(1) Non-trivial Cloud Computing Phenomena: The Impact of Performance Variability on Big Data

(2) Exploring Computing Infrastructure Convergence: HPC and Big Data Graph Processing on Multicores

Alexandru Uta

a.uta@vu.nl

Vrije Universiteit Amsterdam

Data-intensive Scientific Discovery

Science paradigms

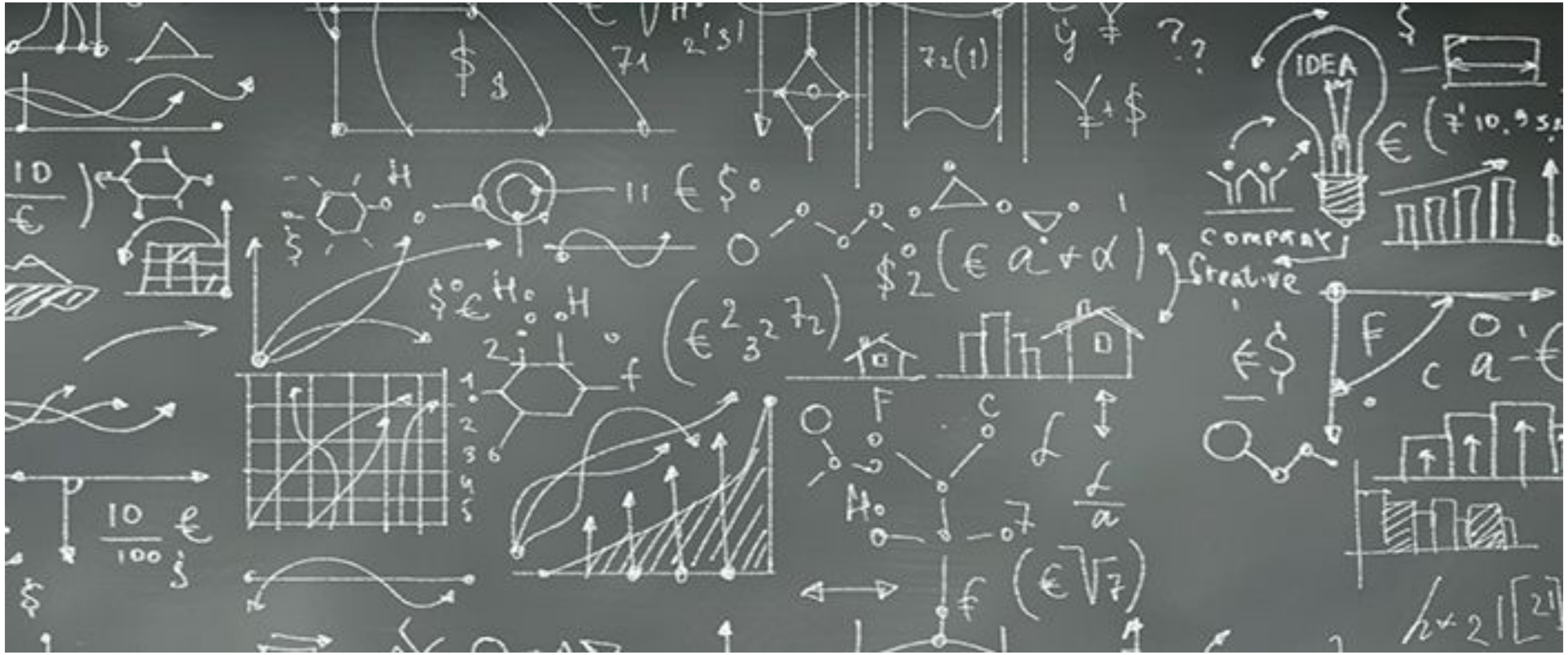
- Thousand years ago: **empirical/experimental** science

The Fourth Paradigm - Data-Intensive Scientific Discovery. T. Hey, S. Tansley and K. Tolle. 2009.

Science paradigms

- Last few centuries: **theoretical** science

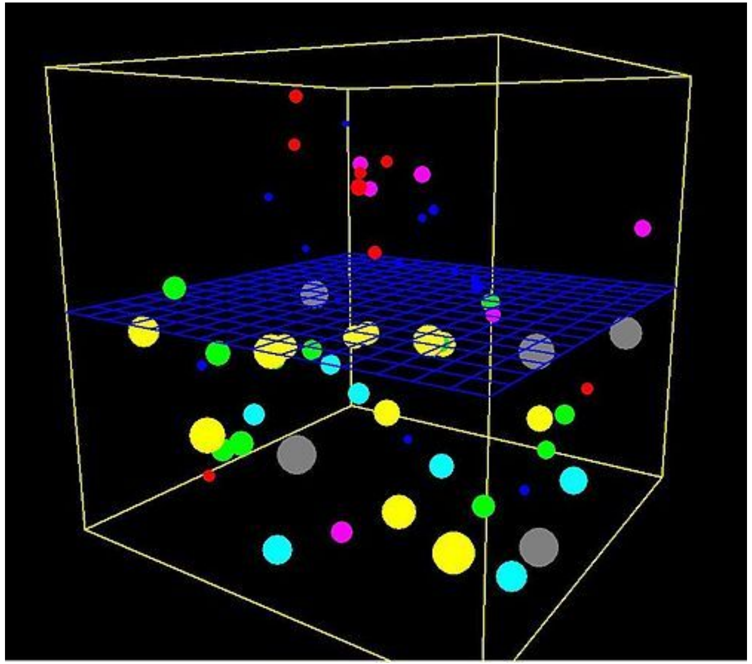
The Fourth Paradigm - Data-Intensive Scientific Discovery. T. Hey, S. Tansley and K. Tolle. 2009.

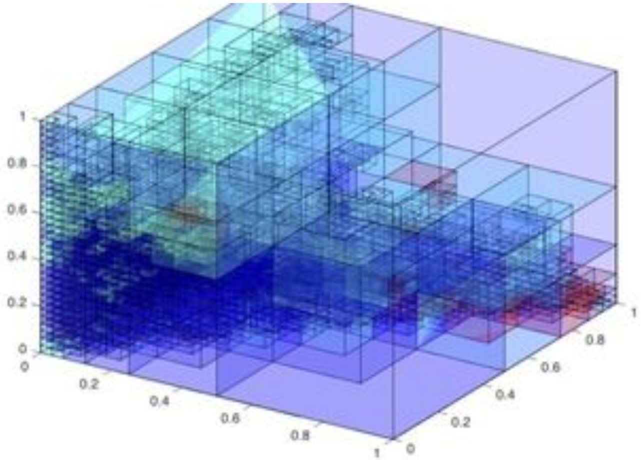


Science paradigms

- Last few decades: **computational** science

The Fourth Paradigm - Data-Intensive Scientific Discovery. T. Hey, S. Tansley and K. Tolle. 2009.

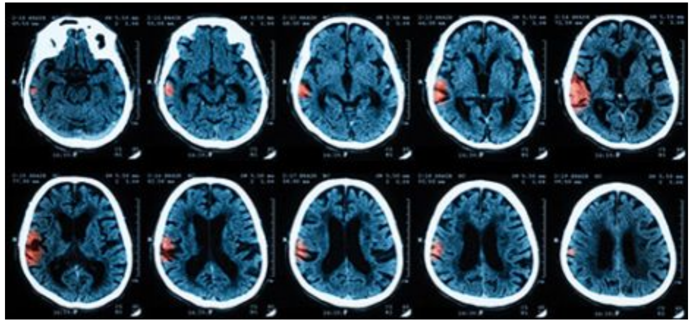




Science paradigms

- Today: **data exploration** (eScience)
  - Data captured by instruments/generated by a generator
  - Processed by software
  - Information/knowledge stored on a computer
  - Analysis of data

The Fourth Paradigm - Data-Intensive Scientific Discovery. T. Hey, S. Tansley and K. Tolle. 2009.

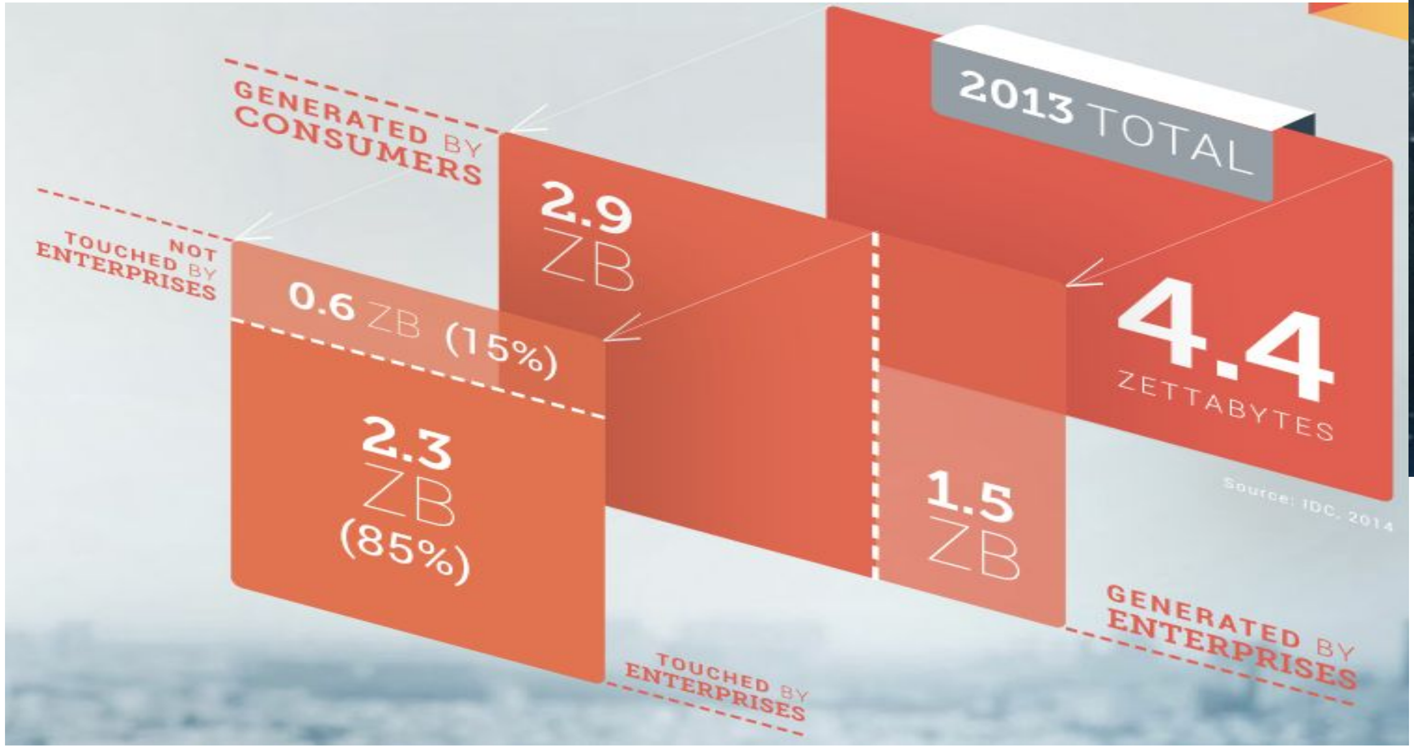


What is Big Data?

