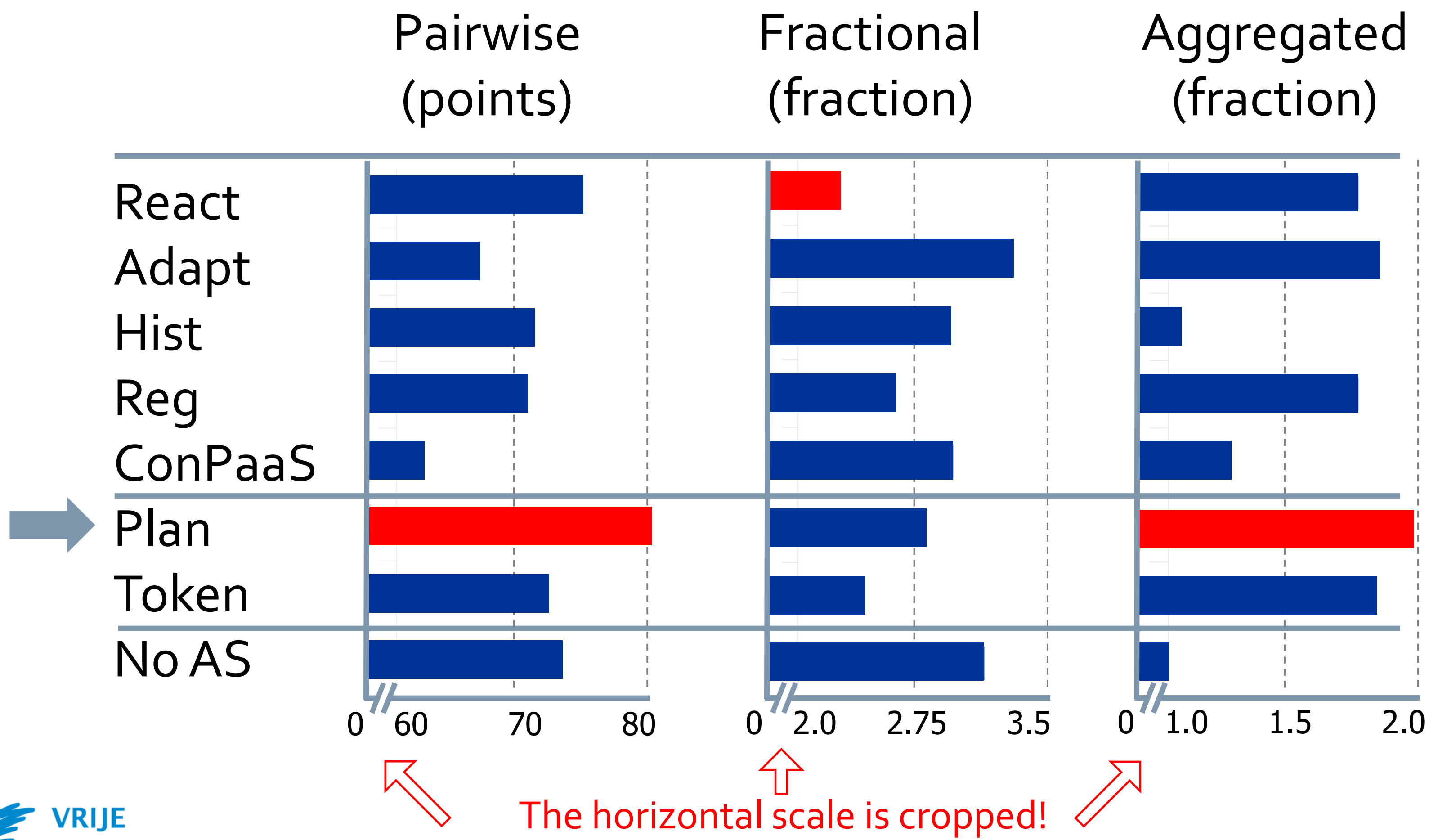
Which Policy is the Best?

Methods for aggregation of metrics

- **Pairwise Comparison** – pairwise compare metrics between autoscalers
- **Fractional Difference Comparison** – compare autoscalers with an ideal case based on the experimental results
- **Aggregated System-oriented Elasticity and User Metrics** (by Fleming et al.)
  Compute speedup ratios and then average the speedups using an unweighted geometric mean

© 2017 Alexandru Iosup. All rights reserved.

Which Policy is the Best?



© 2017 Alexandru Iosup. All rights reserved.

Conclusion

1. We developed a method to compare different autoscalers
2. General autoscalers can achieve similar performance as workflow-specific autoscalers (surprising)
3. No autoscaler is the best:
   Our workflow-specific Plan autoscaler wins 4 out of 5 competitions but is not the best overall
4. The correct choice of an autoscaler is important but significantly depends on the application type
5. Correct parameterization of general autoscalers is very important

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

8. Multi-DC Management and Scheduling

8 Multi-DC | Hybrid Cloud | Federated Clouds | Multi-Cloud / -Grid/ -Cluster

- Federated Clouds/Grids
  - Delegated MatchMaking architecture
  - Hierarchical / Distributed architectures
  - For Bags of Tasks
  - Condor Delegated MatchMaking
- Multi-cluster operation
  - Koala-C
- Hybrid cloud operation
  - With workload migration
  - With workload replication
  - For Bags of Tasks
  - ExPERT system

© 2017 Alexandru Iosup. All rights reserved.

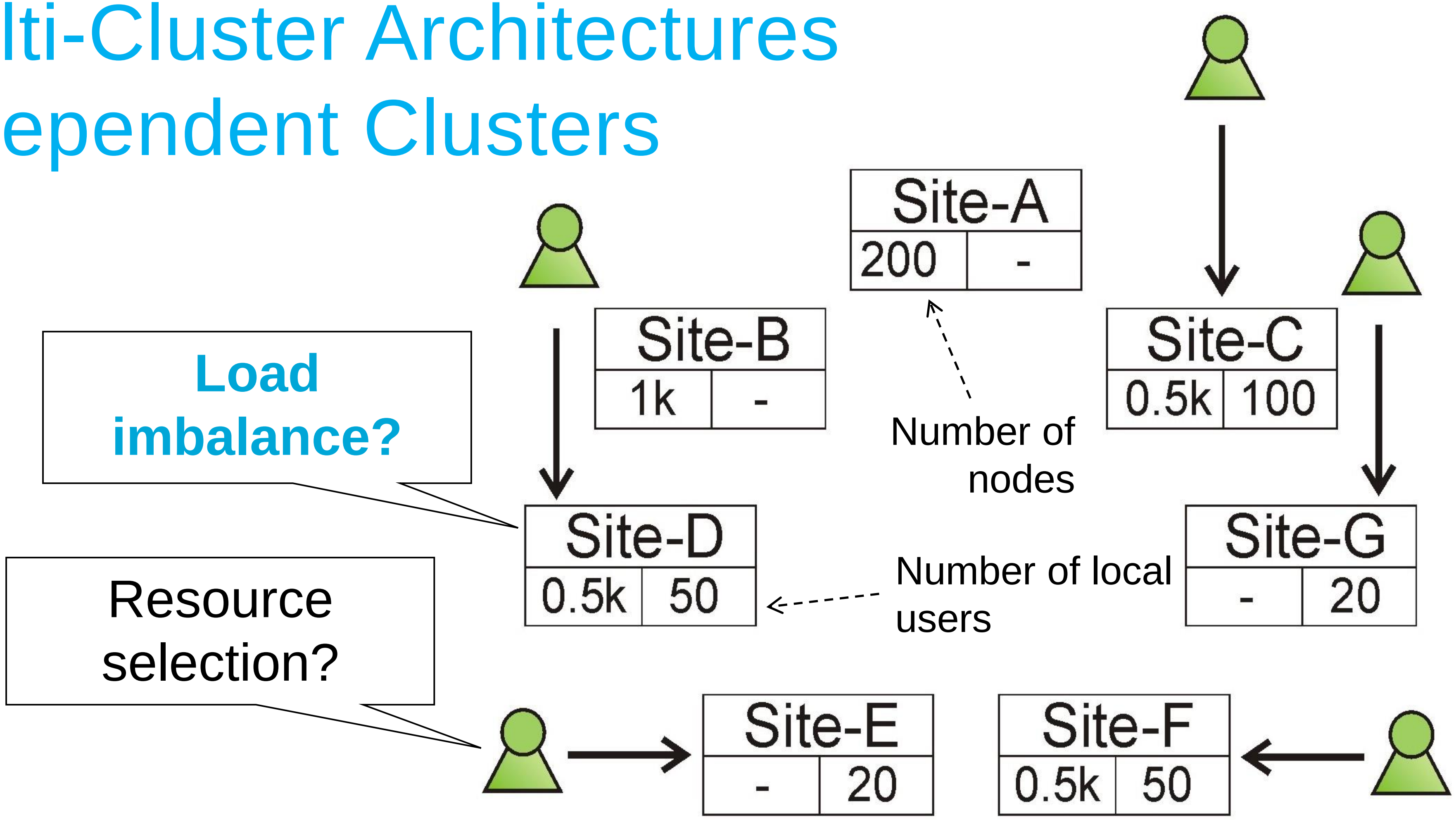
Is there a case for heterogeneous inter-datacenter computing in scientific workloads?

Condor Delegated MatchMaking

Dynamic Load Balancing for High-Performance Graph Processing on Hybrid CPU-GPU Platforms

Iosup, Epema, Tannenbaum, Farrellee, Livny. Inter-operating grids through delegated matchmaking. SC 2007. Nominated for Best Paper Award, Best Student-Paper Award.

Multi-Cluster Architectures
Independent Clusters



Observational scheduling

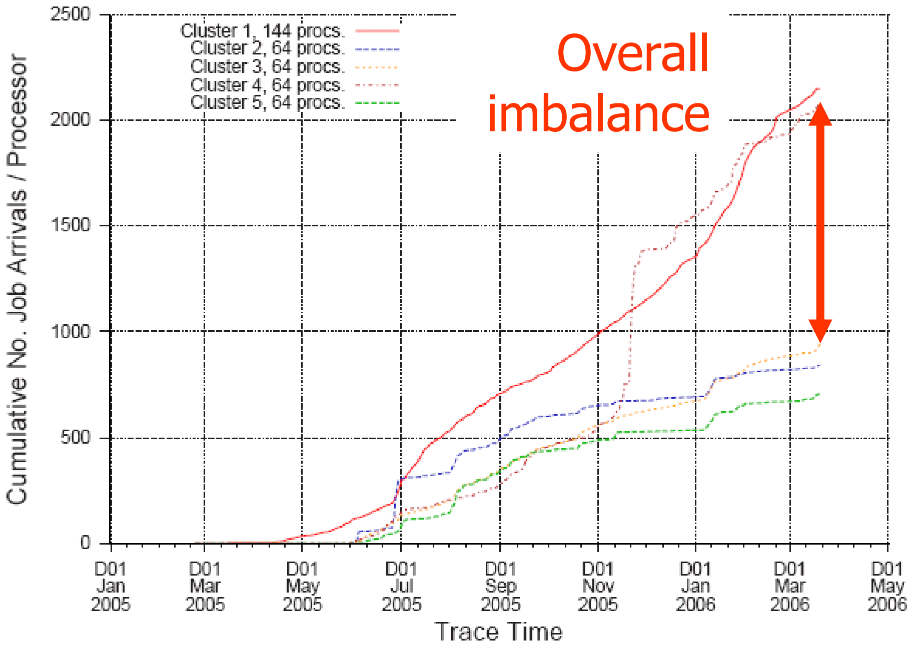
VRIJE UNIVERSITEIT AMSTERDAM

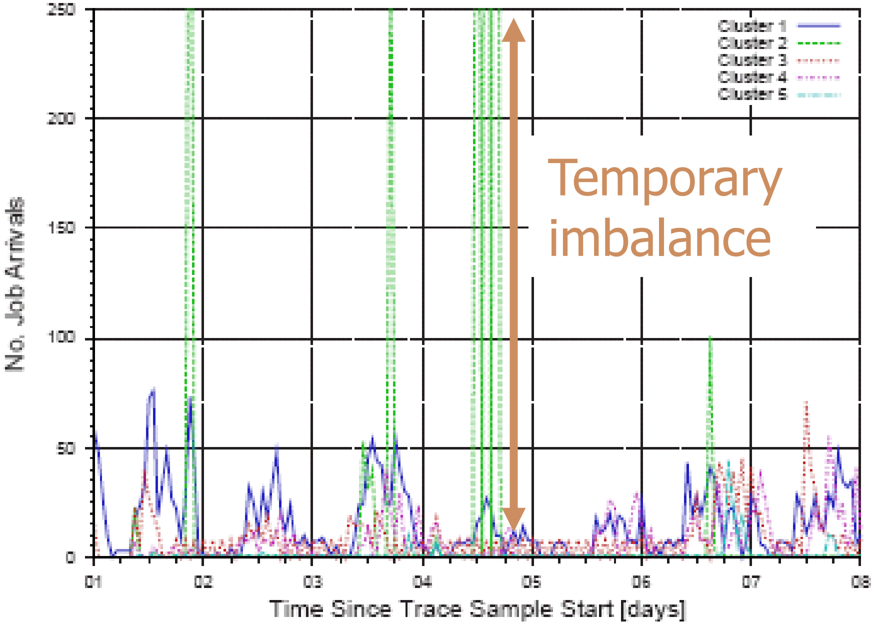
© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Load Imbalance in Independent Clusters

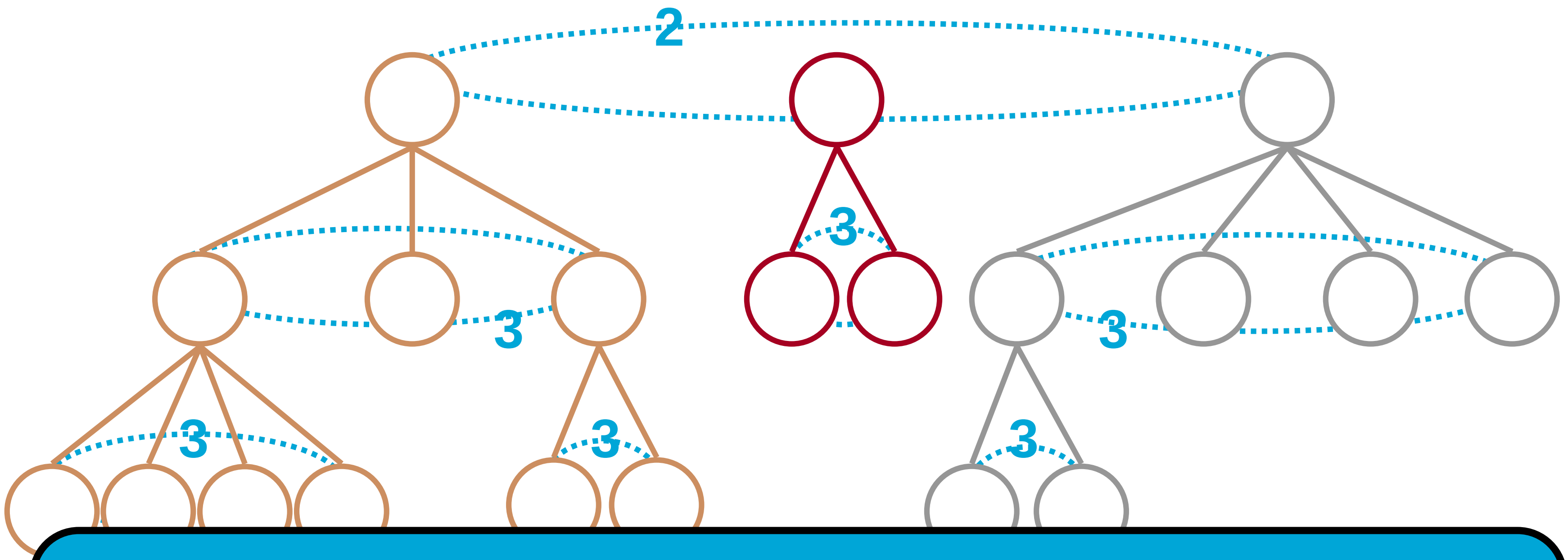
- **Overall workload imbalance**: normalized daily load (5:1)
- **Temporary workload imbalance**: hourly load (1000:1)





© 2017 Alexandru Iosup. All rights reserved.

The Delegated MatchMaking Architecture



1. Sta
2. Let
3. Let

Delegated MatchMaking Architecture= Hybrid hierarchical/decentralized architecture for grid inter-operation

VU VRIJE

Alexandru Iosup, Dick H. J. Epema, Todd Tannenbaum, Matthew Farrellee, Miron Livny: Inter-operating grids through delegated matchmaking. SC 2007: 13. Nominated for Best Paper Award.

rved.

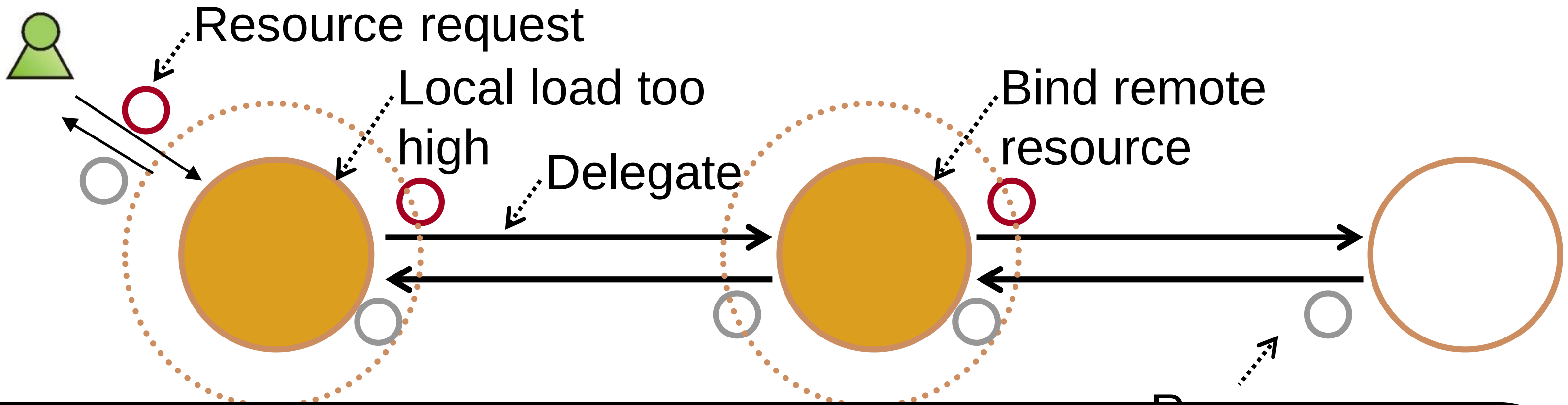
TU Delft

The Delegated MatchMaking Mechanism

Q: Complexity of this approach?

Q: Who controls the delegation?

Delegated MatchMaking



1. Dea
2. Wh ... ocal env't.:
   •
   •
3. Del

The Delegated MatchMaking Mechanism=
Delegate Resource Usage Rights,
Do Not Delegate Jobs

VRIJE

Alexandru Iosup, Dick H. J. Epema, Todd Tannenbaum, Matthew Farrellee, Miron Livny:
Inter-operating grids through delegated matchmaking. SC 2007: 13. Nominated for Best Paper Award.

rved.

Delegated MatchMaking vs. Others

(Higher is better)

- DMM
- Decentralized
- Centralized
- Independent

Delegated MatchMaking delivers good performance

- Low wait time
- Finishes all jobs
- Even better for load imbalance between grids
- Reasonable overhead

Alexandru Iosup, Dick H. J. Epema, Todd Tannenbaum, Matthew Farrellee, Miron Livny: Inter-operating grids through delegated matchmaking. SC 2007: 13. Nominated for Best Paper Award.

VU
TU Delft
© 2021 Alexandru Iosup. All rights reserved.

Koala-C

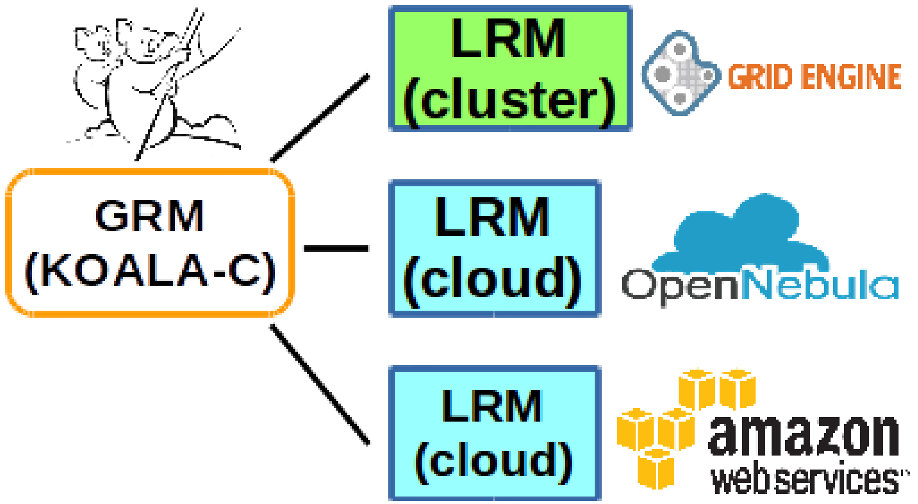
A task allocator for integrated multicluster and multicloud environments

Fei, Ghit, Iosup, Epema. KOALA-C: A task allocator for integrated multicluster and multicloud environments. CLUSTER 2014: 57-65

KOALA-C: integrated multicluster and multicloud environment

Extend local cluster infrastructure with on-demand cloud resources.

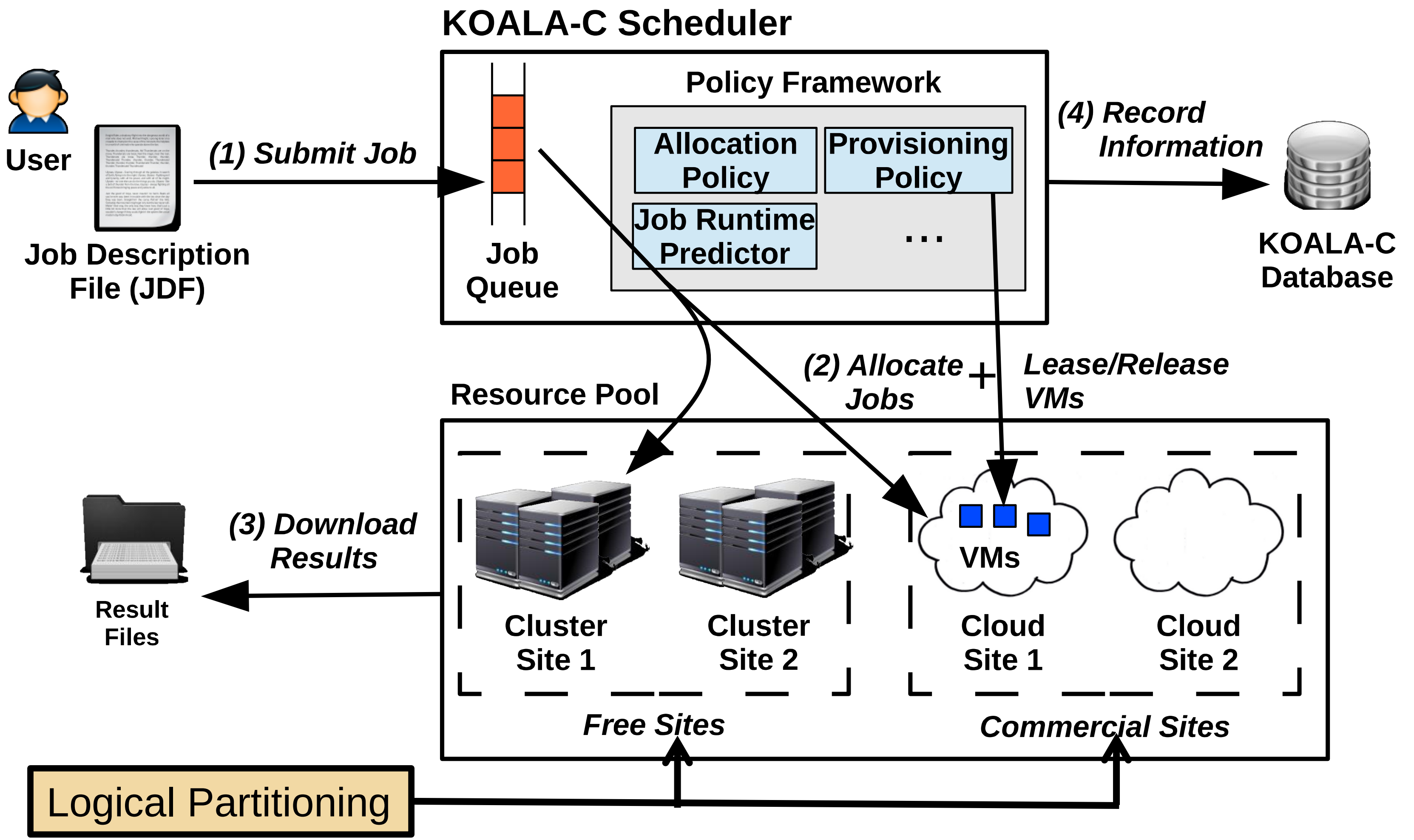
Logically partitioning of resources to isolate jobs of different sizes.



L. Fei, B.I. Ghit, A.Iosup, and D.H.J. Epema, "KOALA-C: A Integrated Multicluster and Multicloud Environment" *IEEE CLUSTER 2014*

© 2017 Alexandru Iosup. All rights reserved.

The KOALA-C Scheduler



VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

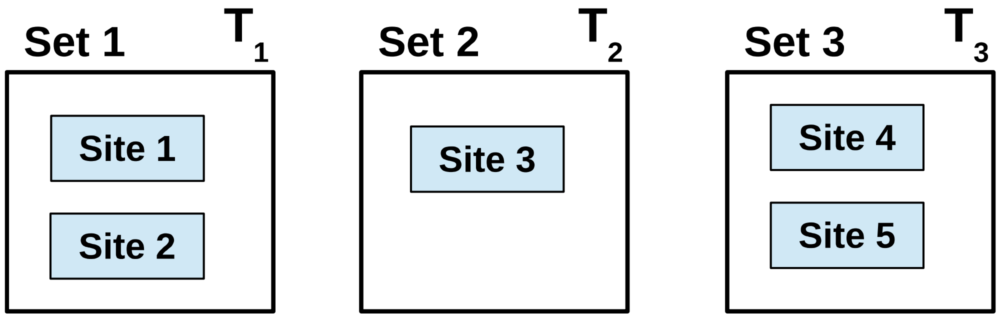
TU Delft

TAGS-based Policy Design

Achieve low slowdown **without prediction.**

Partition the sites into sets to serve jobs of different runtime ranges:

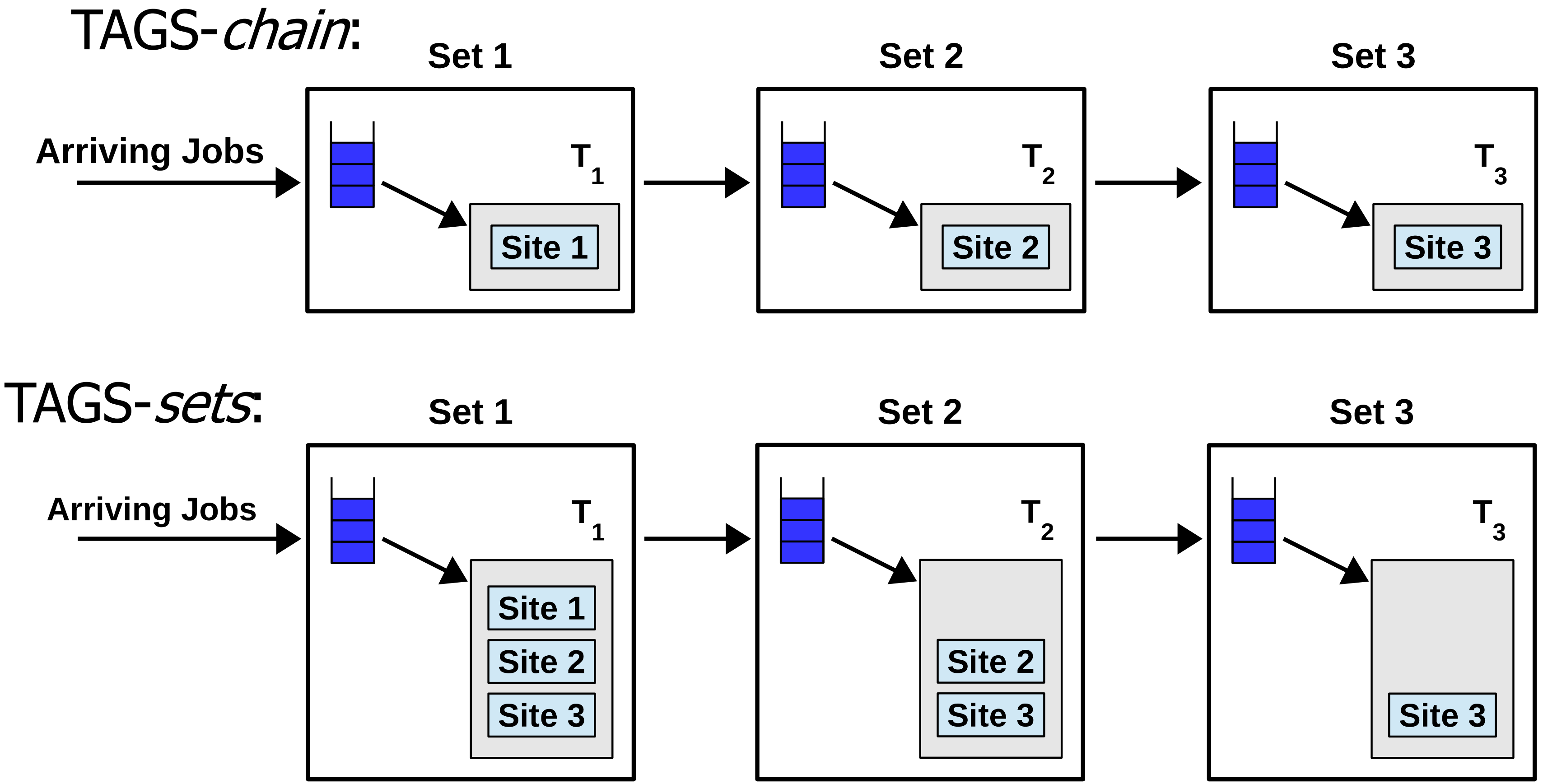
- A number of sets of sites
- Set i allows jobs to run for T_i amount of time ($T_i < T_{i+1}$)
- The last set has a T of ∞ (all jobs will finish without being killed)



$T_1 < T_2 < T_3$
$T_3 = \infty$

© 2017 Alexandru Iosup. All rights reserved.

Policy Design TAGS-*chain* and TAGS-*sets*



© 2017 Alexandru Iosup. All rights reserved.

Experimental Setup

Resources: 2 sites of the DAS-4 system (32 nodes each).

Cloud: OpenNebula-based private cloud of DAS-4 (up to 32 VMs)

Amazon EC2 as public cloud (up to 64 VMs).

Workload: A part of the CTC-SP2 workload (≈12 hours), CPU-intensive jobs.

70% average utilization on the system (with the max cloud size).

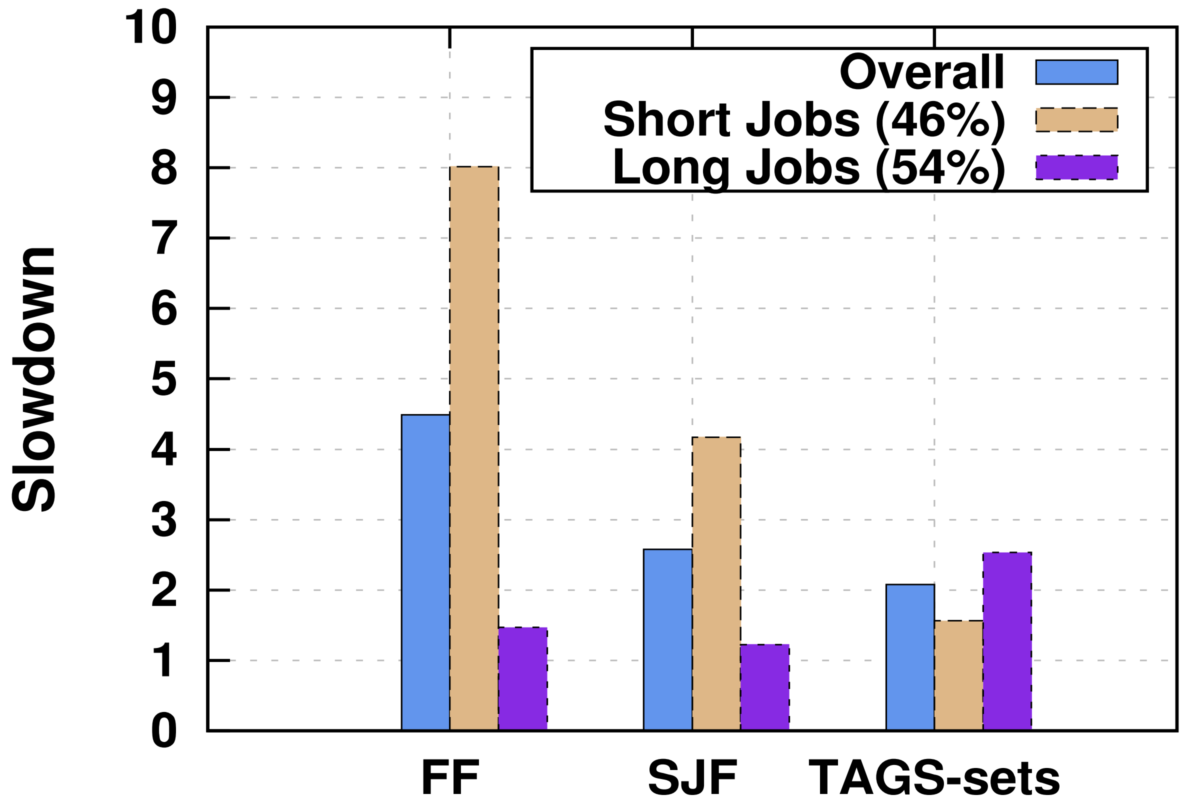
Policies: FF, SJF, and TAGS-*sets.*

≤10min **Short Job Set**	**Long Job Set**
ALL sites	**ALL without Cluster 1**

[4] James Patton Jones and Bill Nitzberg, "Scheduling for Parallel Supercomputing: A Historical Perspective of Achievable Utilization", 1999

© 2017 Alexandru Iosup. All rights reserved.

Experimental Results



TAGS-sets has better short-job and overall slowdown, at the expense of long jobs

© 2017 Alexandru Iosup. All rights reserved.