Big Data is data that is **difficult to process** and extract **value** from.

Why is it difficult?

Volume: The "Data Deluge"

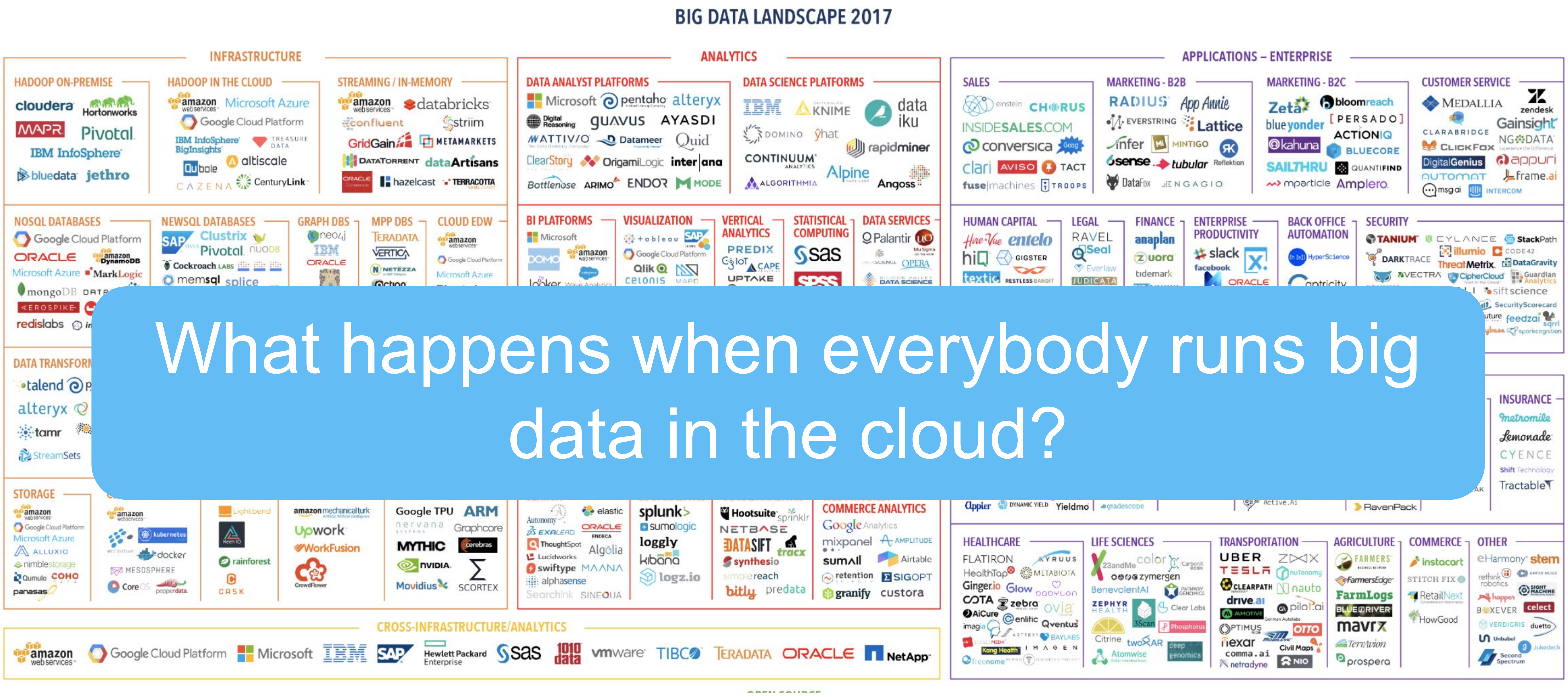


Many Vs of Big Data:

- **Volume**: the amount of data to process
- **Velocity**: the rate at which new data arrives
- **Variety**: different forms of data
- **Veracity**: uncertainty of data

How do we explore and extract value from big data?

Wide variety of frameworks

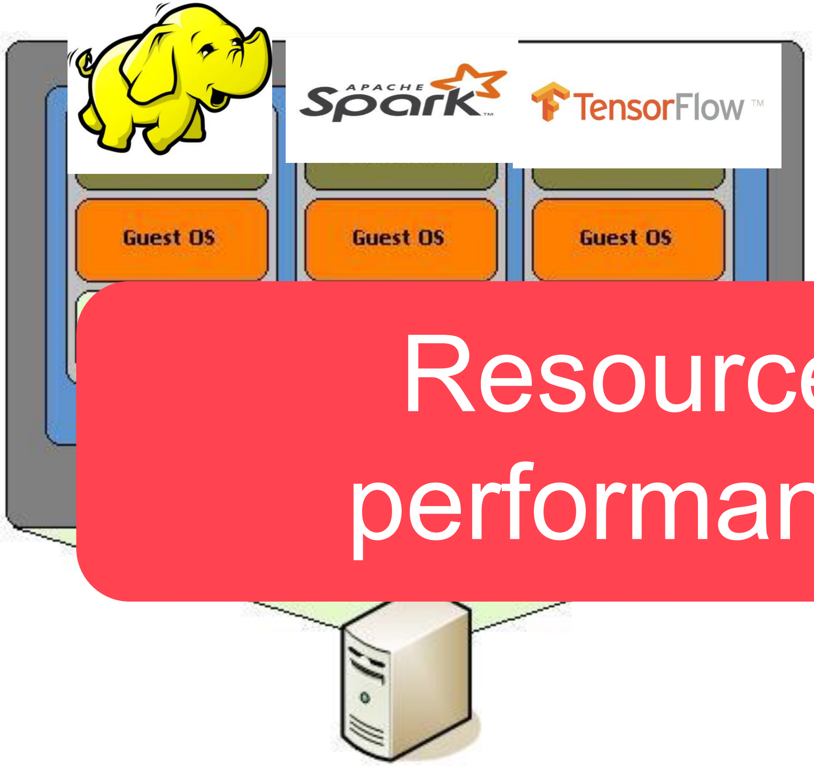


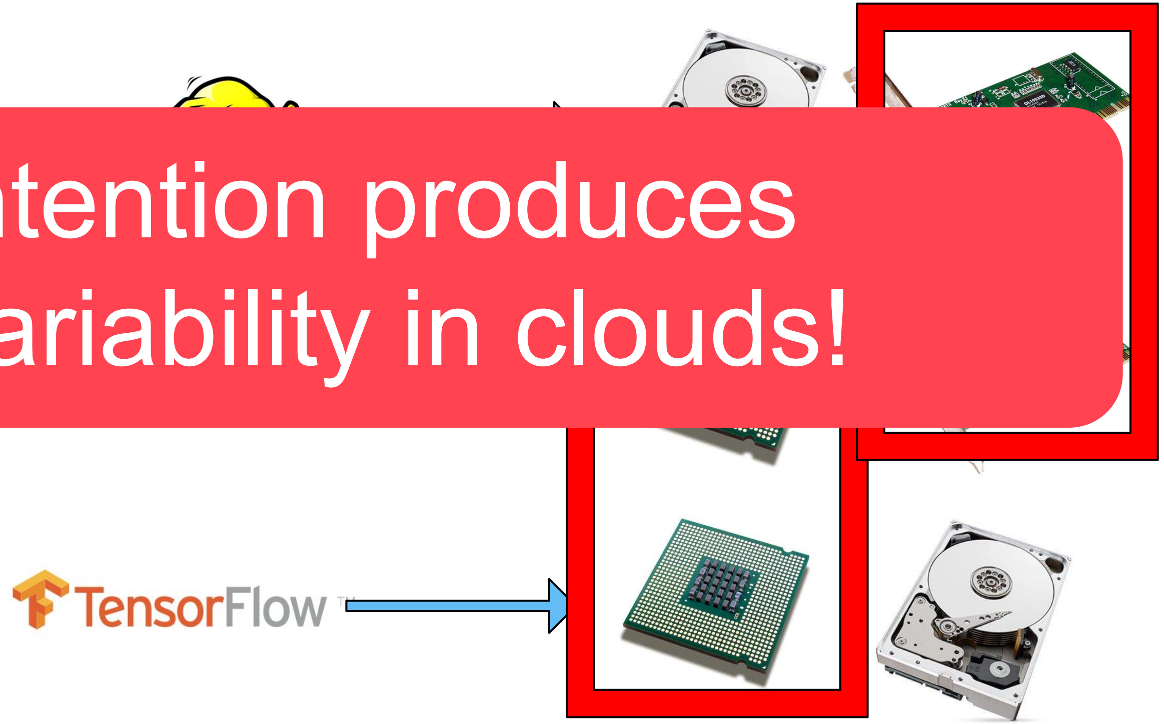
What happens when everybody runs big data in the cloud?