Orna Agmon-Ben Yehuda
Technion

Alexandru
Iosup

Dick
Epema

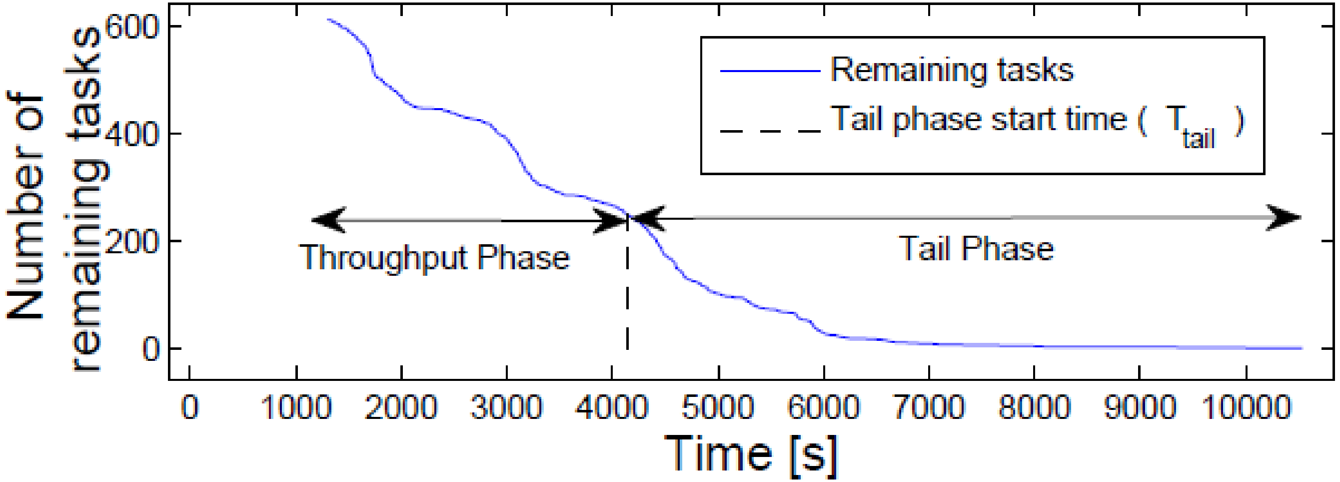
ExPERT cloud scheduler

Pareto-efficient replication of tasks to run Bags-of-Tasks workloads in hybrid clouds

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Helping the User Select with ExPERT: Pareto-efficient Replication of Tasks

Workload: Bags of Tasks



Environment

- Reliable nodes = (slow, no failure free)
- Unreliable nodes = (fast, failures, costly)

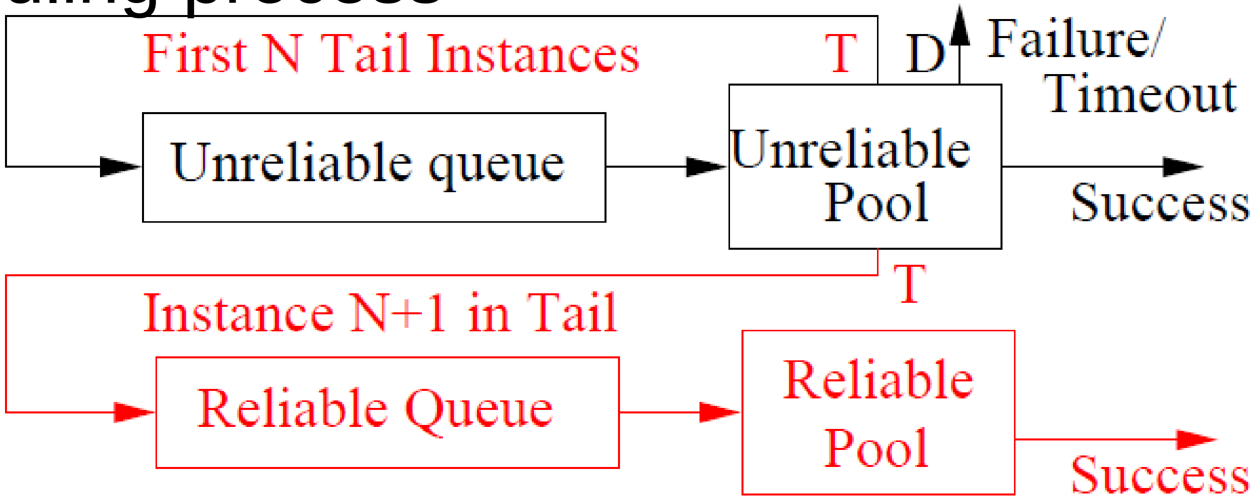
VU VRIJE UNIVERSITEIT AMSTERDAM

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

TU Delft

Our Replication Mechanism

Scheduling process

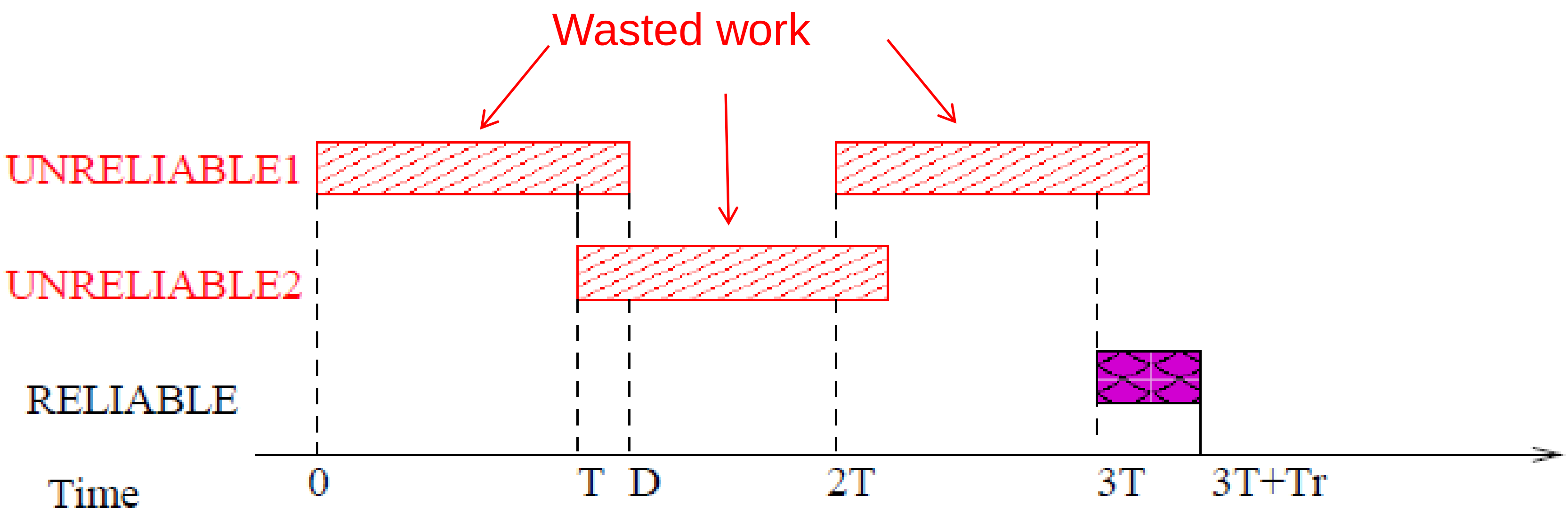


Scheduling policy = (N,T,D,Nr) tuple

- N—how many times to replicate on <u>un</u>reliable?
- T—when to replicate?
- D—task instance deadline
- Nr—max ratio reliable:unreliable

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

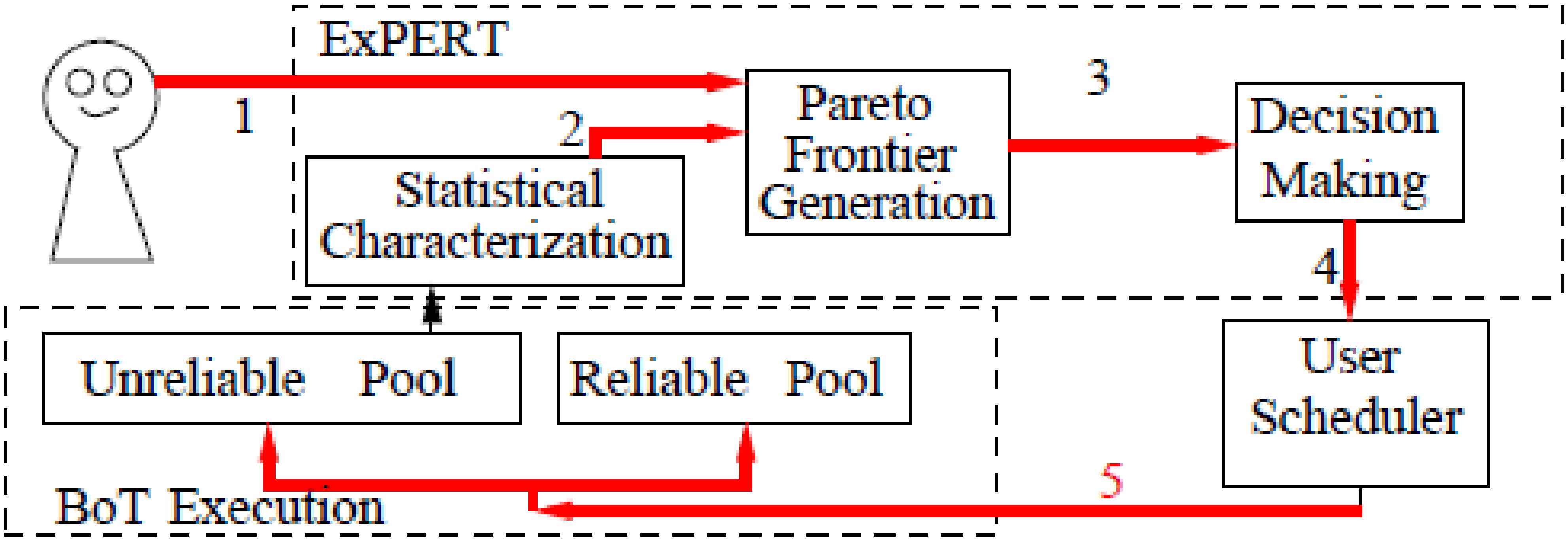
An Example with 1 Task, 2 Unreliable+1 Reliable Systems



Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

The ExPERT Policy* Recommender

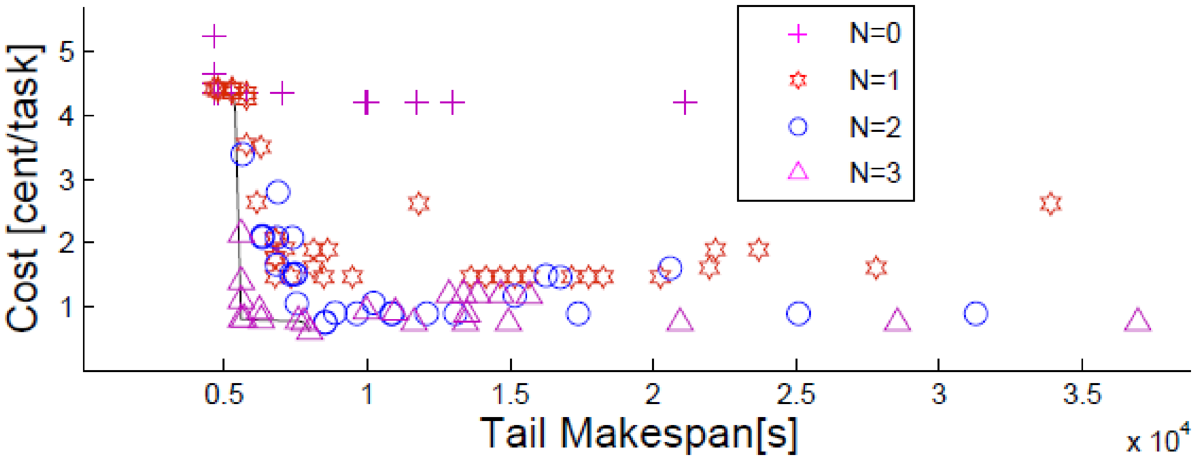
* = (N,T,D,Mr) tuple

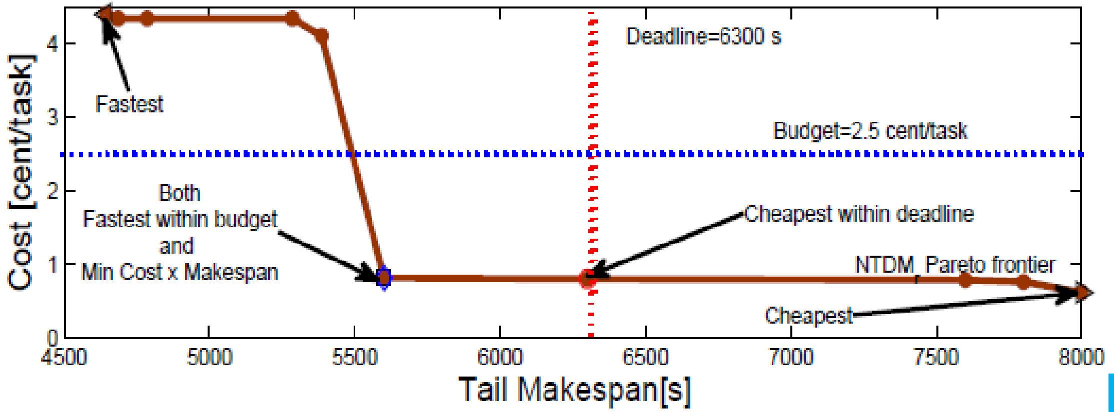


Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

Anecdotal Features, Real-System Traces

- Non-Pareto (unoptimized) policies are wasteful
- Optimization non-trivial, many options
- Choice of policies at runtime: online interpretation of offline results, *point-and-click*





Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

ExPERT in Practice

Environment

Reliable Pool	Properties
Technion	20 self-owned CPUs in the Technion.
EC2	20 large EC2 cloud instances.

Unreliable Pool	Properties
UW-M	UW-Madison Condor pool (preempts).
OSG	Open Science Grid (no preemption).
UW-M + OSG	Combined: half $\sharp ur$ from each pool.
UW-M + EC2	Combined: 200 UW-M, 20 EC2.
UW-M + Technion	Combined: 200 UW-M, 20 Technion.

Workload

- Bioinformatics workloads, previously launched with GridBot

VRIJE UNIV AMS

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

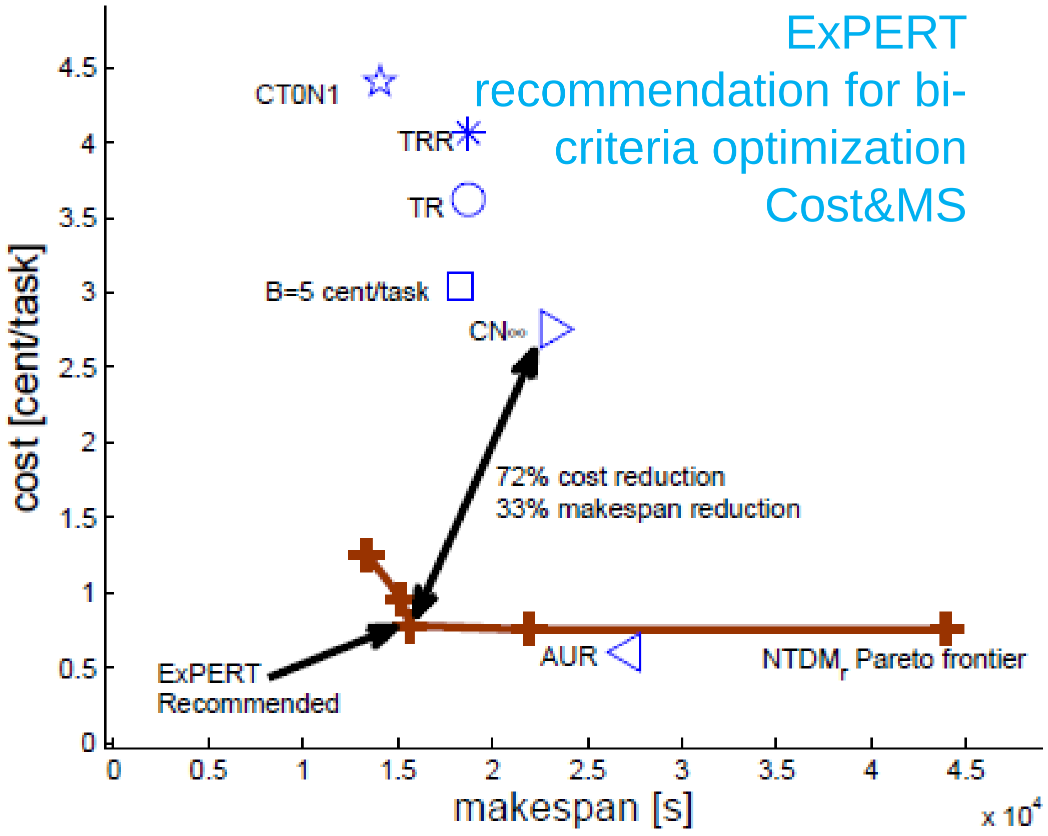
Delft

ExPERT in Practice

- D—task instance deadline
- T—when to replicate?
- N—how many times to replicate on unreliable?
- Nr—max ratio reliable:unreliable

Policies

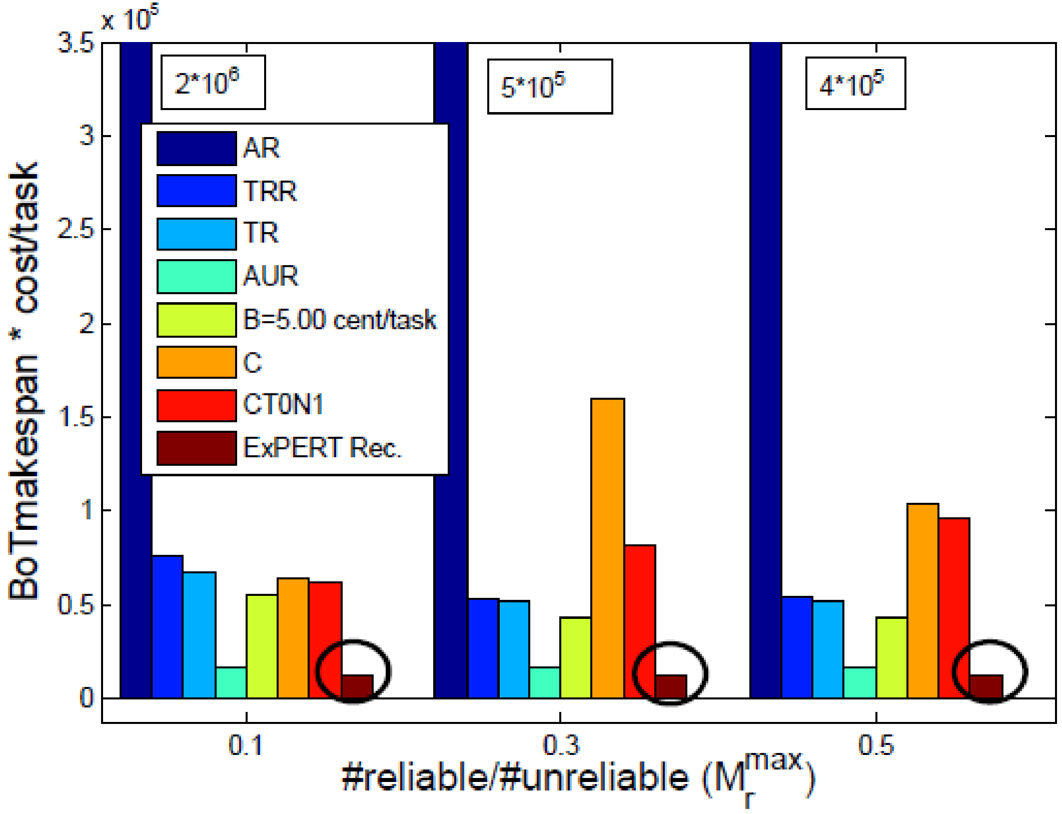
- AR—all to reliable
- AUR—all to unreliable, no replication
- TRR—Tail Replicate immediately to Reliable (N=0,T=0)
- TR—Tail to Reliable (N=0,T=D)
- CNinf—combine resources, no replication
- CT0N1—combine resources, replicate immediately at tail, N=1
- B=*cents/task—budget



ExPERT recommendation for bi-criteria optimization Cost&MS

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS’12.

ExPERT for utility U = Cost x MakeSpan: 25% better than 2nd-best, 72% better than 3rd-best



VRIJE UNIVERSITEIT AMSTERDAM

Agmon Ben-Yehuda, Schuster, Sharov, Silberstein, Iosup. ExPERT: pareto-efficient task replication on grids and a cloud. IPDPS'12.

TU Delft

9. Workload Specification

Game	Business	Eng	Game	Scientific	9 Workload Specification	CUPs	Workflows	Bags-of-Tasks
	Graph	P2P	Big Data	Cloud/Grid		SLAs	Non-functional requirements	

- CUPs and SLAs
  - Specification of cloud scenarios
  - Specification of SLAs, including penalties for non-compliance
  - Utility functions
  - SPEC CUP specification
  - ExPERT scheduler
- Workflows with Functional & Non-Functional Requirements
  - Performance, Availability, Elasticity, Security
  - Requirements changing over time
  - Soft guarantees

© 2017 Alexandru Iosup. All rights reserved.

Cloud Usage Patterns

A task allocator for integrated multicluster and multicloud environments

Milenkoski, Iosup, Kounev, Sachs, Mularz, Curtiss, Ding, Rosenberg, and Rygielski. CUP: A Formalism for Expressing Cloud Usage Patterns for Experts and Non-Experts. IEEE Cloud Computing, 2017 (in print)

CUP slides originally by Aleksandar Milenkoski, with help from co-authors.

Cloud Usage Patterns

What cloud services exist? Abstract answer:

- SLA-based services
- Value chains
- Value chains with mediators
- Hybrid service provisioning
- ...

How to represent them? Through formal, textual and/or visual descriptions

https://www.ogf.org/ogf/doku.php/documents/documents

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

Aleksandar Milenkoski, Alexandru Iosup, Samuel Kounev, Kai Sachs, Piotr Rygielski, Jason Ding, Walfredo Cirne, and Florian Rosenberg. Cloud Usage Patterns: A Formalism for Description of Cloud Usage Scenarios. Technical Report SPEC-RG-2013-001 v. 1.0, SPEC Research Group - Cloud Working Group, April 2013. https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.

Cloud Usage Patterns: Usage and Benefits

Potential and actual cloud users:
Specification of service requirements

Cloud system designers:
Identification of frequently used cloud service patterns

Researchers and consultants:
Classification and comparison of cloud usage scenarios

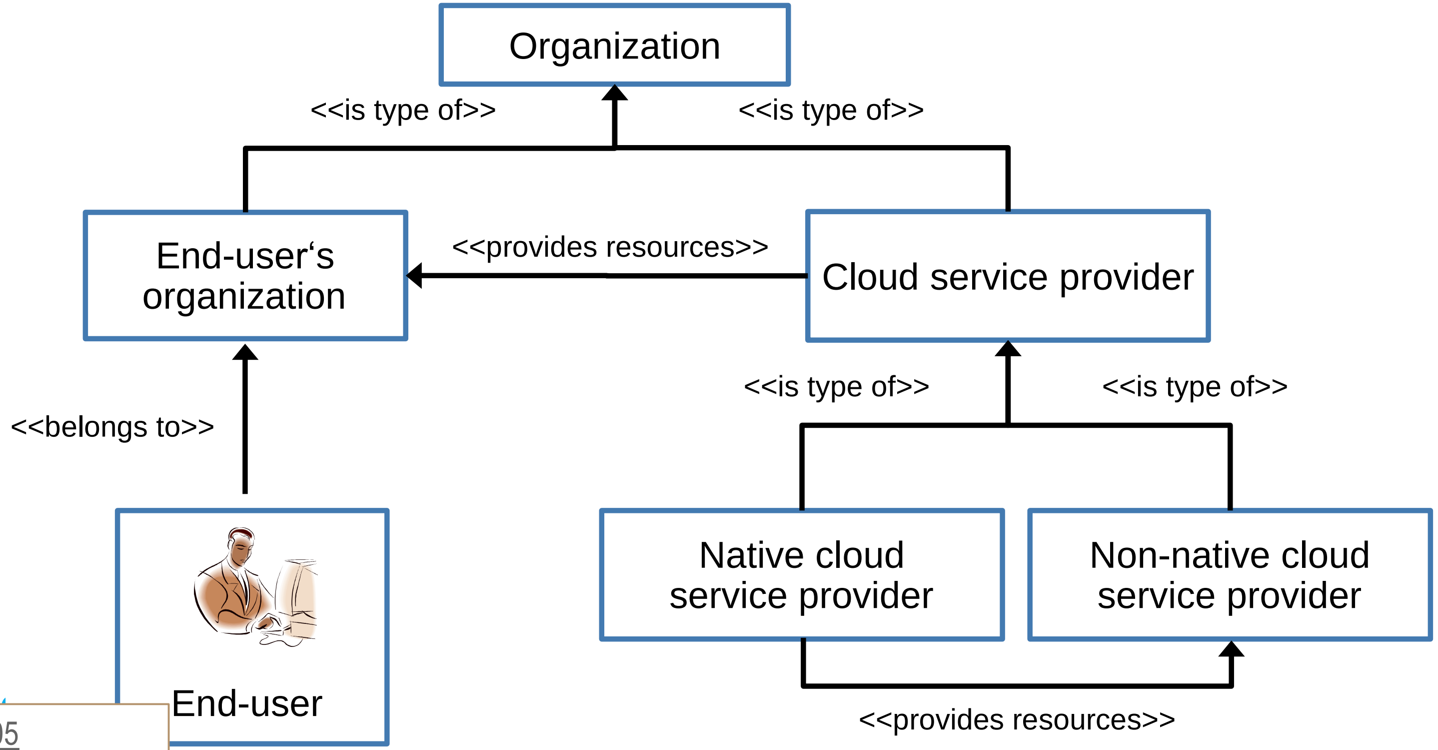
SPEC CUPs: all stakeholders need to communicate using same language

https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.

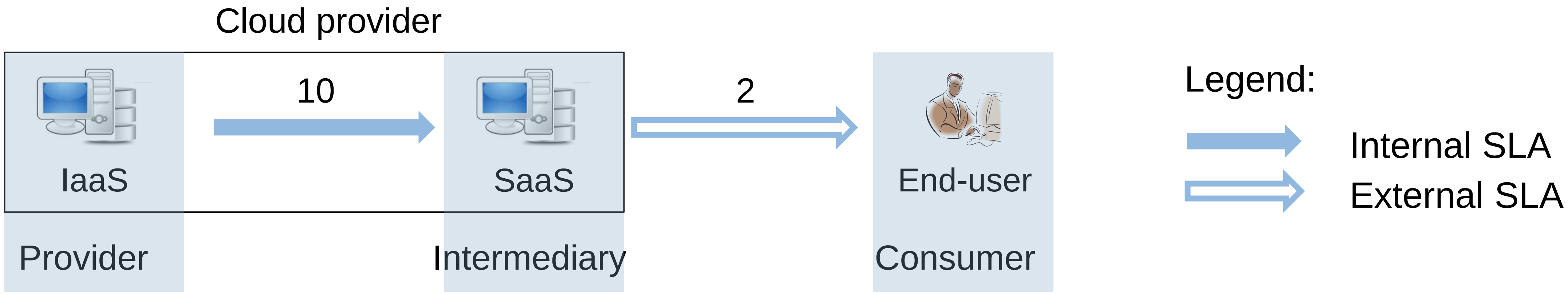
Cloud Usage Patterns: Dimensions

- Abstraction levels
  - Hardware resources ➤ IaaS ➤ PaaS ➤ SaaS
- Stakeholders



Cloud Usage Patterns: Dimensions (cont.)

- Roles: Provider, Intermediary, Consumer
- Server Level Agreements (SLAs)
  - Size/Volume
  - Others (see article)

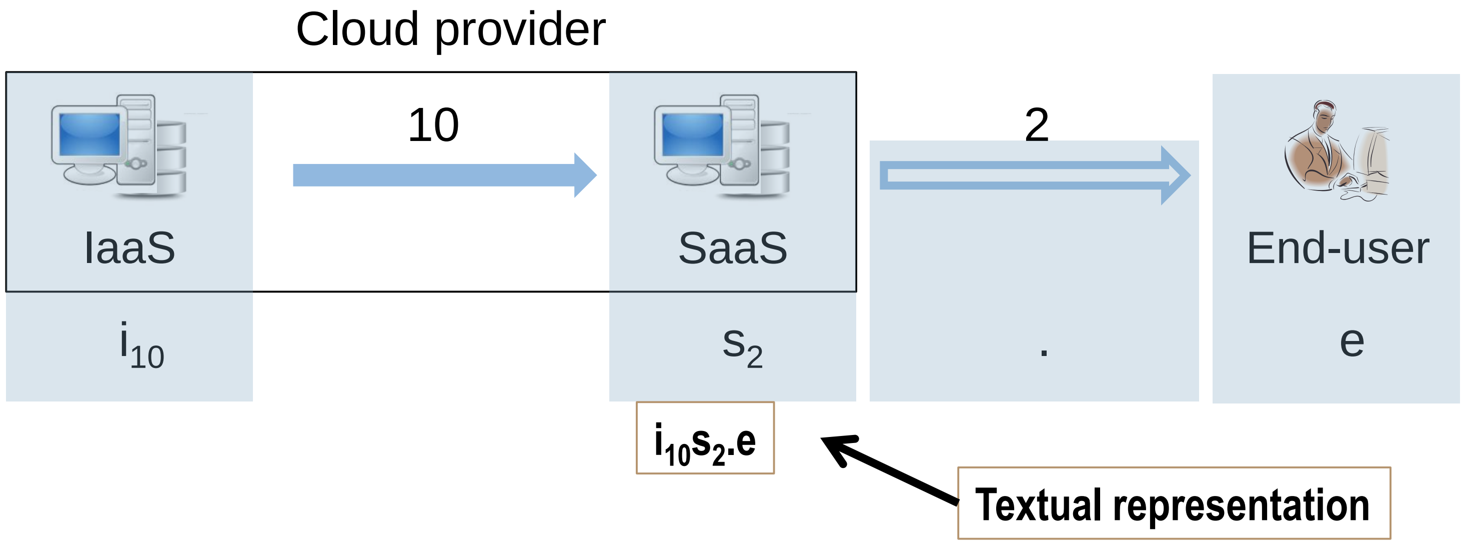


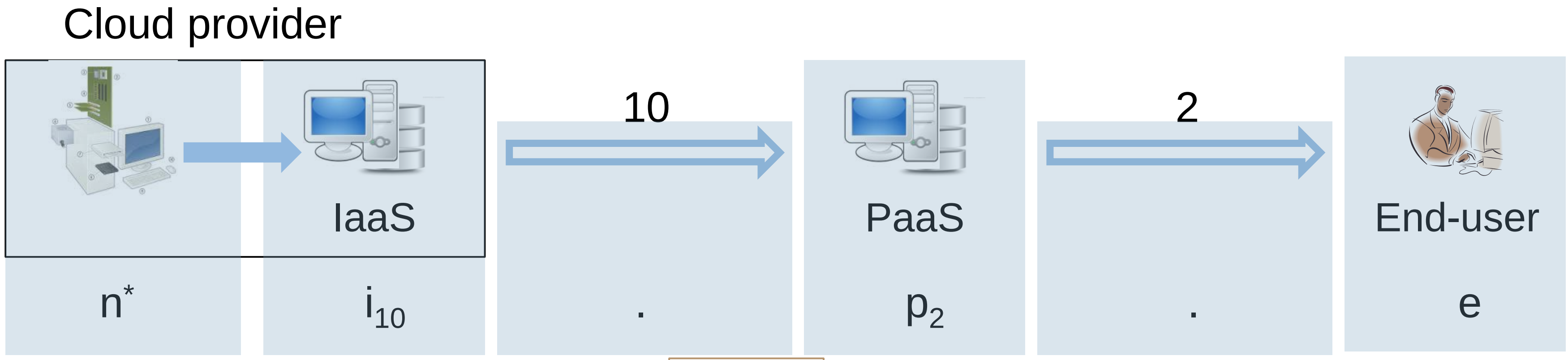
Milenkoski, Iosup, Kounev, Sachs, Mularz, Curtiss, Ding, Rosenberg, and Rygielski. CUP: A Formalism for Expressing Cloud Usage Patterns for Experts and Non-Experts. IEEE Cloud Computing, 2017 (in print)

https://research.spec.org/index.php?id=1105

© 2017 Al…xandru Iosup. All rights reserved.

Cloud Usage Patterns: Value Chains
Textual Representation





* Hardware resources (no virtualization)

https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.