Image courtesy of mattturck.com

Co-location induces (resource) performance variability

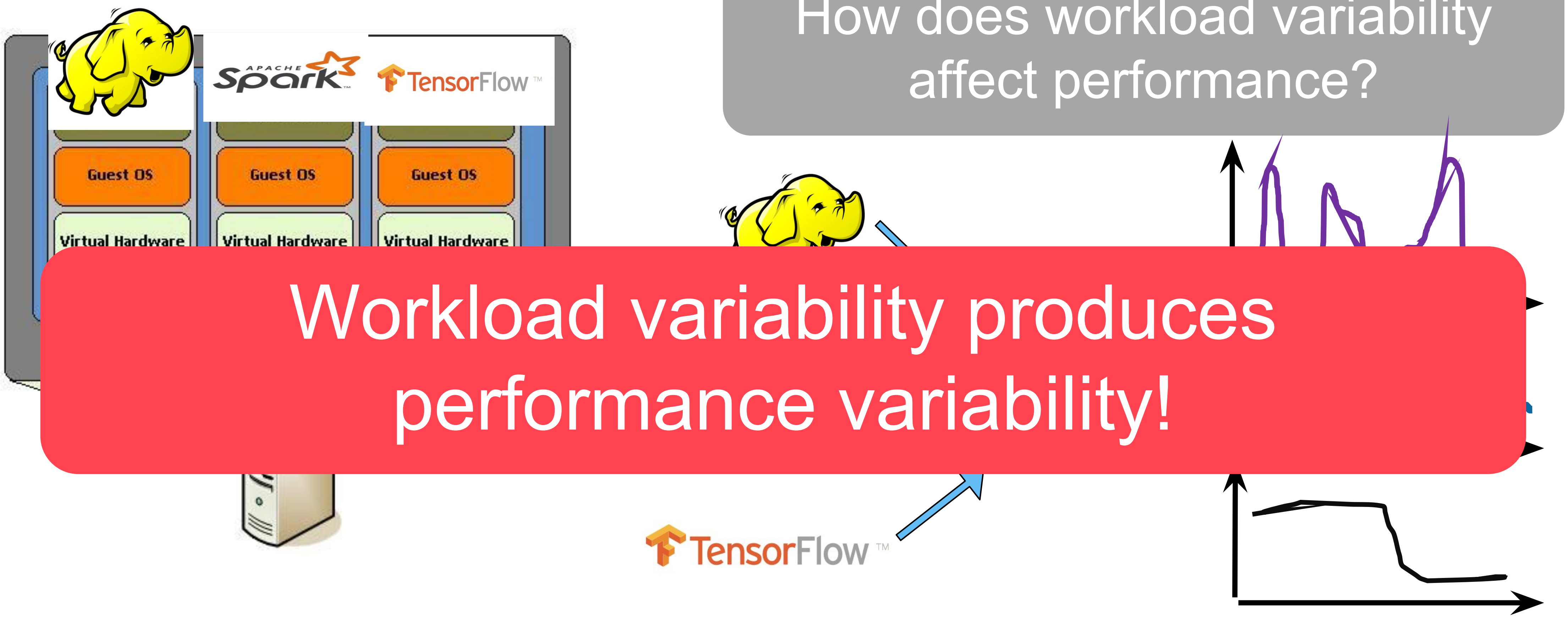
How does resource interference affect performance?





Resource contention produces performance variability in clouds!

Co-location induces (resource) performance variability

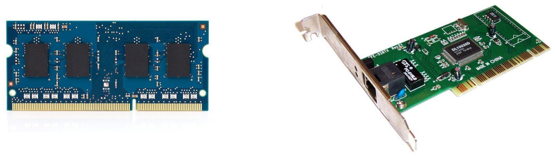


How does workload variability affect performance?

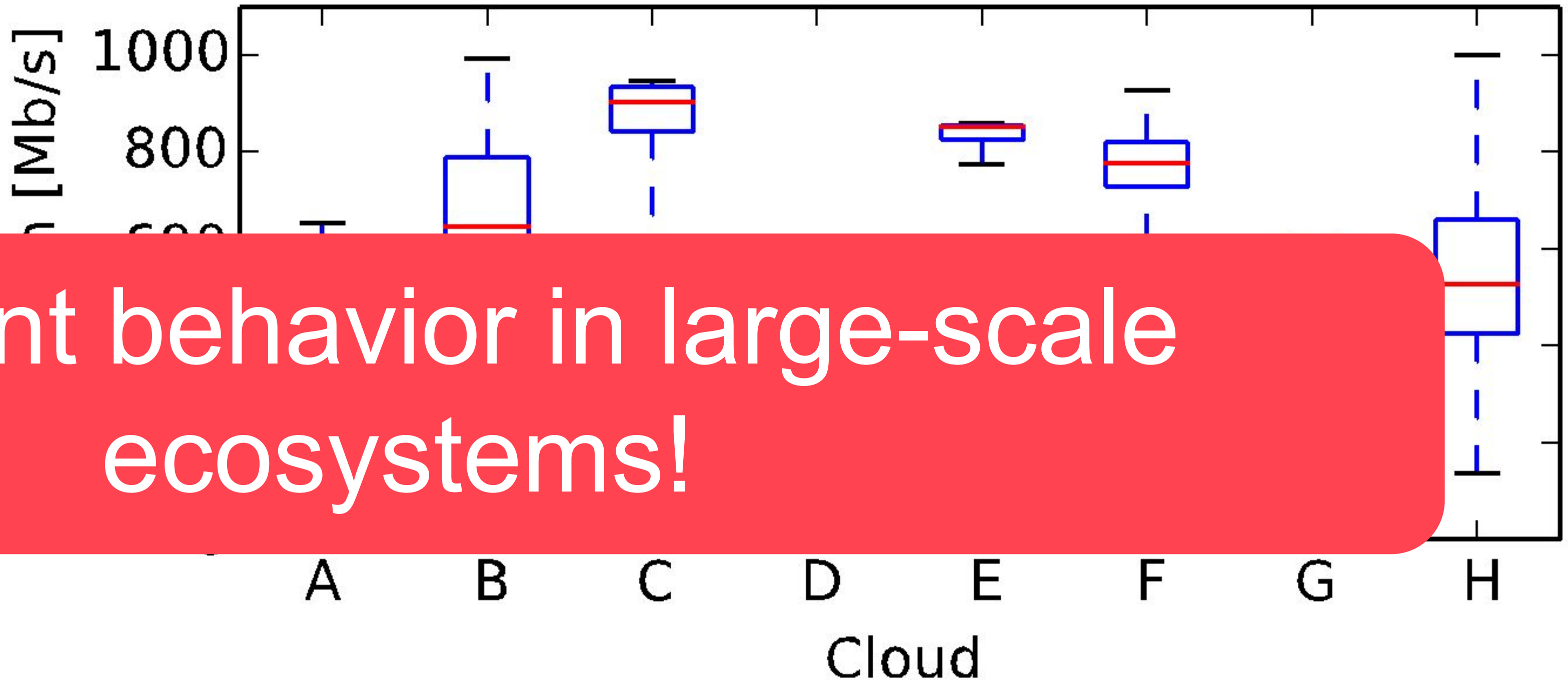
Workload variability produces performance variability!

Cloud (resource) performance is highly variable!

- Due to:
  - Co-location
  - Virtualization
  - Workload variability
  - Net
- Affec



Emergent behavior in large-scale ecosystems!



Ballani et al., SIGCOMM 2011

Convenient to use big data + cloud, but…

Variability entails:

- **Poor performance predictions**
- **Poor scheduling decisions**

How to study performance variability?
How to control the variability?

How to study performance variability?

Traditional performance analysis:

- **(1) Trace analysis**
- **(2) Benchmarking**
- **(3) Performance modeling**

Current models, benchmarks do not consider resource variability!

- No study on resource performance variability and big data
- Variability **within** clouds and **between** clouds (performance portability issues)