Cloud Usage Patterns in Practice: Value Chains
Textual and Visual Representations

Amazon Web Services

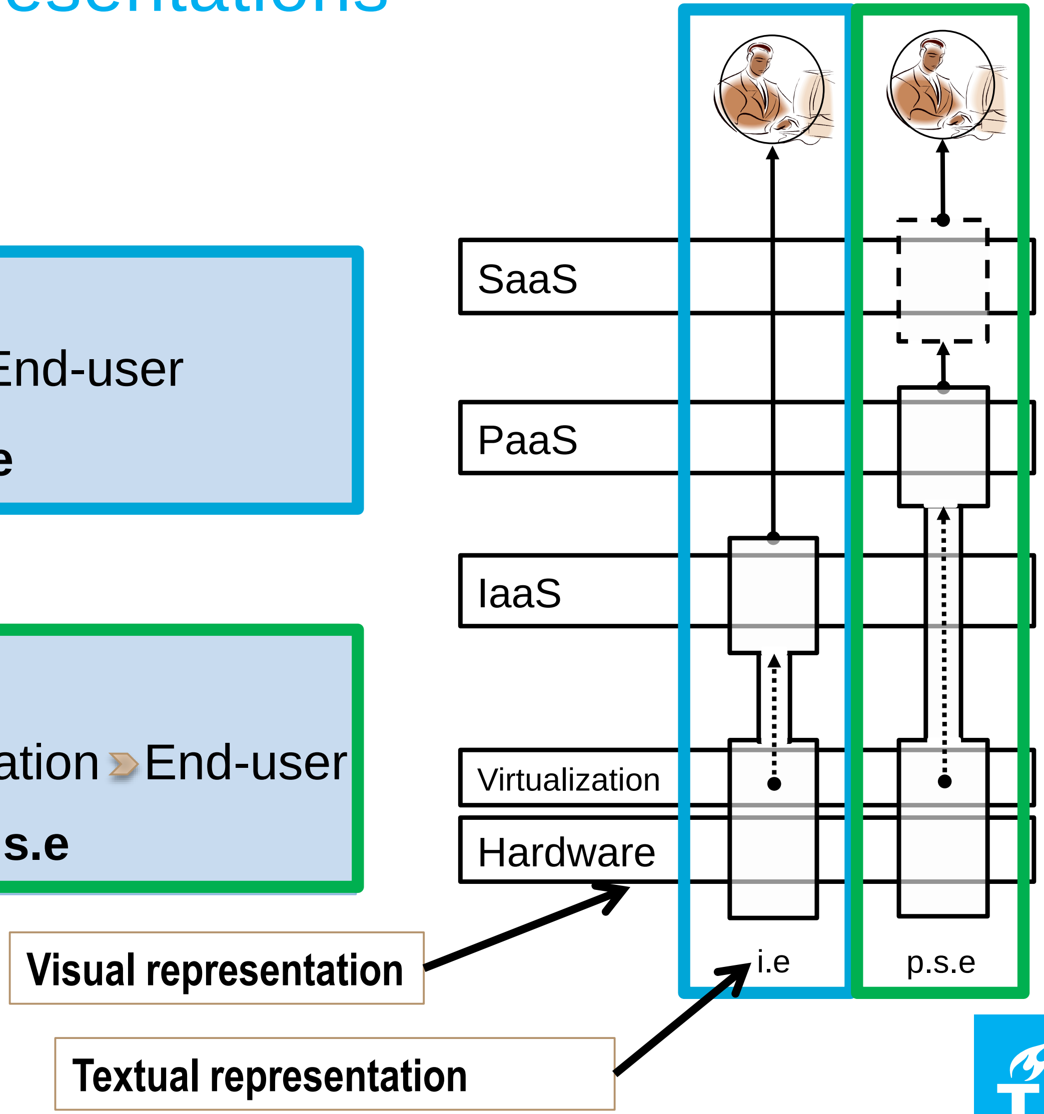
Infrastructure resources ➤ End-user

Textual cloud usage pattern: **i.e**

EZAsset: Asset Management

Google Engine APIs ➤ Application ➤ End-user

Textual cloud usage pattern: **p.s.e**

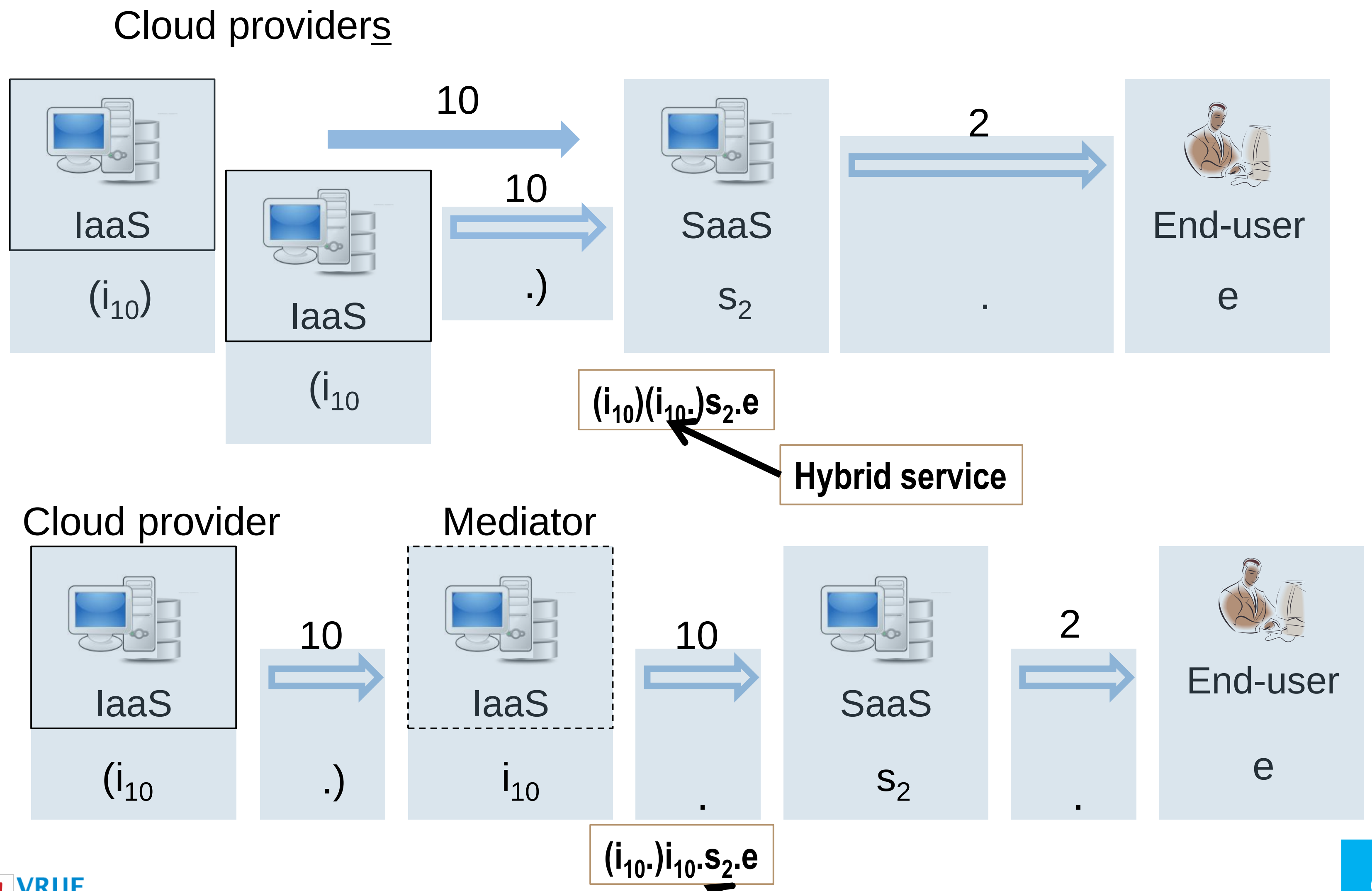


VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Textual Cloud Usage Patterns:
Hybrid Service Provisioning + Value Chains with Mediators



https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.

Cloud Usage Patterns in Practice:
Hybrid Service Provisioning and Value Chain with Mediators

Zynga: Online Gaming services

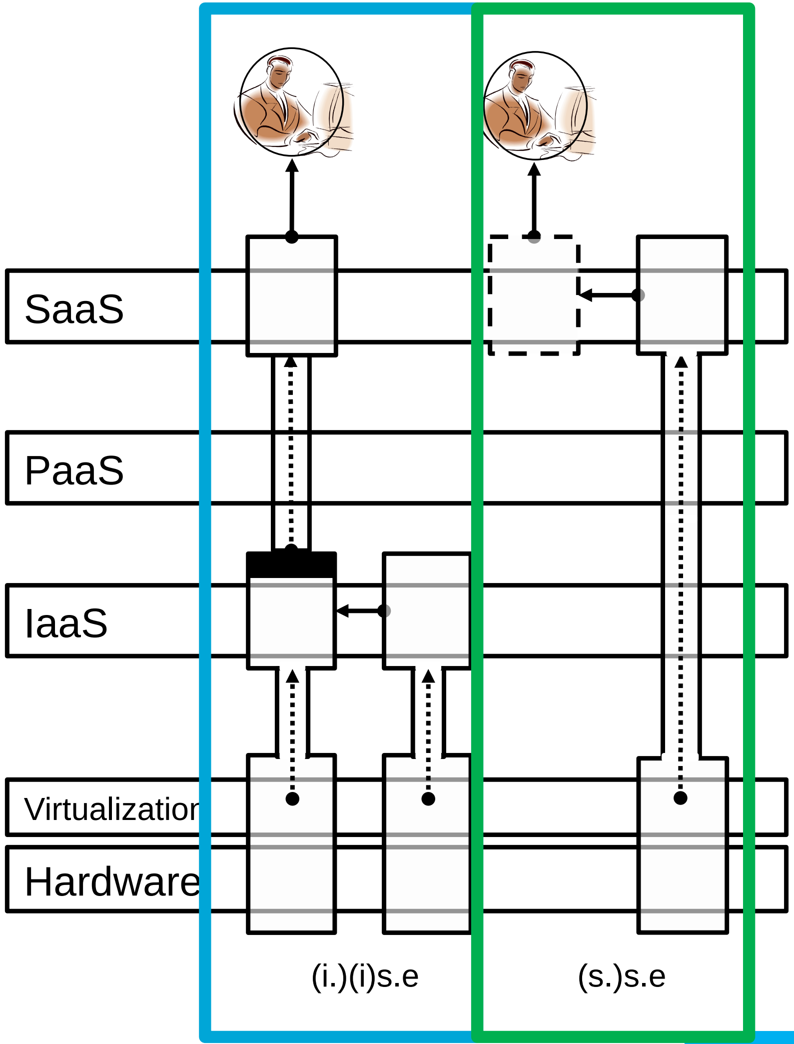
Infrastructure resources (Zynga + Amazon) ➢ Application ➢ End-user

Textual cloud usage pattern: **(i.)(i)s.e**

Dito: Google App reseller

Google Apps software ➢ Reseller ➢ End-user

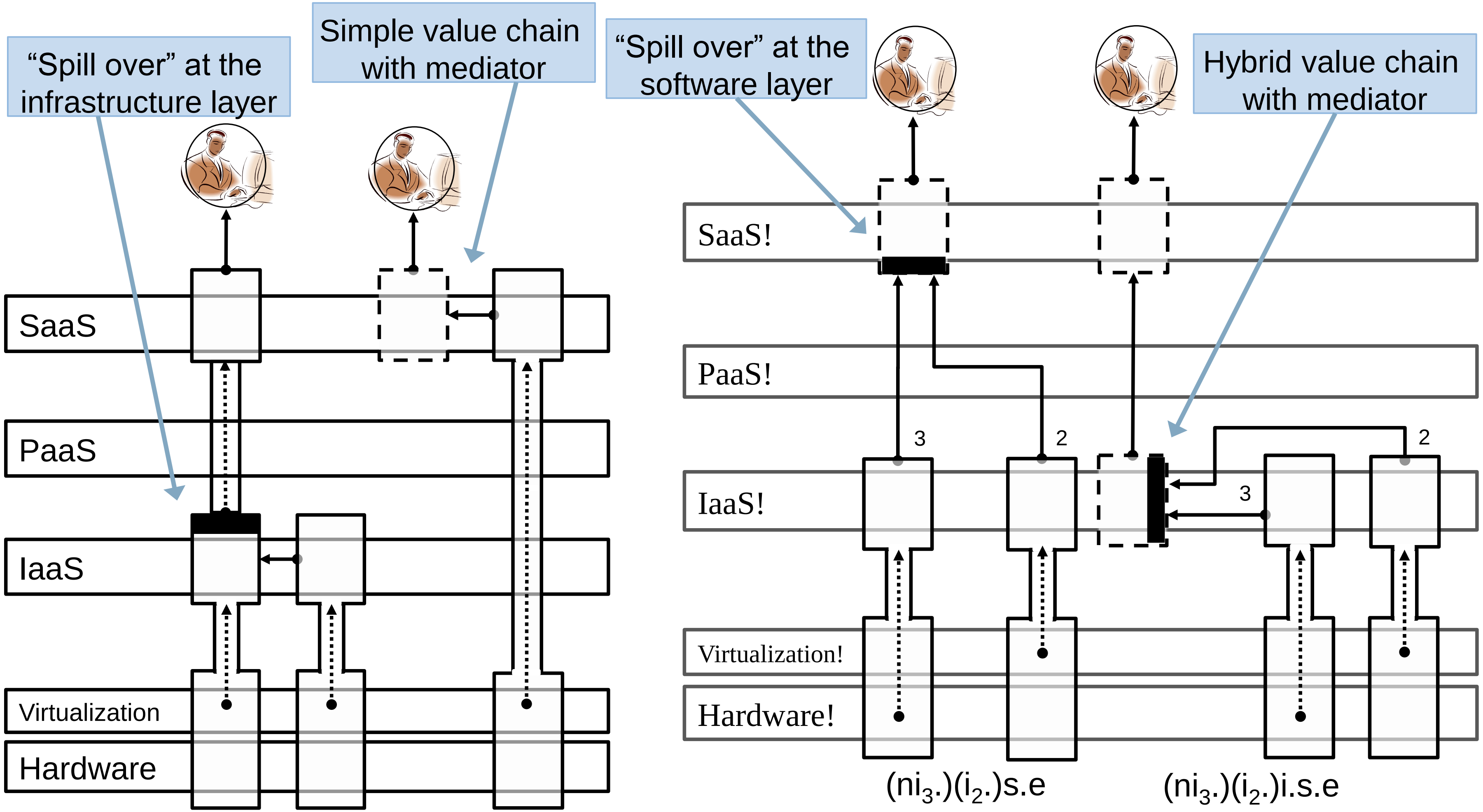
Textual cloud usage pattern: **(s.)s.e**



https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.

Ongoing Development: Deeper CUPs



Milenkoski, Iosup, Kounev, Sachs, Mularz, Curtiss, Ding, Rosenberg, and Rygielski. CUP: A Formalism for Expressing Cloud Usage Patterns for Experts and Non-Experts. IEEE Cloud Computing, 2017 (in print)

https://research.spec.org/index.php?id=1105

xandru Iosup. All rights reserved.

Cloud Usage Patterns in Practice: Cloud Usage Patterns and Real-World Cloud Applications

Facebook

Cloud usage pattern: **nps.e**

“We find within our testing that a realized [non-virtualized] environment brings efficiencies and the ability to scale much more effectively.”

Gio Coglitore, PC World Magazine, IDG News Service, March, 2011 [1]

EasyJet

Cloud usage pattern: **ip.s.e**

“We don't have to build a new high-availability service platform, make firewall configuration changes, or deploy lots of new servers. From the service consumer's point of view, there is no difference in how they get to that service.”

Bert Craven, Microsoft, Case Studies, August, 2011 [2]

Zynga

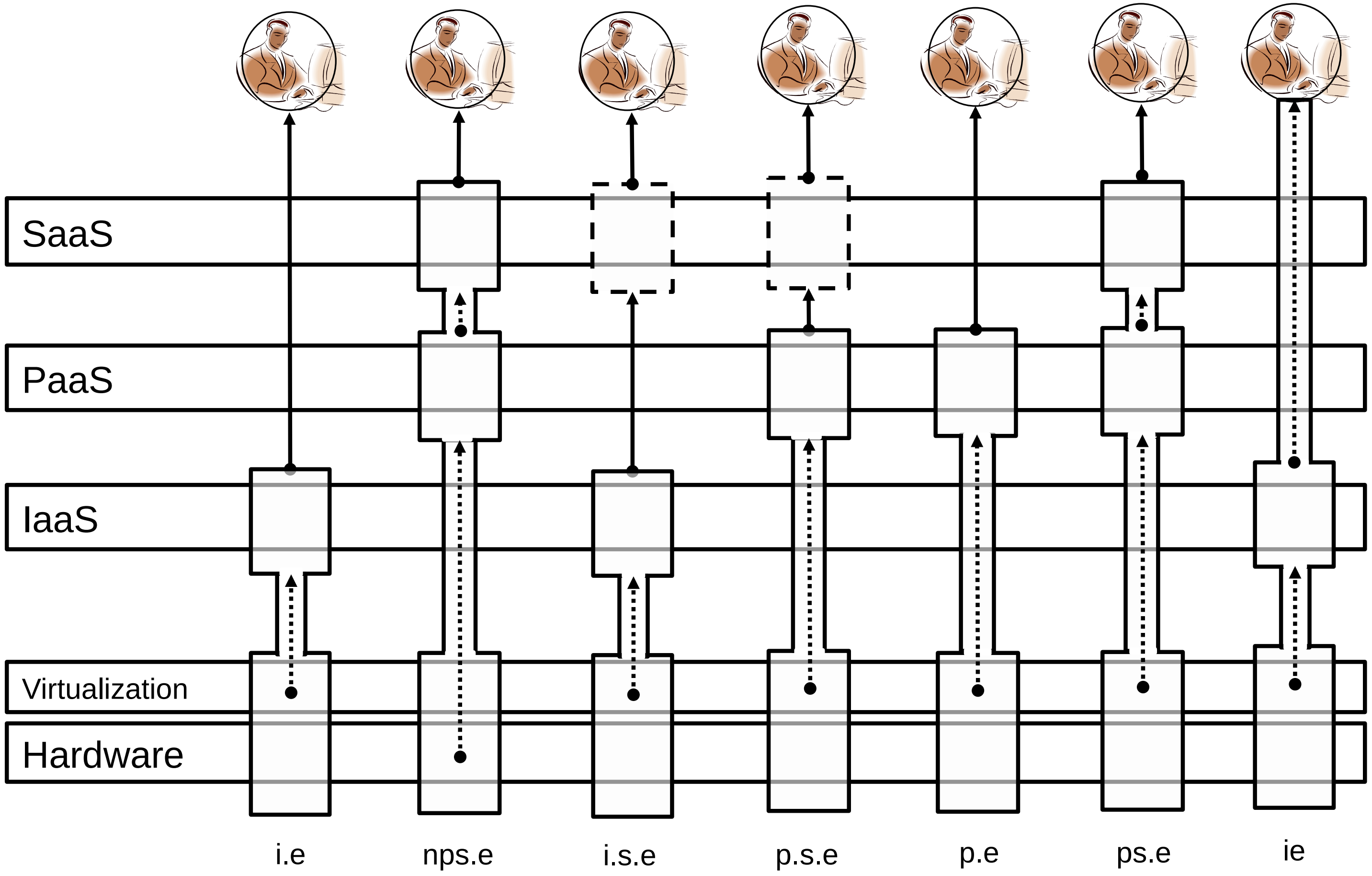
Cloud usage pattern: **(i.)(i)s.e**

“...we came to the realization that we were renting what we could own. The public cloud isn't your own infrastructure; it isn't something you can own and operate in your own way, and it isn't capital equipment, so it was an operating expense.”