VU VRIJE UNIVERSITEIT AMSTERDAM

A Framework for Studying Performance Variability

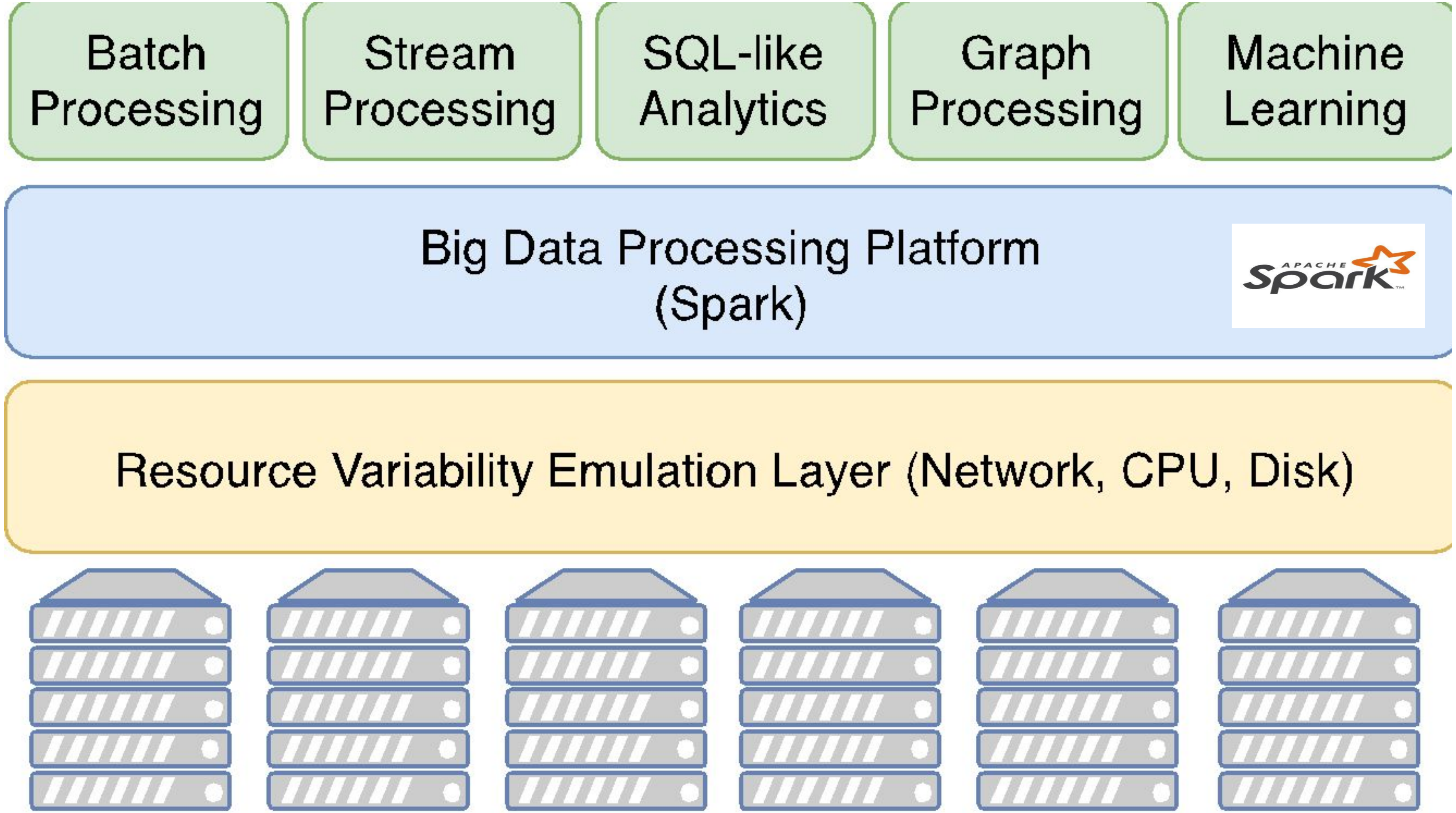
1 • Fallback to empirical evaluation based on previous observations

2 • Controlled environment that emulates real-world variability scenarios

3 • Multiple classes of big data applications
• Statistical analysis and performance modeling to understand correlations

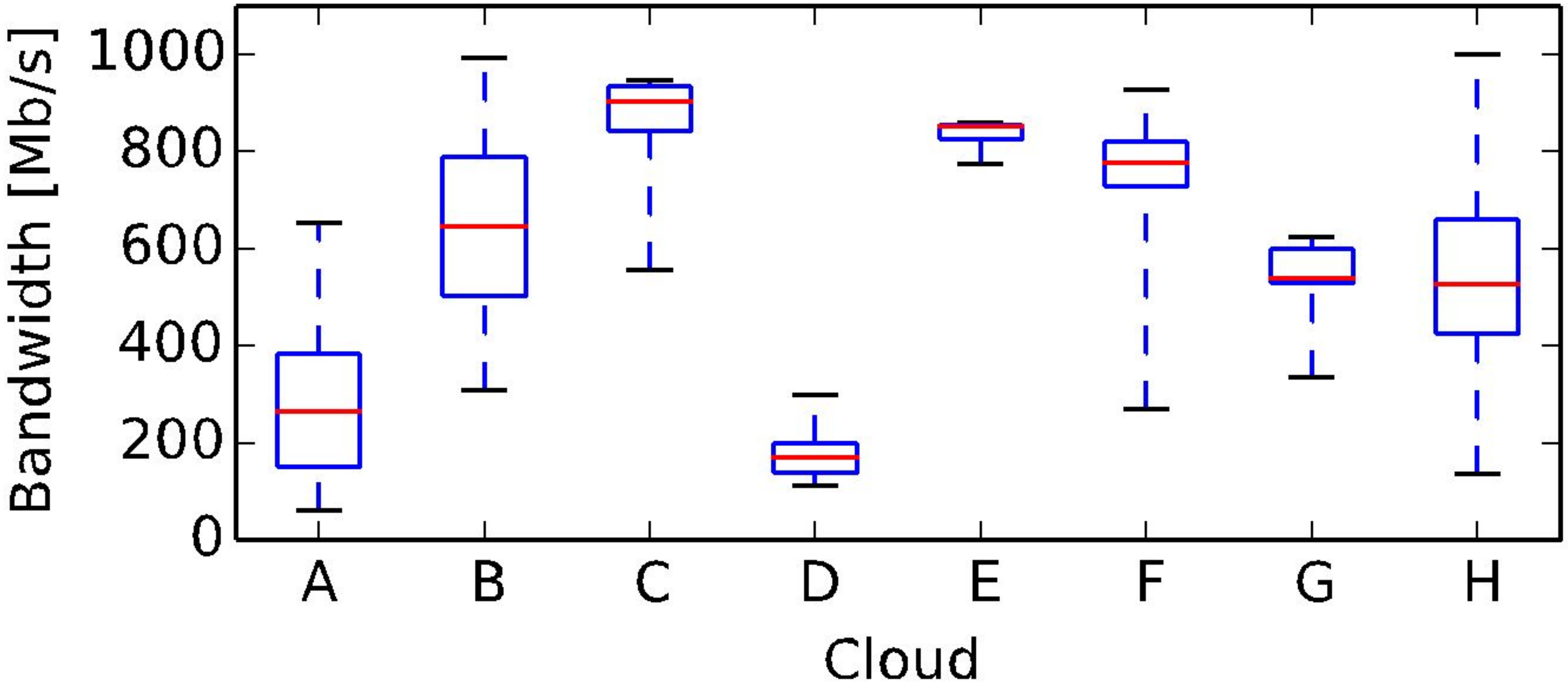
(1) Traces → (2) Benchmark → (3) Modeling

Benchmarking Performance Variability



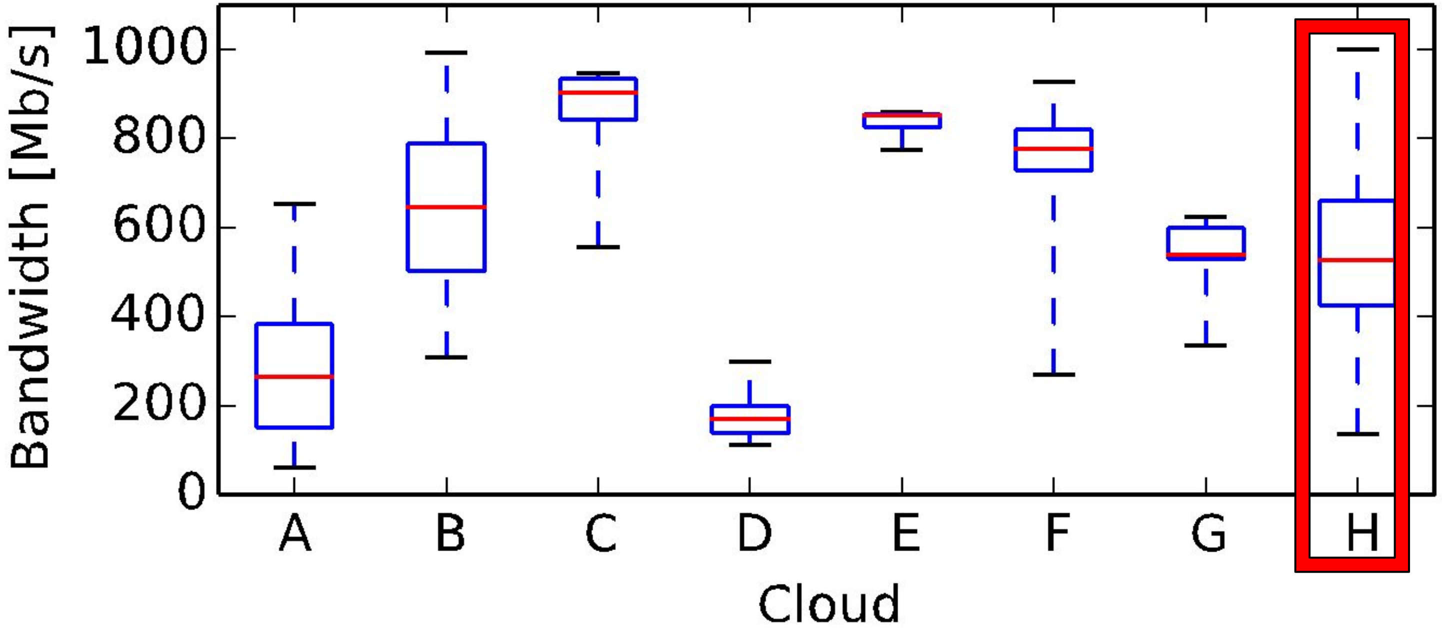
Quantifying network variability impact on Big Data

- Systematic study using A-H cloud bandwidth distributions
- Run a series of big data applications



Cloud network bandwidth emulation

- For each distribution:

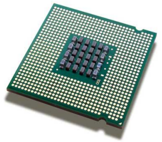


Vary bandwidth

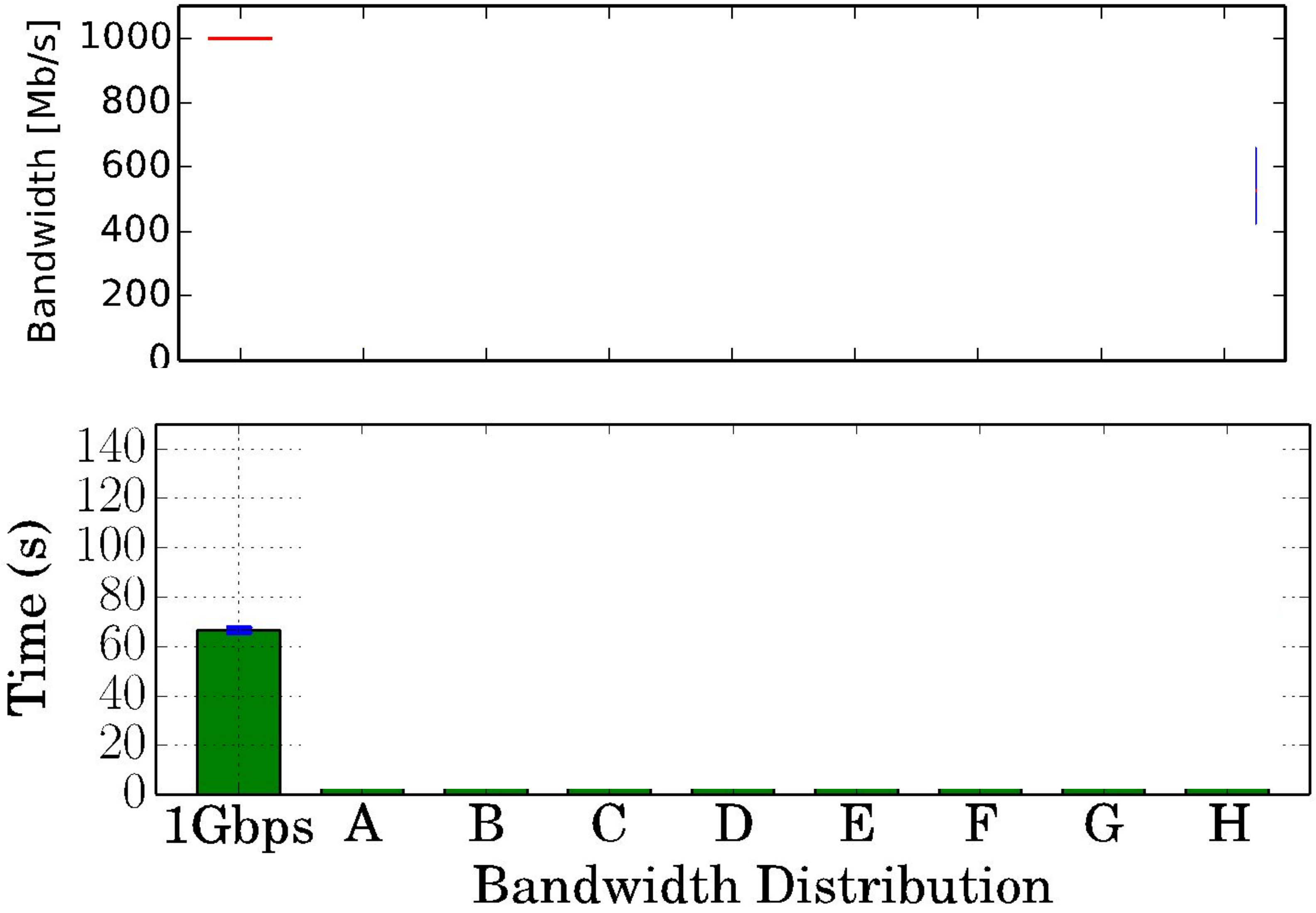
Cluster

Big Data Workloads

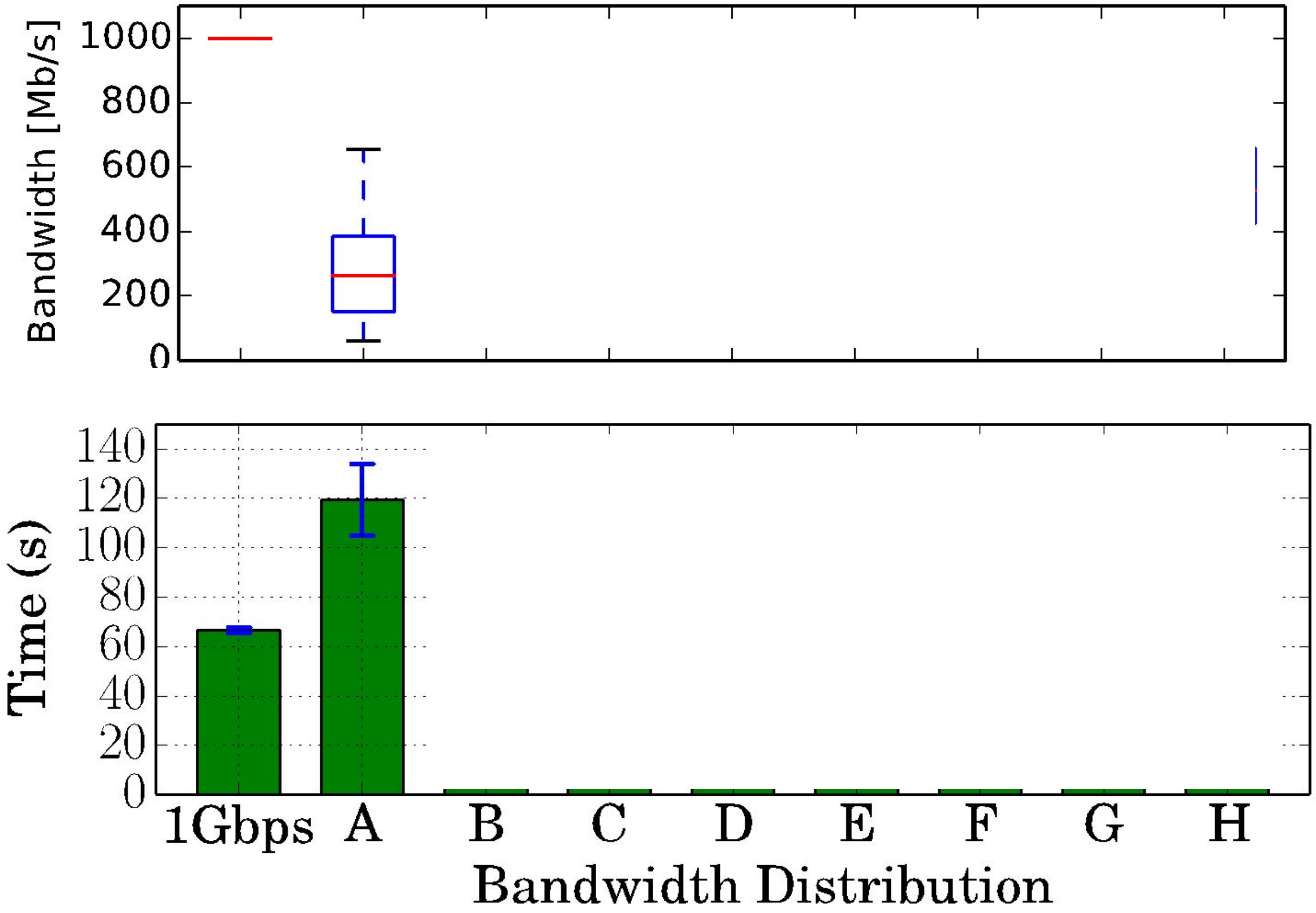
- HiBench suite, MapReduce-style apps
- 6 real-world applications from various domains
- Each app having different resource usage

Application				
Wordcount	++	--	0	0
Sort	--	++	0	++
Terasort	++	0	++	++
Naïve Bayes	0	0	++	--
K-means	++	--	0	--
PageRank	0	--	0	--

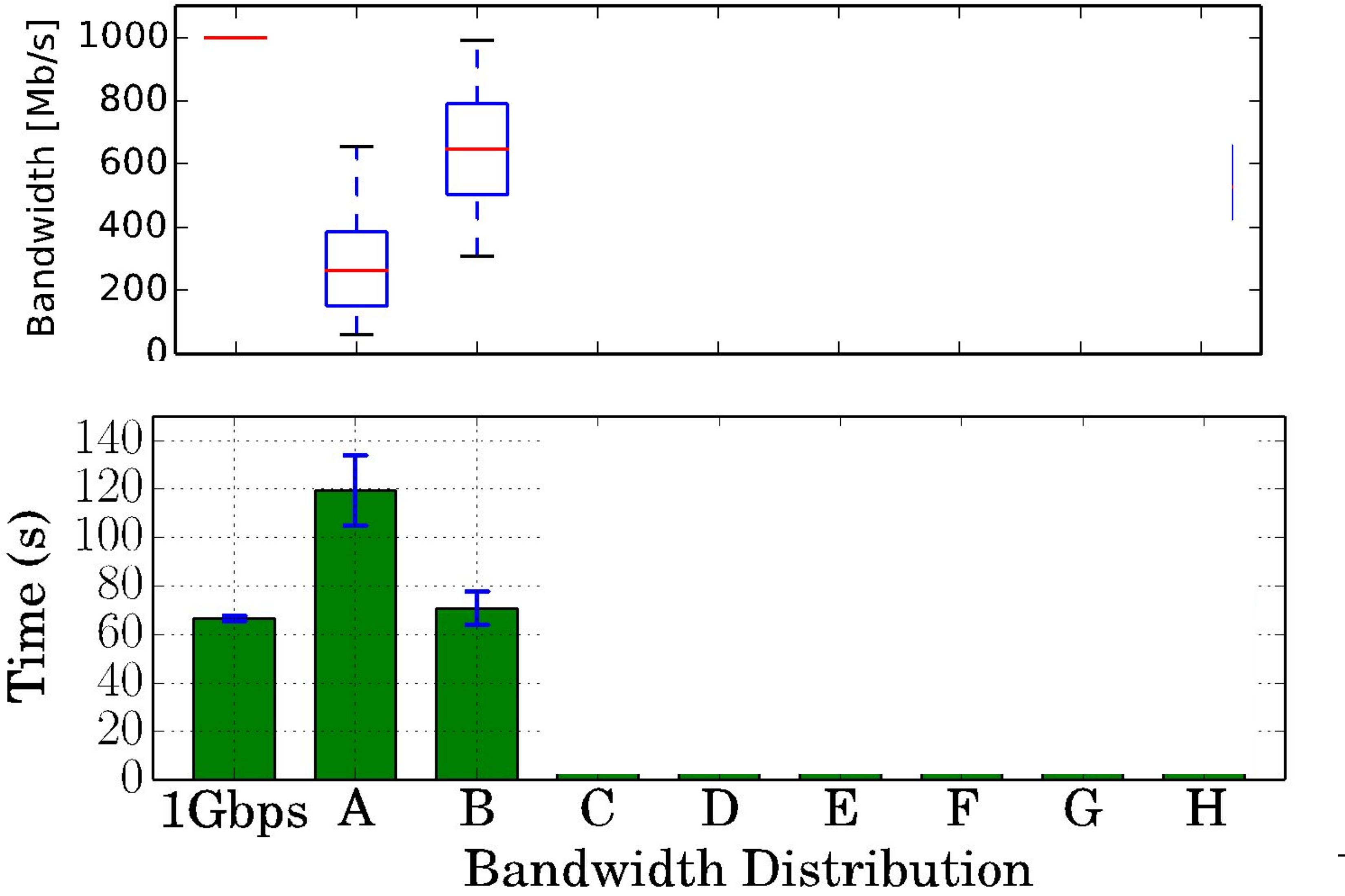
Variable network = Variable Runtime (Terasort)