Allan Leinward, TechRepublic, Blog Entry, March, 2012 [3]

© 2017 Alexandru Iosup. All rights reserved.

Cloud Usage Patterns:
Diverse Value Chains, Visual + Textual Representations



Milenkoski, Iosup, Kounev, Sachs, Mularz, Curtiss, Ding, Rosenberg, and Rygielski. CUP: A Formalism for Expressing Cloud Usage Patterns for Experts and Non-Experts. IEEE Cloud Computing, 2017 (in print)

https://research.spec.org/index.php?id=1105

© 2017 Alexandru Iosup. All rights reserved.



Laurens Versluis | Erwin van Eyk | Alexandru Iosup

Is there a case for fine-grained, dynamic non-functional requirements for DC workflows?

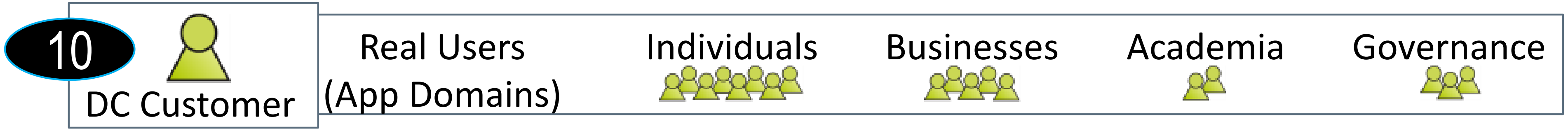
Workflows with Fine-Grained, Dynamic Non-Func'l. Requirements

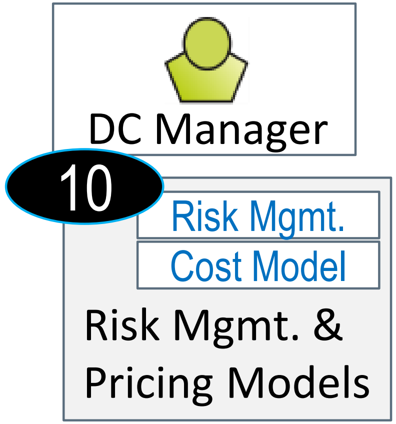
Formalism for Specifying fine-grained, dynamic non-functional requirement for DC workflows

(Jun 2017)

(unpublished, so please do not record or share)

10. Support for DC Customers & Management





- DC Customers
  - Scientific computing, e-Science applications
  - Onling gaming applications
  - Business-critical applications
- DC Management: Risk and Pricing
  - Metrics
  - Tools to assess risk severity
  - Risks: Performance non-compliance, non-absorbed catastrophic failures
- Systems
  - POGGI
  - CAMEO

© 2017 Alexandru Iosup. All rights reserved.


Alexandru Iosup


Siqi Shen


Radu Prodan

DC Support for Online Games

Hosted Cloud-based Architecture, Support for Virtual Worlds, Game Analytics, Content Generation

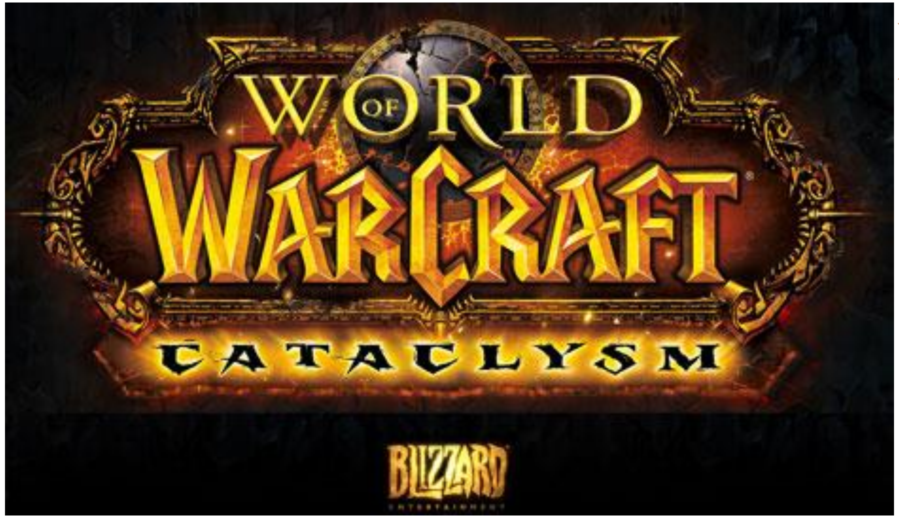
Nae, Iosup, Prodan. Dynamic Resource Provisioning in Massively Multiplayer Online Games. IEEE Trans. Parallel Distrib. Syst. 22(3): 380-395 (2011)

Iosup. POGGI: generating puzzle instances for online games on grid infrastructures. Concurrency and Computation: Practice and Experience 23(2): 158-171 (2011)

Iosup, Lascateu, Tapus. CAMEO: Enabling social networks for Massively Multiplayer Online Games through Continuous Analytics and cloud computing. NETGAMES 2010: 1-6

Iosup, Shen, Guo, Hugtenburg, Donkervliet, Prodan. Massivizing online games using cloud computing: A vision. ICME Workshops 2014: 1-4

World of Warcraft, a Traditional HPC User



- 10 data centers
- 13,250 server blades, 75,000+ cores
- 1.3PB storage
- 68 sysadmins (1/1,000 cores)

http://www.datacenterknowledge.com/archives/2009/11/25/wows-back-end-10-data-centers-75000-cores/

VU VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Online games hosting model

- Generic Online Games (non-MM)
  - **Static:** dedicated isolated single servers

- MMOGs
  - **Static:** dedicated clusters - using parallelization techniques

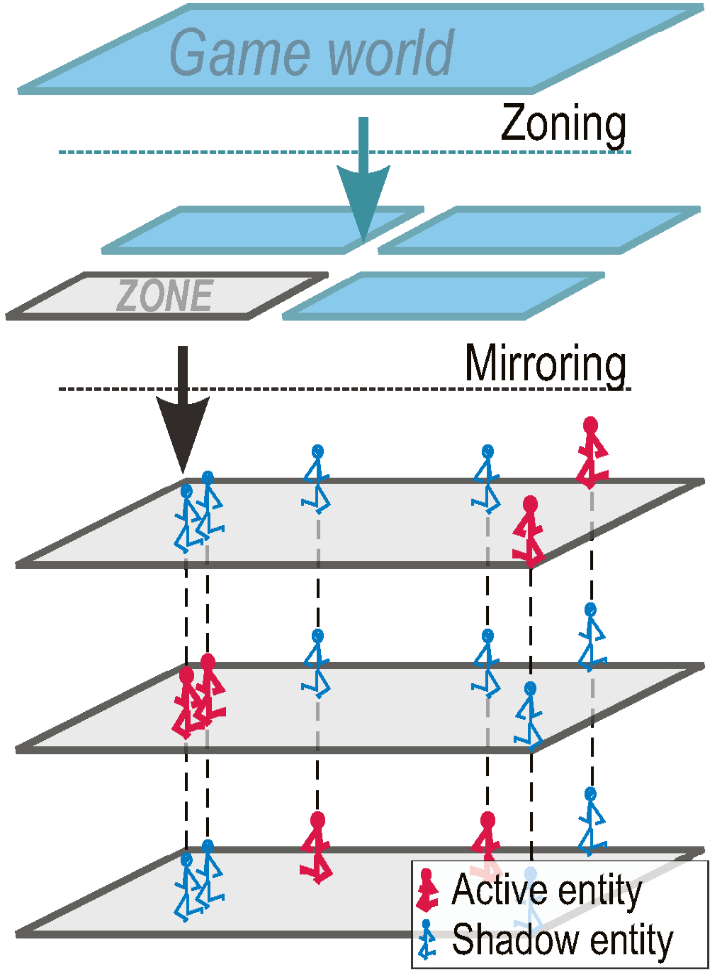
- Problems with these approaches
  1. Large amount of over-provisioning
  2. Non-efficient coverage of the world for the provided service

[Source: Nae, Iosup, and Prodan, ACM SC 2008]

…ts reserved.

Game parallelization models

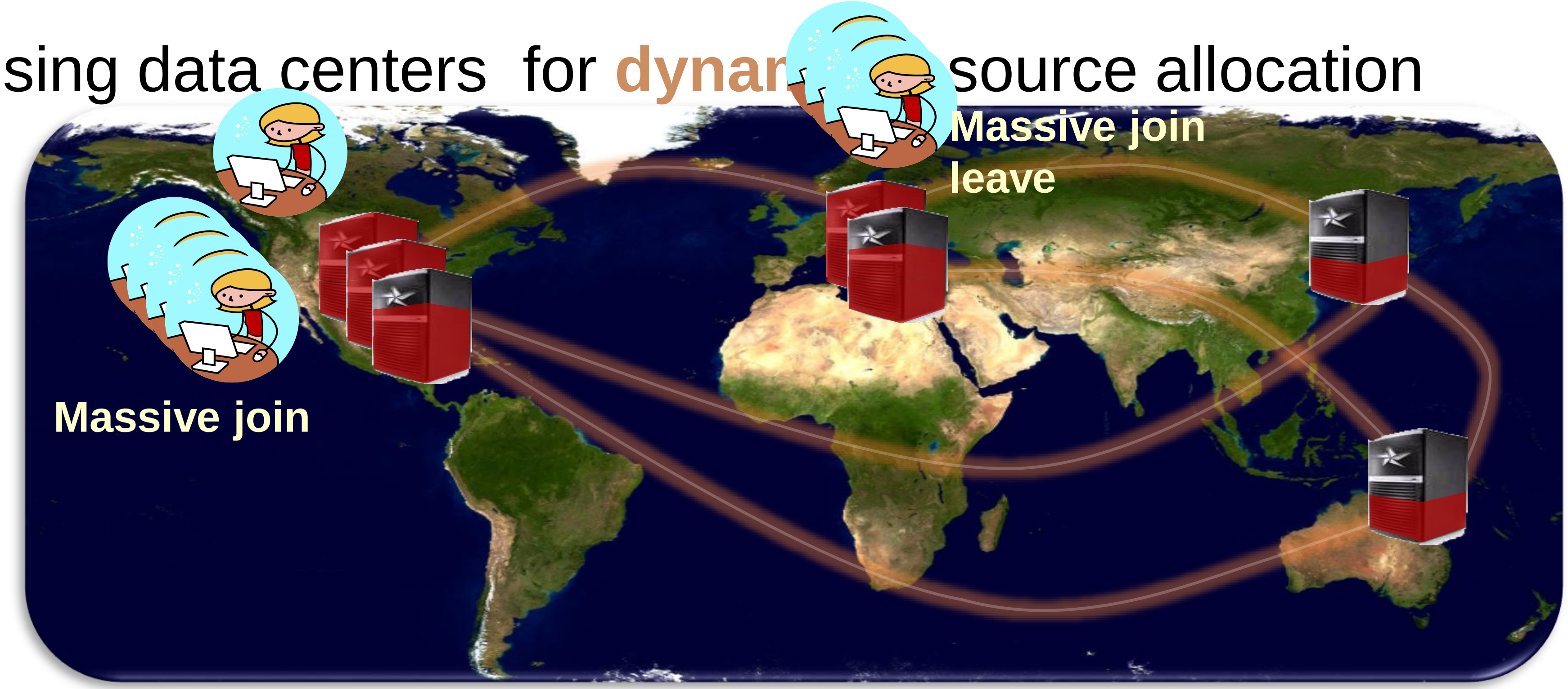
- Models:
  - Zoning: huge game-world division into geographical sub-zones – each zone is handled by different machines
  - Mirroring: the same game-world handled by different machines, each one handling a subset of the contained entities (synchronized states)
  - **Instancing/sharding**: multiple instances of the same zone with independent states. (World of Warcraft, Runescape,..)



© 2017 Alexandru Iosup. All rights reserved.

Proposed hosting model: dynamic

- Using data centers for **dynan** source allocation



- Main advantages:
  1. Significantly lower over-provisioning
  2. Efficient coverage of the world is possible

[Source: Nae, Iosup, and Prodan, ACM SC 2008]

ıts reserved.

Experimental Setup [1/3]
Discrete-Event Simulator

- Input
  - Traces from RuneScape, a real top-5 MMOG
    - 7 countries, 3 continents
    - More than 130 game worlds
  - Consisting of
    - Geographical location
    - Number of clients
    - Over 10,000 samples at 2 min. interval, 2 weeks

- Output (for every time-step)
  - Resource allocation decisions
  - Resource allocation
  - Performance metrics

[Source: Nae, Iosup, and Prodan, ACM SC 2008]

ıts reserved.

Experimental Setup [2/3]
Environment

- 1 game operator
- 17 data centers
- 11 data center time-space renting policies

Location		Data Centers	Machines (total)
Continent	Country		
Europe	Finland	2	8 machines
	Sweden	2	8 machines
	U.K.	2	20 machines
	Netherlands	2	15 machines
North America	U.S. (West)	2	35 machines
	Canada (West)	1	15 machines
	U.S. (Central)	1	15 machines
	U.S. (East)	2	32 machines
	Canada (East)	1	10 machines
Australia	Australia	2	8 machines

VU VRIJE UNIVERSITEIT AMSTERDAM

[Source: Nae, Iosup, and Prodan, ACM SC 2008]

its reserved.

TU Delft

Experimental Setup [3/3]
Performance Metrics

- Resource over-provisioning [%]
  - The wasted resources vs. optimal provisioning at each simulation time step for all utilized machines (cumulative)
- Resource under-provisioning [%]
  - The amount of resources needed but not allocated, for all machines (computed individually)
- Significant under-provisioning events (count)
  - Count of events: resource under-provisioning is >1%, for a period of 2 minutes → **people leave**

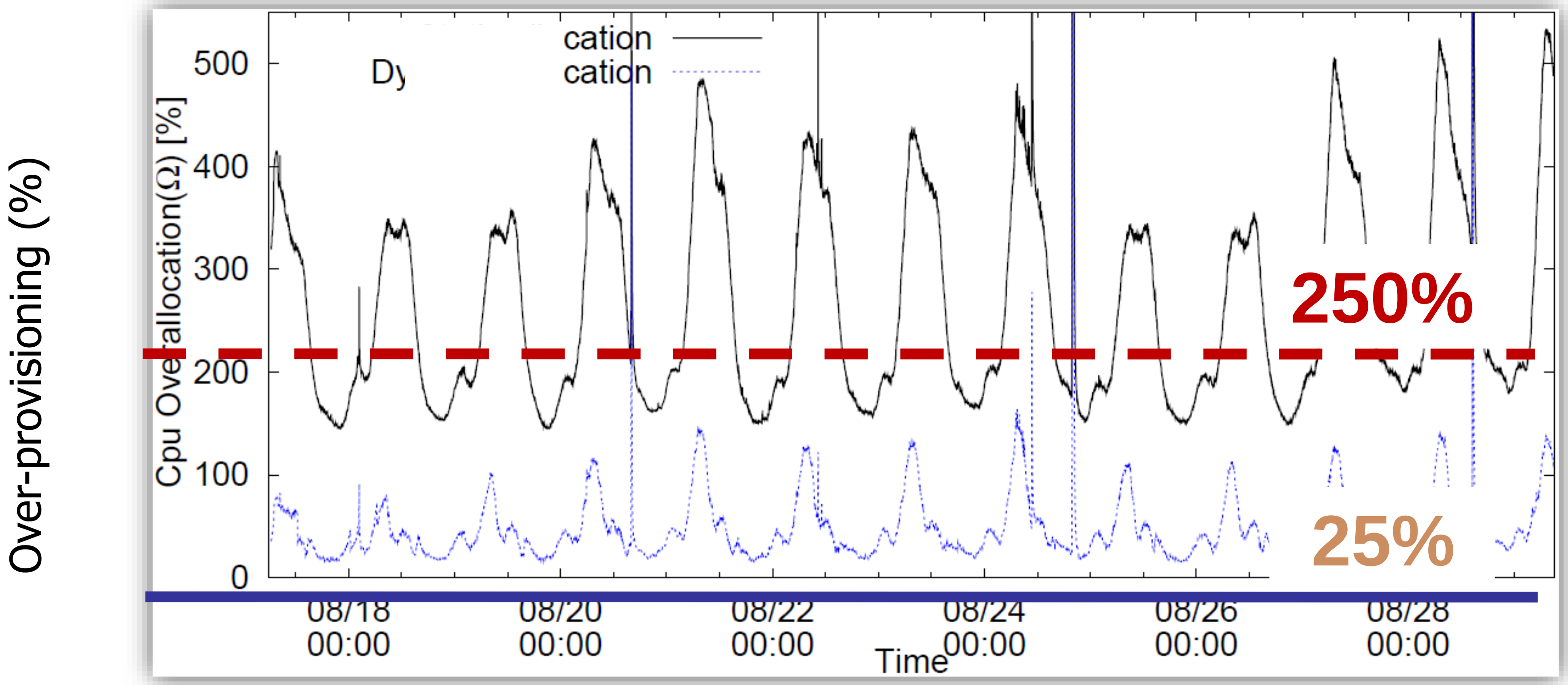
VU VRIJE UNIVERSITEIT AMSTERDAM

[Source: Nae, Iosup, and Prodan, ACM SC 2008]

ıts reserved.