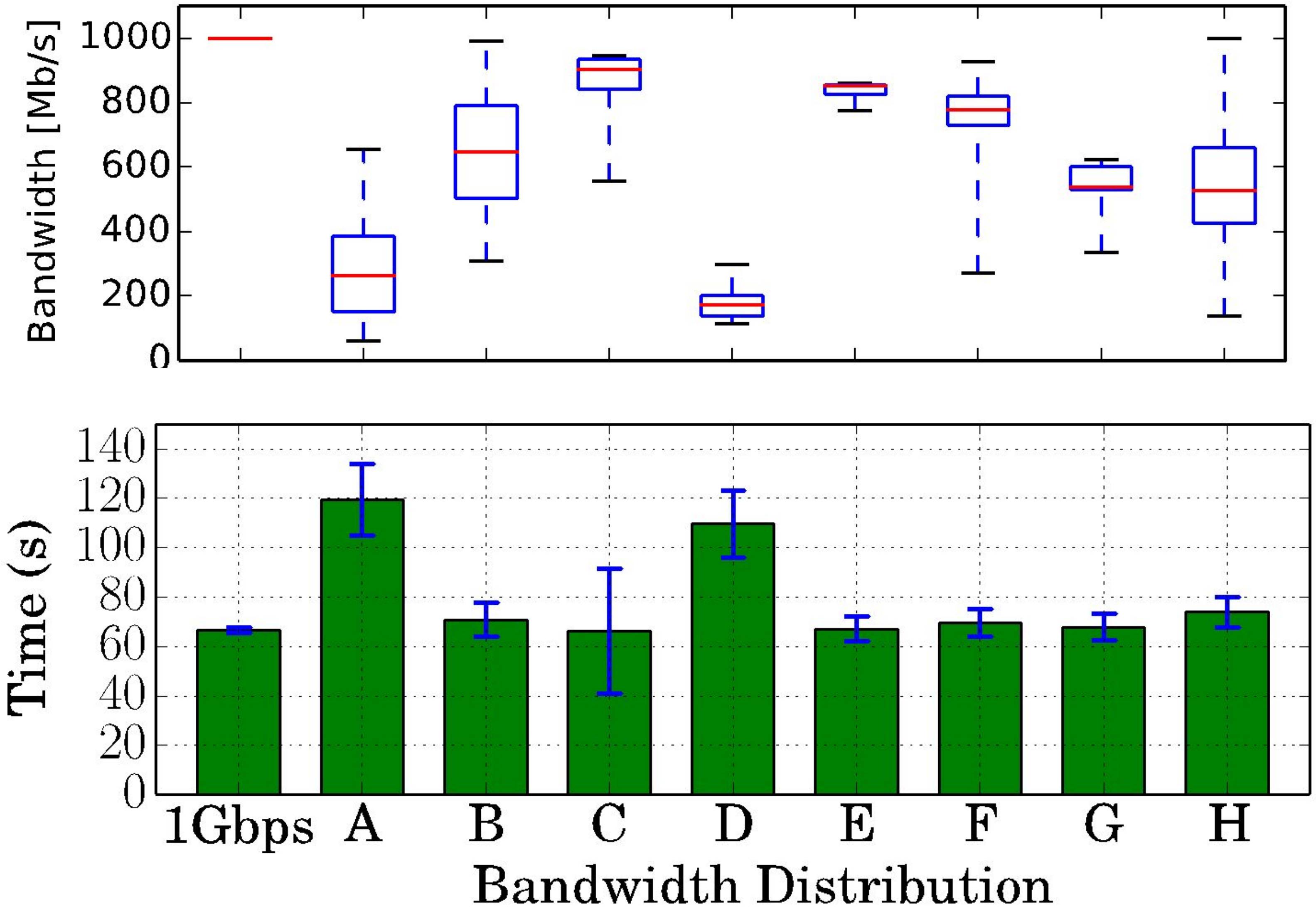
Variable network = Variable Runtime (Terasort)



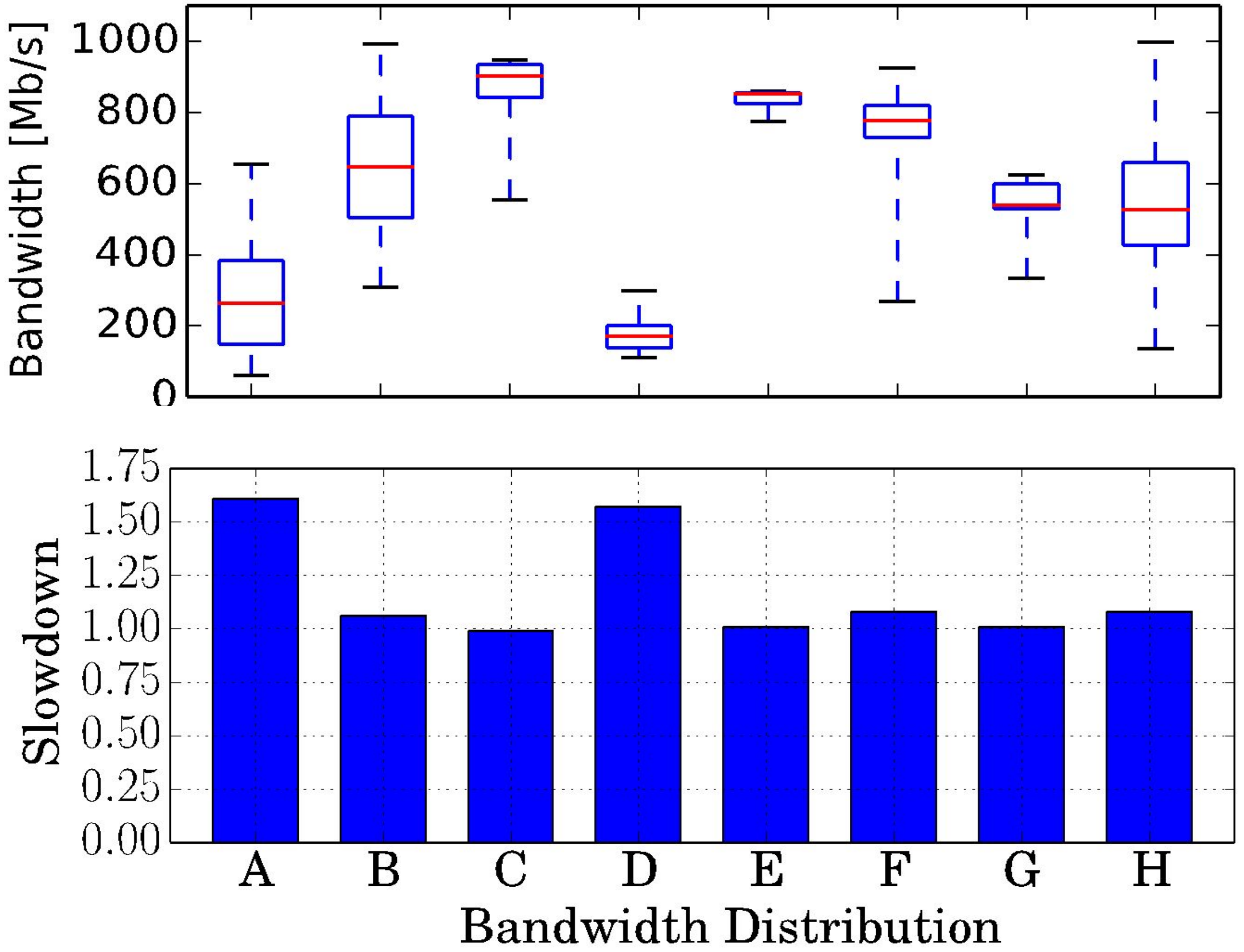
Variable network = Variable Runtime (Terasort)



Variable network = Variable Runtime (Terasort)