TU Delft

Resource Provisioning and Allocation
Static vs. Dynamic Provisioning

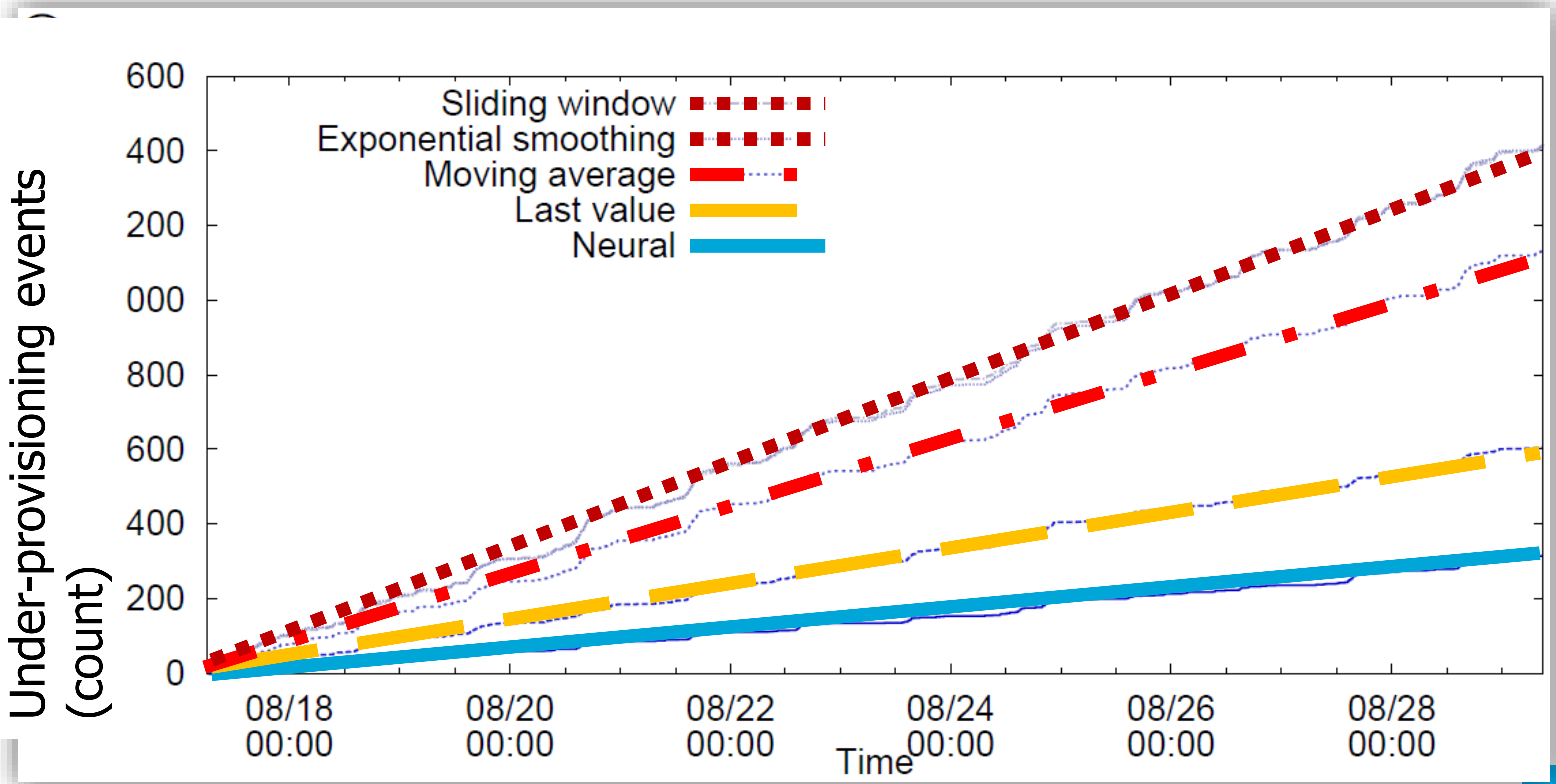


[Source: Nae, Iosup, and Prodan, ACM SC 2008]

ıts reserved.

Impact of Load Prediction Accuracy

Q: How does the prediction accuracy impact resource provisioning? A: Good prediction matters.



VU VRIJE UNIVERSITEIT AMSTERDAM

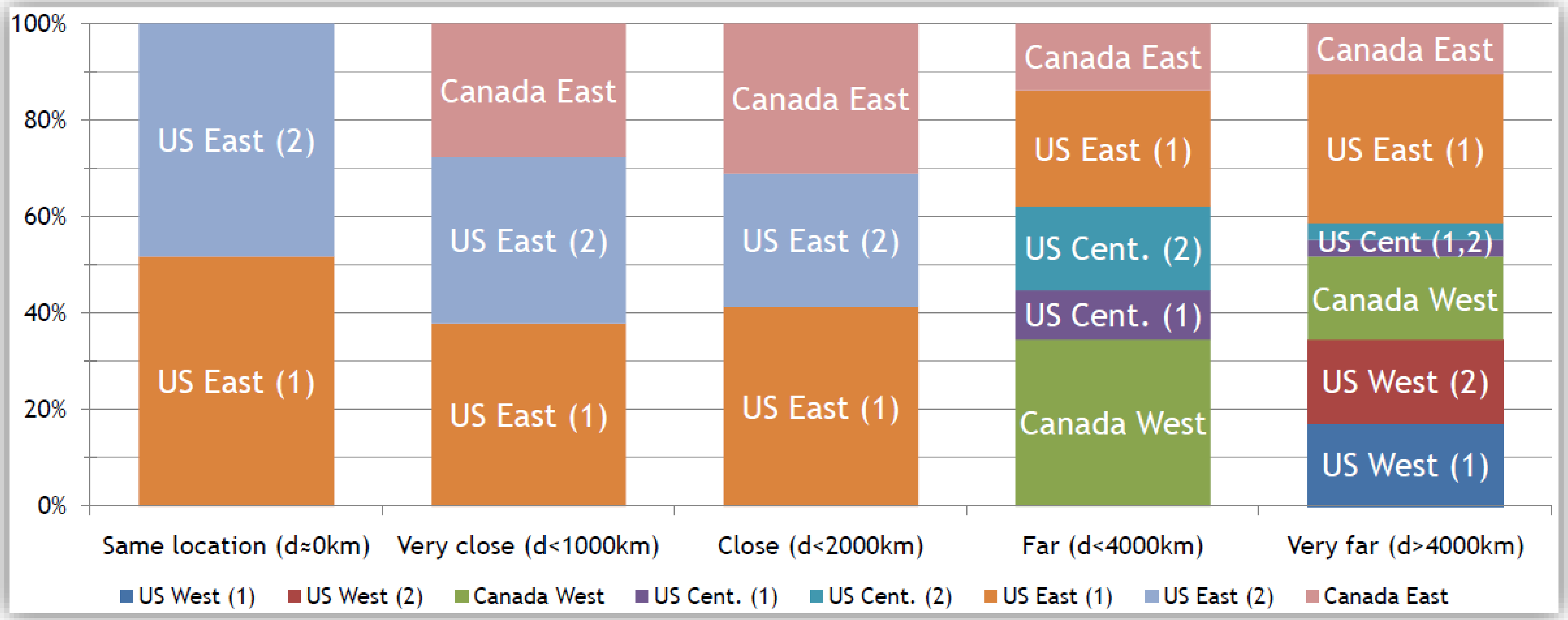
[Source: Nae, Iosup, and Prodan, ACM SC 2008]

its reserved.

TU Delft

Latency Tolerance: From None to High

Q: What is the impact of latency tolerance on hosting?



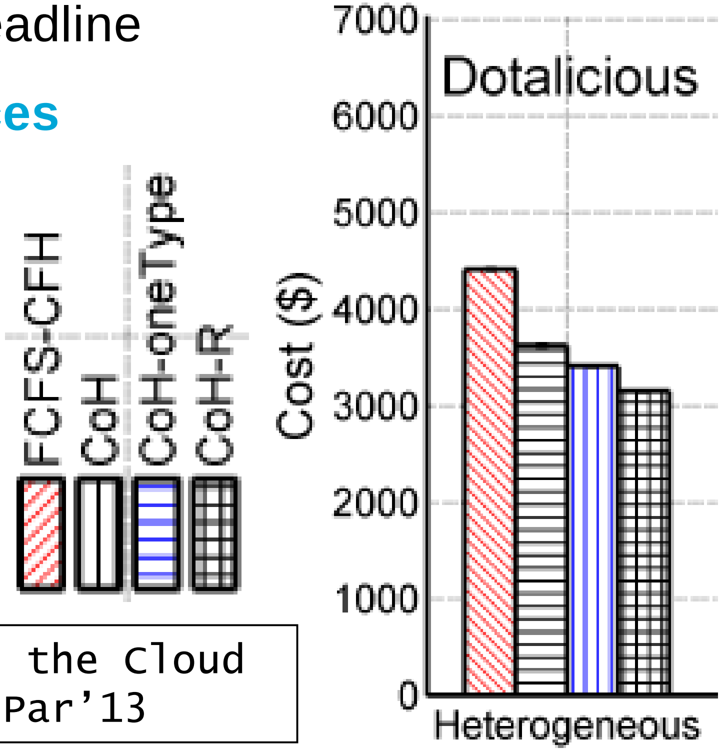
A:	(left)	(mid)	(right)
	very sensitive	sensitive	non-sensitive
	very costly	costly	cheap

VU

Portfolio Scheduling for Online Gaming
(also for Scientific Workloads)

- **CoH =** Cloud-based, online, Hybrid scheduling
  - Intuition: keep rental cost low by finding good mix of machine configurations and billing options
  - Main idea: **portfolio scheduler** = run *both* solver of an Integer Programming Problem and various heuristics, then pick best schedule at deadline
  - Additional feature: Can use **reserved cloud instances**
- Promising early results, for **Gaming** (and scientific) workloads

Trace	#jobs	average runtime [s]
Grid5000	200,450	2728
LCG	188,041	8971
DotaLicious	109,251	2231



Shen, Deng, Iosup, Epema, Scheduling Jobs in the Cloud Using On-demand and Reserved Instances, EuroPar'13

Also Studied

- Via real game measurements
  - Interactivity model (short-term msmt.)
  - Effects of underperforming platform (long-term msmt.)
- Via prototype implementation
  - Match model-reality [TPDS’11]
- Via simulation
  - Impact of virtualization [NetGames’11][IJAMC’11] and un-availability [EuroPar WS’14]
  - Economic and pricing models [ICPE’11] [CAC’13] [MMSys’14]

VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

Alexandru
Iosup

CAMEO

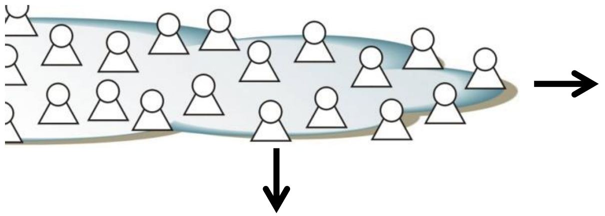
Continuous Analytics and cloud computing to enable social networks for MMOGs

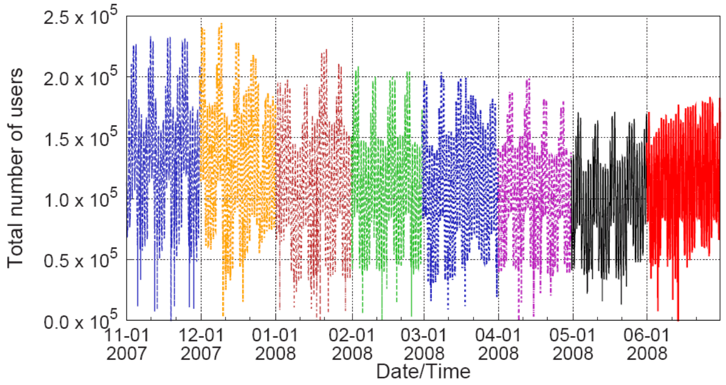
Iosup, Lascateu, Tapus. CAMEO: Enabling social networks for Massively Multiplayer Online Games through Continuous Analytics and cloud computing. NETGAMES 2010: 1-6

Continuous Analytics for MMOGs

Analyzing the behavior of millions of players, on-time

- **Data mining**, data access rights, cost v. accuracy, …
- Reduce upfront costs
- Low response time & Scalable
- Large-scale Graph Processing

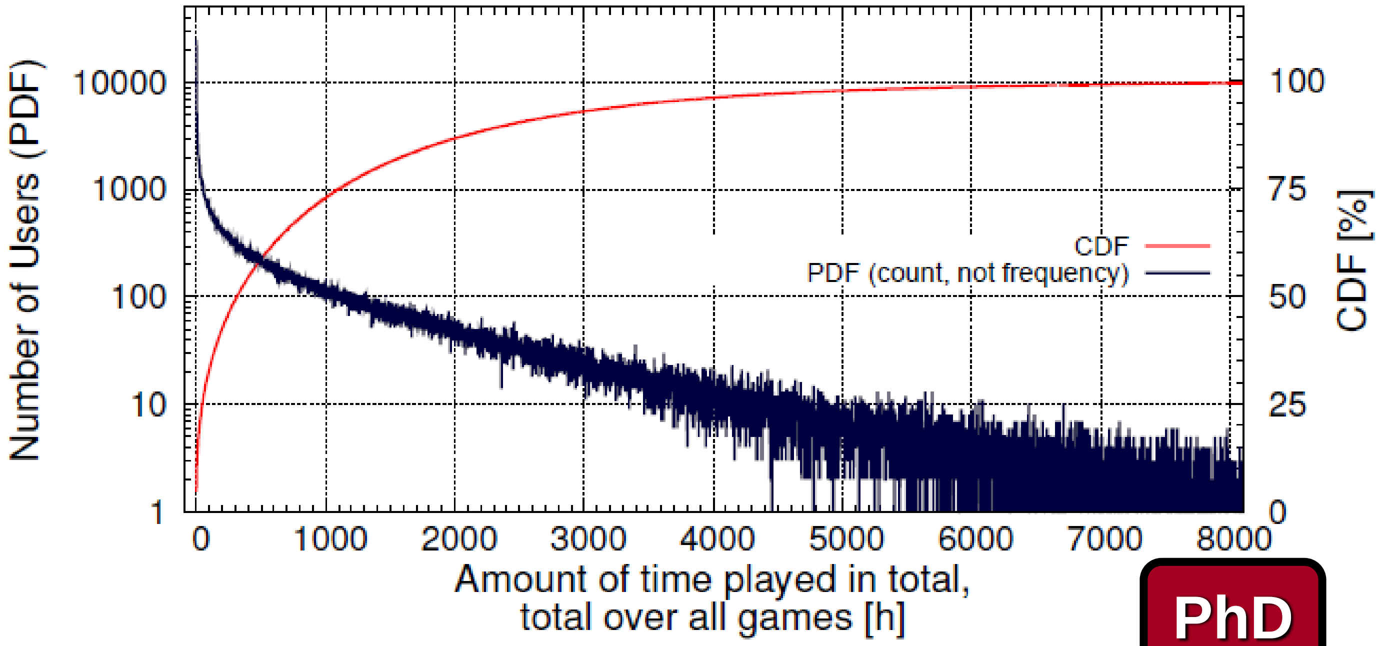




Analysis of Meta-Gaming Network

"When you play a number of games, not as ends unto themselves but as parts of a larger game, you are participating in a metagame." (Dr. Richard Garfield, 2000)

XFire: since 2008, 3+ years, covered 500K/20M players (2.5%)



PhD

S. Shen, and A. Iosup, The XFire Online Meta-Gaming Network: Observation and High-Level Analysis, MMVE 2011

…rved.

>skip to results
>skip all

DotA communities

- Players are loosely organised in communities
  - Operate game servers
  - Maintain lists of tournaments and results
  - Publish statistics and rankings on websites

- Dota-League: players join a queue and matchmaking forms teams
- DotAlicious: players can choose which match/team to join

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

erved.

Our Datasets

- We have crawled all matches played and per match have:
  - Names of the players for each team
  - Active, start and end time
  - Game-play statistics per team
  - The team that won the match
- Dota-League:
  - ~1.5M matches played between Nov’08 and Jul’11, 61K players
- DotAlicious:
  - ~0.6M matches played between Apr’10 to Feb’12, 62K players

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

erved.

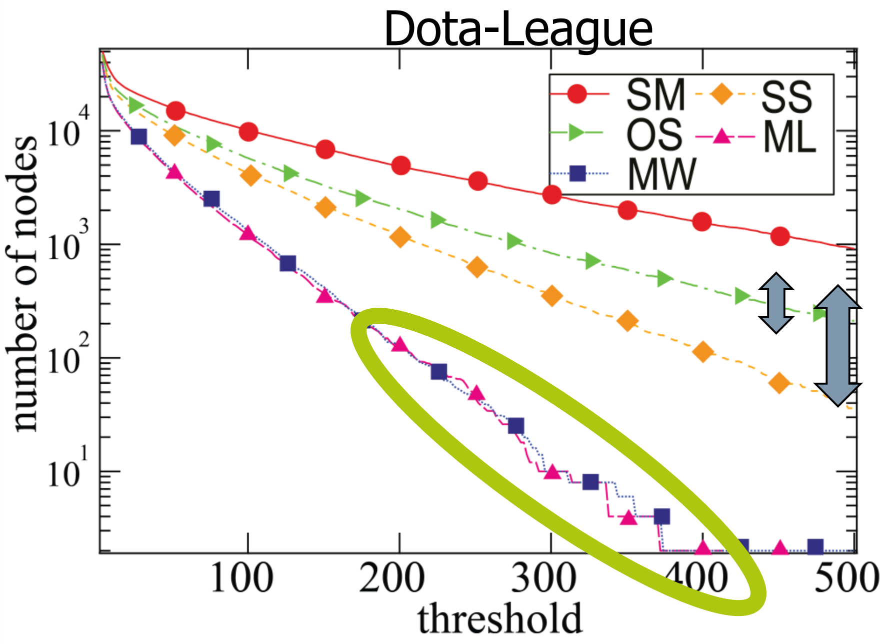
From game instances to social ties

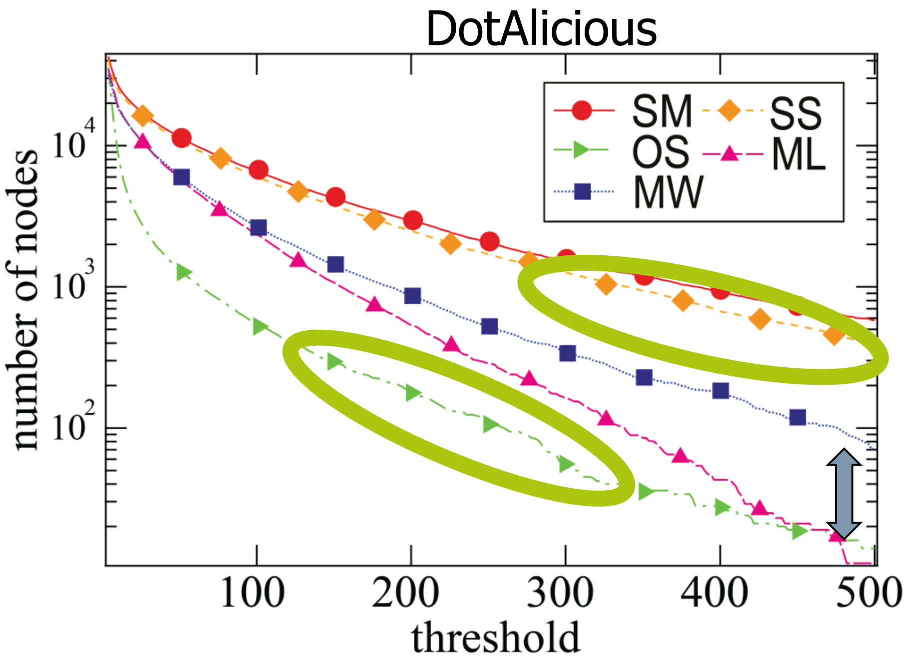
- We need to define how to map the relationships found in real-world matches to a **gaming graph** (nodes and links)
- We use six different mappings and various thresholds:
  - **SM:** two players occur more than *n* times in the **same match**
  - **SS:** two players occur more than *n* times on the **same side**
  - **OS:** two players occur more than *n* times on **opposing sides**
  - **ML:** two players have **lost** more than *n* **matches together**
  - **MW:** two players have **won** more than *n* **matches together**
  - **PP:** a directed version of the mappings above. A link exists if a player has played more than n percent of his matches together

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

erved.

Network sizes (w/o isolated nodes) in the Gaming Graph



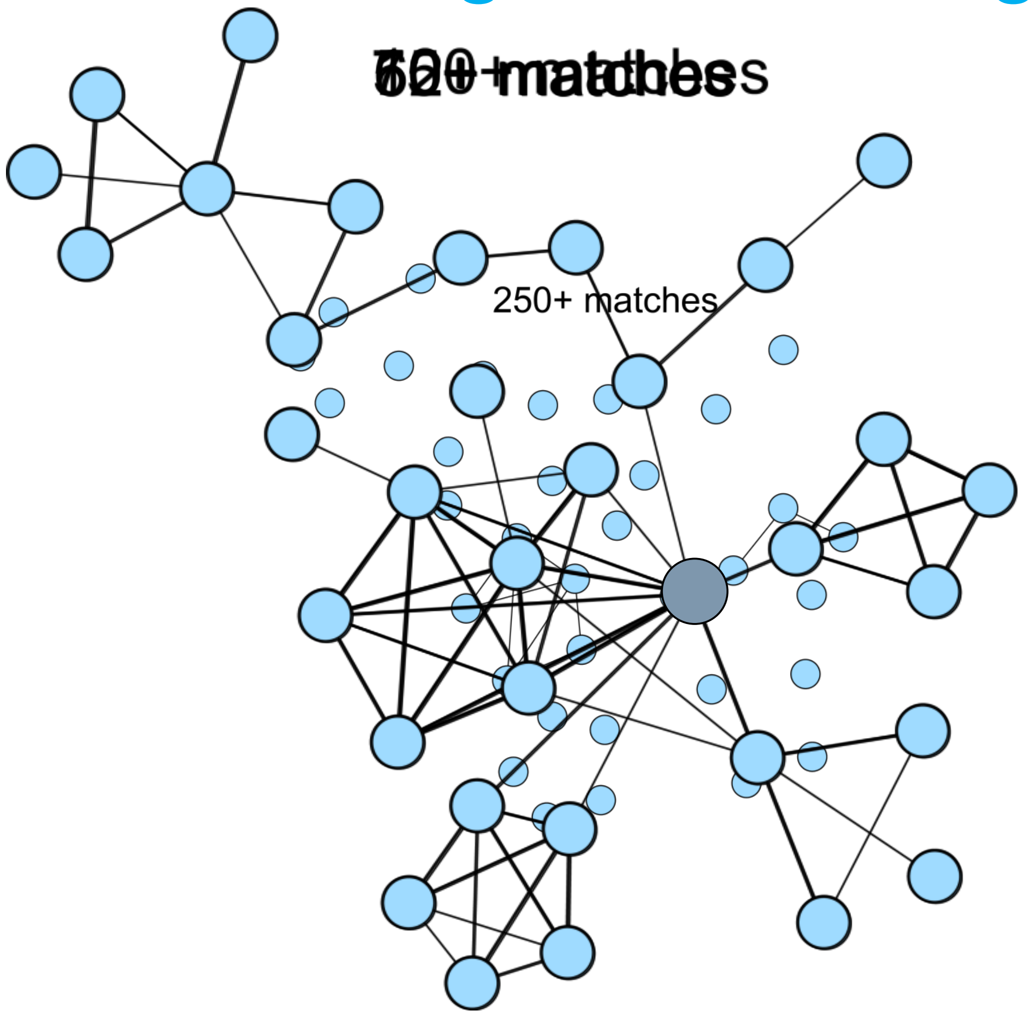


Number of nodes in the network as a function of the threshold

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

erved.

Small clusters show strong ties in the gaming graph



R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

VU

erved.

TU Delft

Relationships in the gaming graph

- Players who regularly play together in DotAlicious do so in more diverse combinations than in Dota-League
- Contrary to Dota-League, DotAlicious players tend to play on the same side: playing together intensifies the social bond
- Winning together increases friendship relationships, while loosing together weakens friendship relationships
- Small clusters of friends with very strong social ties exist

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

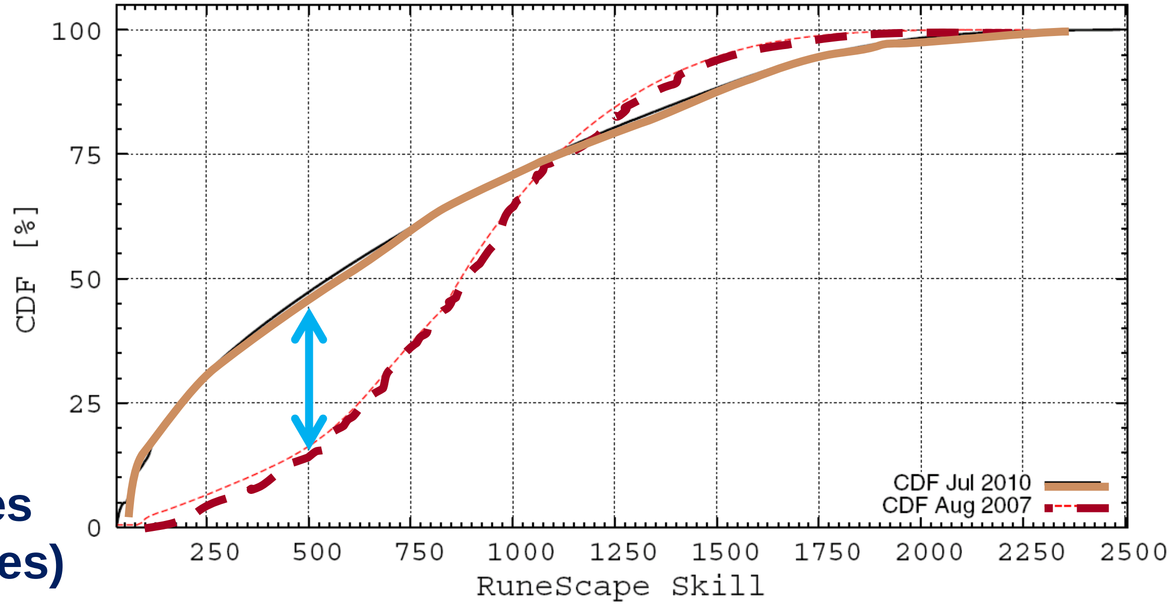
erved.

@large: Sample Analytics Results

Skill Level Distribution in RuneScape

- Runescape: 135M active accounts, 7M active (2008)
- High-scoring players: 1.8M (2007) / 3.5M (2010)
- **Largest MMOG msmt.**
- **Player skill: distribution changes over time**

Need dynamic (procedural) content generation for games (using hosted cloud machines)



A. Iosup, A. Lascateu, N. Tapus, CAMEO: Enabling Social Networks for Massively Multiplayer Online Games through Continuous Analytics and Cloud Computing, ACM NetGames 2010.

erved.

Alexandru Iosup

POGGI

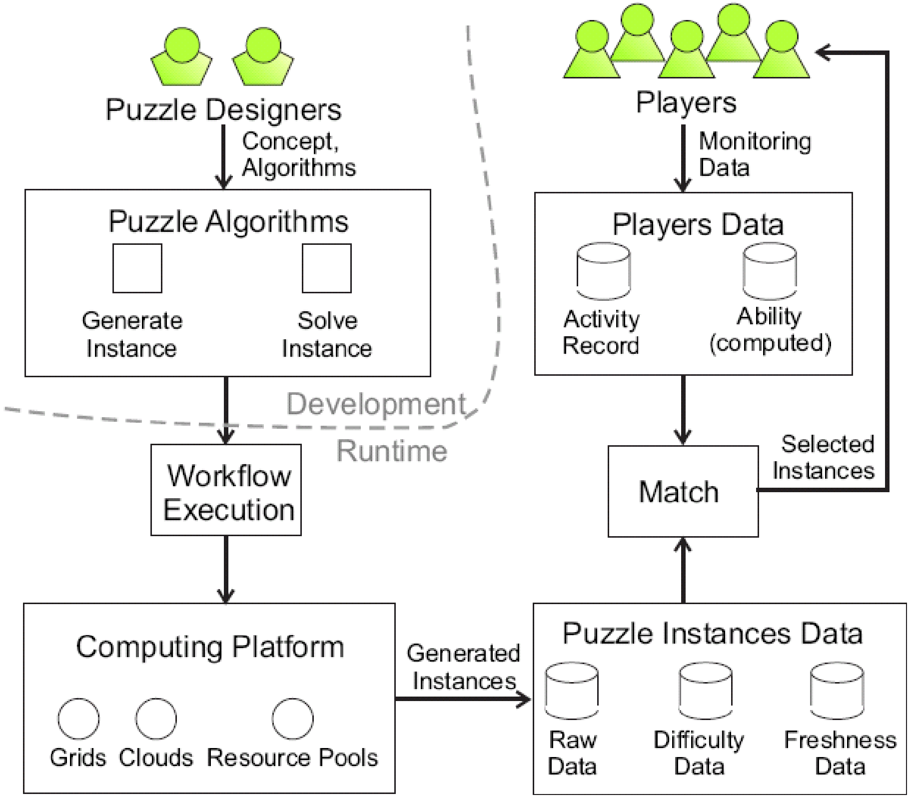
Continuous Analytics and cloud computing to enable social networks for MMOGs

Iosup. POGGI: generating puzzle instances for online games on grid infrastructures. Concurrency and Computation: Practice and Experience 23(2): 158-171 (2011)

Iosup. POGGI: Puzzle-Based Online Games on Grid Infrastructures. Euro-Par 2009: 390-403. Distinguished Paper Award.

The POGGI Content Generation Framework

Only the puzzle concept, and the instance generation and solving algorithms, are produced at development time



Hosted cloud system to generate instances on-demand, reliably, efficiently, and with performance guarantees

* A. Iosup, POGGI: Puzzle-Based Online Games on Grid Infrastructures, EuroPar 2009 (Best Paper Award)

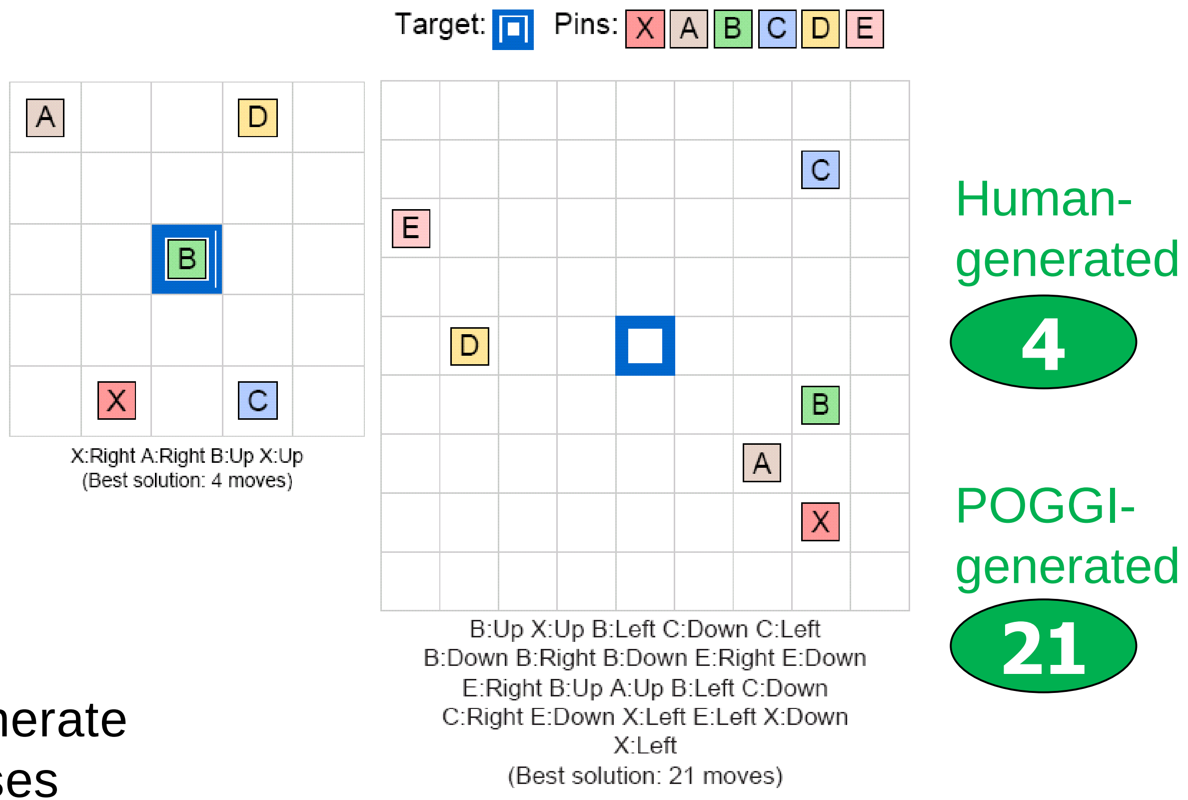
Puzzle-Specific Considerations
Generating Player-Customized Content

Puzzle difficulty

- Solution size (moves to solve)
- Solution alternatives
- Variation of moves
- Skill moves

Player ability

- Keep population statistics and generate enough content for most likely cases
- Match player ability with puzzle difficulty yet take into account puzzle freshness



VRIJE UNIVERSITEIT AMSTERDAM

© 2017 Alexandru Iosup. All rights reserved.

TU Delft

60’

Massivizing Computer Systems
A Proposal for Collaboration, with Topics

~2’ —— About the Massivizing Computer Systems Group

5’ —— The Golden Age of Large-Scale Computer Systems

5’ —— Yet We Are in Crisis

- The main challenges
- How we address them

~40’ —— Our Vision and Topics

10’ —— Take-Home Message

Consider Reading the Following:

1. Iosup et al. LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms. PVLDB 9(13): 1317-1328 (2016)
2. Guo et al.: Design and Experimental Evaluation of Distributed Heterogeneous Graph-Processing Systems. CCGrid 2016: 203-212
3. van Beek et al.: Self-Expressive Management of Business-Critical Workloads in Virtualized Datacenters. IEEE Computer 48(7): 46-54 (2015)
4. Jia et al.: Socializing by Gaming: Revealing Social Relationships in Multiplayer Online Games. TKDD 10(2): 11 (2015)
5. Ghit et al.: V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows. CCGRID 2014: 927-932
6. Guo et al.: How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis. IPDPS 2014: 395-404
7. Javadi et al.: The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. J. Parallel Distrib. Comput. 73(8): 1208-1223 (2013)
8. Iosup and Epema: Grid Computing Workloads. IEEE Internet Computing 15(2): 19-26 (2011)
9. Iosup et al.: On the Performance Variability of Production Cloud Services. CCGRID 2011: 104-113
10. Iosup et al.: Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. IEEE Trans. Parallel Distrib. Syst. 22(6): 931-945 (2011)