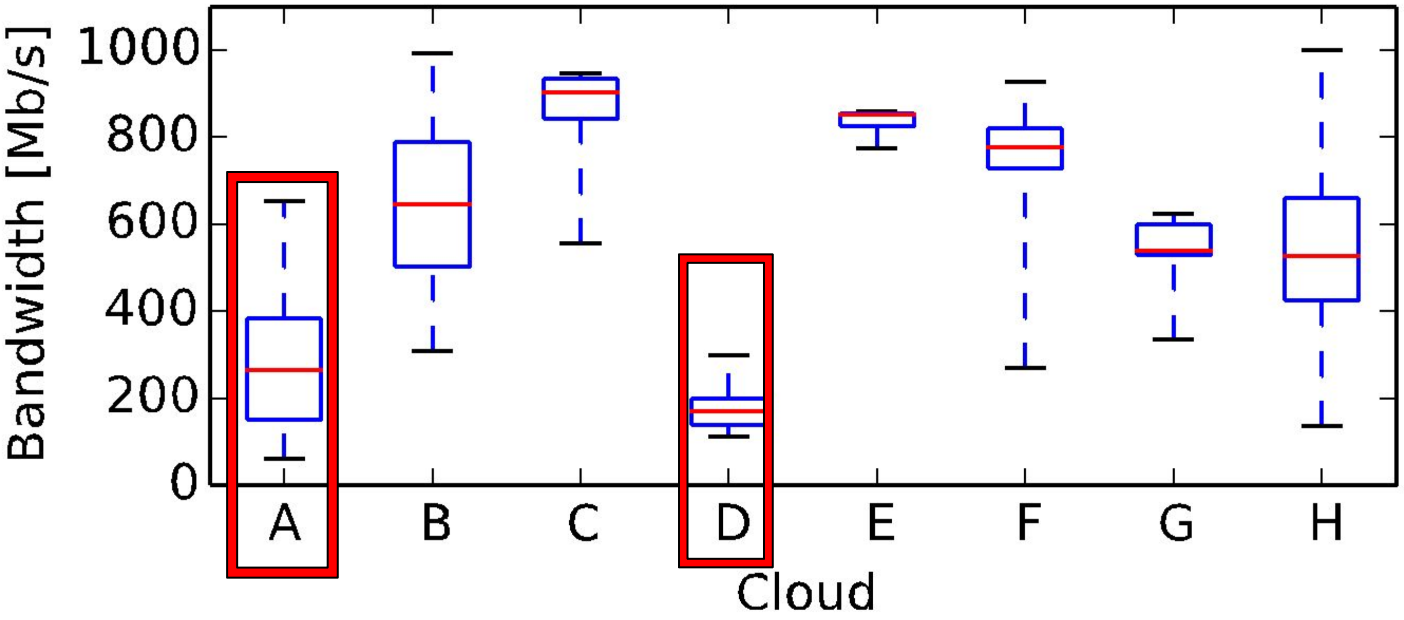


Surprisingly, non-network-intensive Wordcount slowed down



Most apps are slowed down on real clouds

Application	Maximum Slowdown	Bandwidth Distribution
Wordcount	1.61	A
Sort	1.51	D
Terasort	1.79	A
K-Means	1.48	D
Bayes	1.14	A
Pagerank	1.07	A



VU VRIJE UNIVERSITEIT AMSTERDAM

Take-home message

- Network variability leads to high slowdown for big data in the cloud
- Network variability also affects performance portability
- Surprisingly, also apps not network-bound applications slow down

Future work:

- In-depth statistical analysis
- Performance modeling tools
- Control through better scheduling

Exploring Computing Infrastructure Convergence: HPC and Big Data Graph Processing on Multicores

Do you have experience with … ?

OPEN MPI

HIVE

APACHE Spark™

NVIDIA® CUDA®

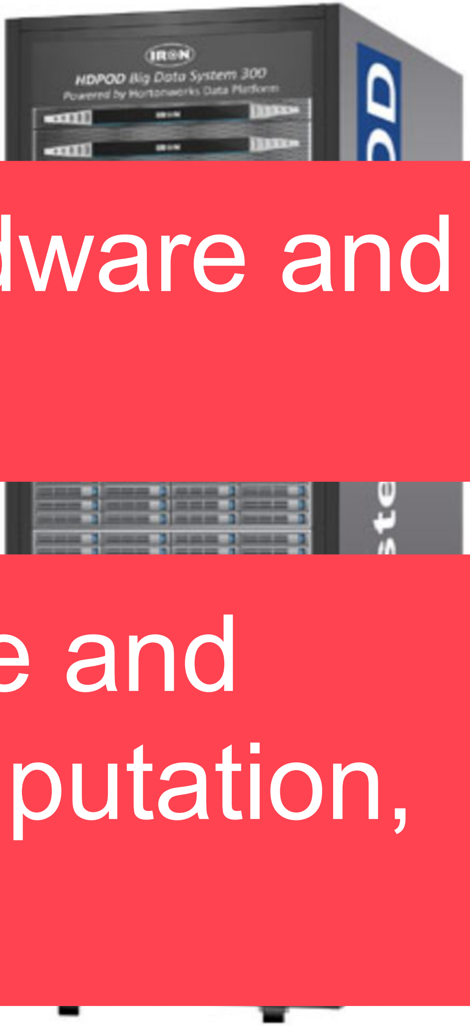
OpenCL

hadoop HDFS

OpenACC

More Science, Less Programming

HPC and Big Data Infrastructure



Highly divergent in both hardware and software!

Divergence is expensive and unsustainable: energy, computation, human resources!

Divergence - unsustainable and expensive!

How does the hardware and software landscape look for these paradigms?

HPC Infrastructure

- Large numbers of (thinner, low-power) cores
- Intricate NUMA topologies
- Fast interconnects (InfiniBand, 40+ Gb Ethernet)
- Accelerators (GPUs, FPGAs, TPUs)
- Compute-intensive workloads (simulations)

Big Data Infrastructure

- (generally) commodity hardware
- Fat-core CPUs
- large memory (and caches) per core
- Large storage
- Less emphasis on fast networks
- Often virtualized clusters (cloud)
- Data-intensive workloads

HPC vs. Big Data Software

OPEN MPI

MVAPICH

HIVE

APACHE Spark™

Most big data stacks are unable to take advantage of (HPC) hardware features.

Distributed • Resilient • Real-time

slurm workload manager

OpenACC

More Science, Less Programming

MESOS

Addressing the HPC and Big Data Convergence

- **Only in software: porting big data to HPC hardware**

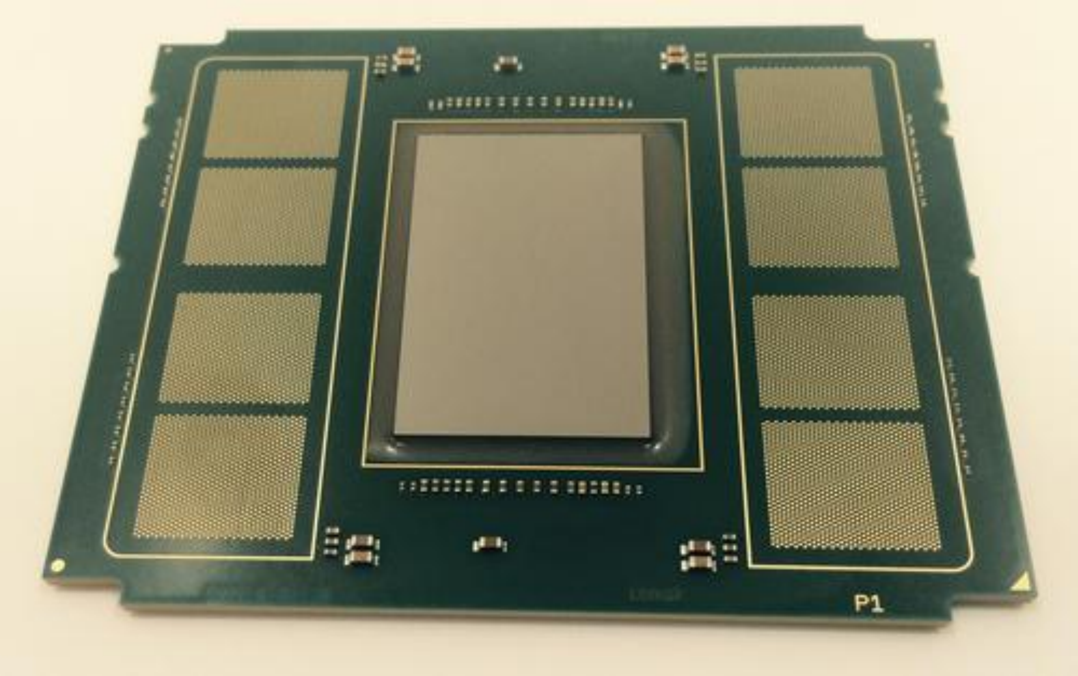
Significant effort in porting and tuning!

Can we run big data directly on HPC hardware? What are the trade-offs?

Big Data on HPC-capable Many-cores

Representative:

- Intel KNL – 2nd generation Xeon Phi



Can run Big Data:

- Accelerator-like self-booting CPU
- Full x86_64 compatibility

HPC Features:

- (up to) 72 low-power Intel Atom cores
- Wide vector instructions (512B)
- 16GB high-bandwidth on-chip memory

Intel KNL – Highly Representative for HPC

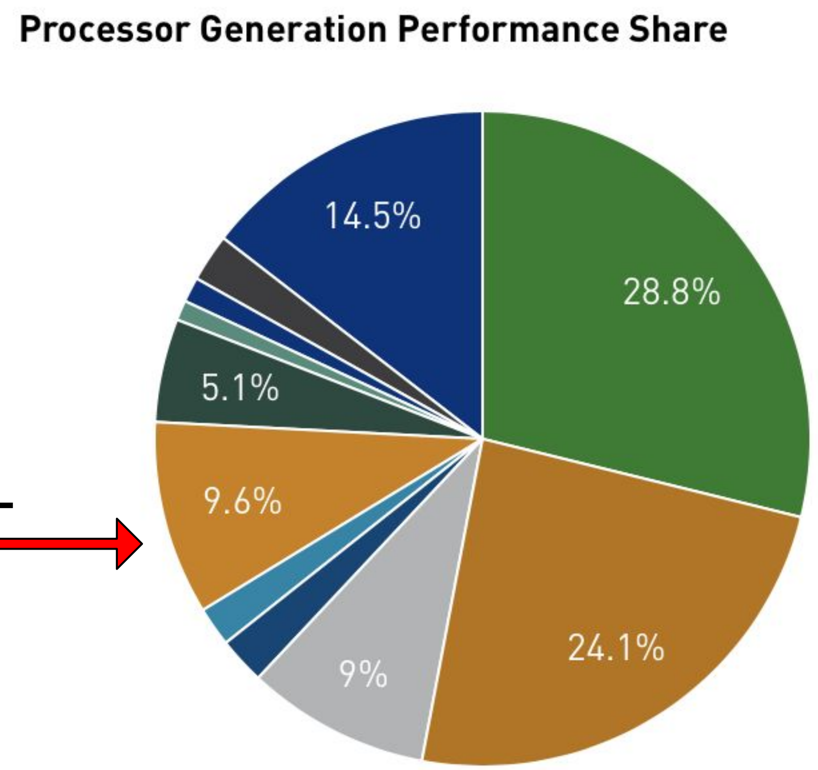
Representative for Top500:

- 3 clusters in top 10 of top500.org contain KNL
- ~3% of the share of CPUs in top500
- ~10% of the performance share of top500

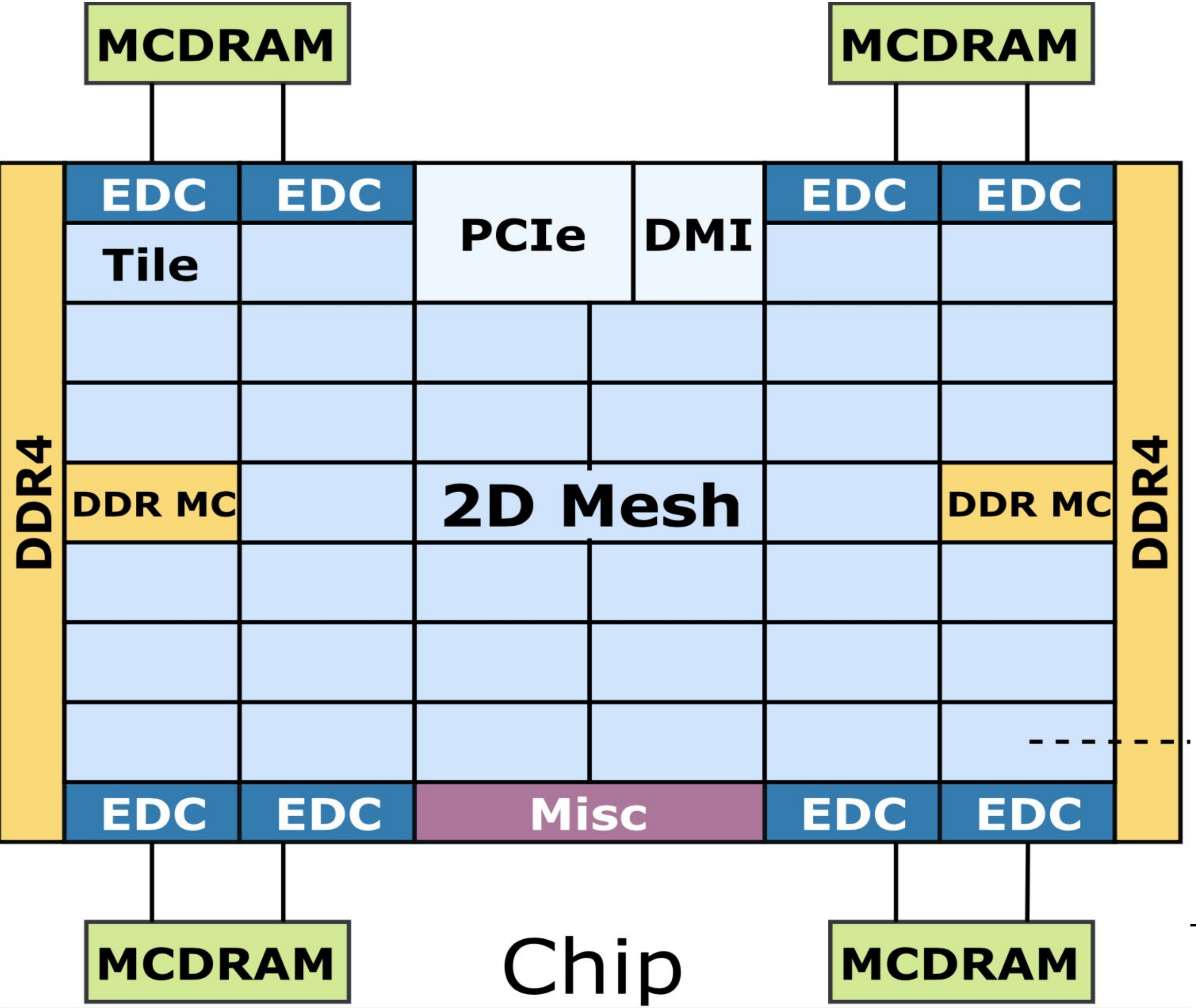
Many performance facets:

- Highly configurable at boot time
- Works as many different machines (due to configurable clustering and memory modes)

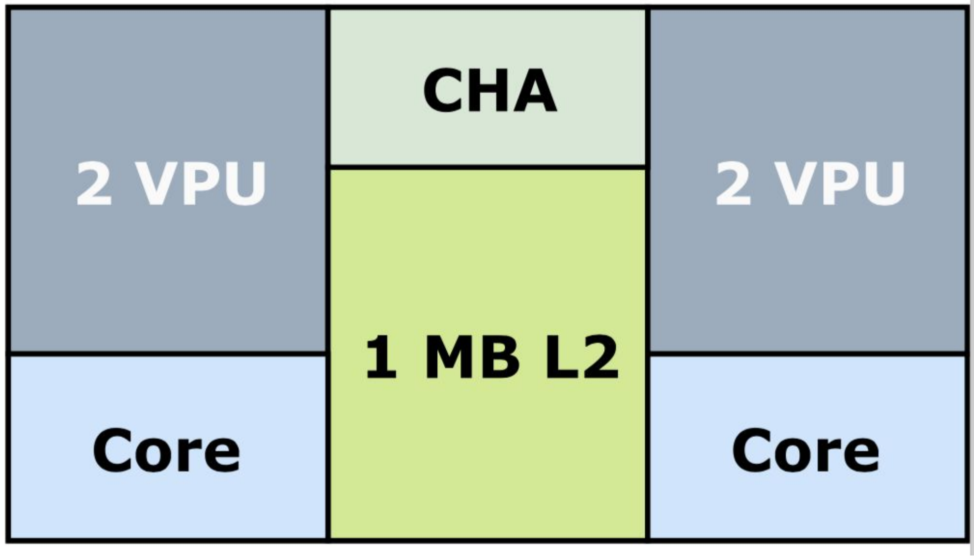
KNL



KNL Architecture



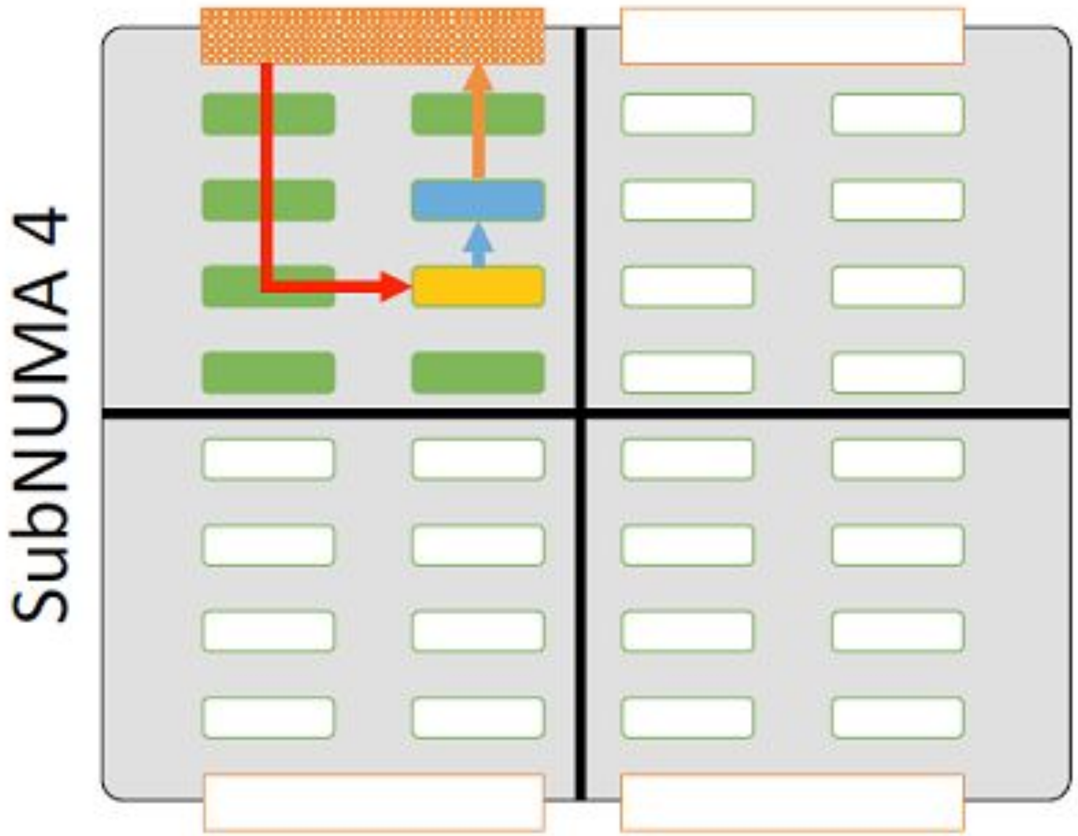
Chip

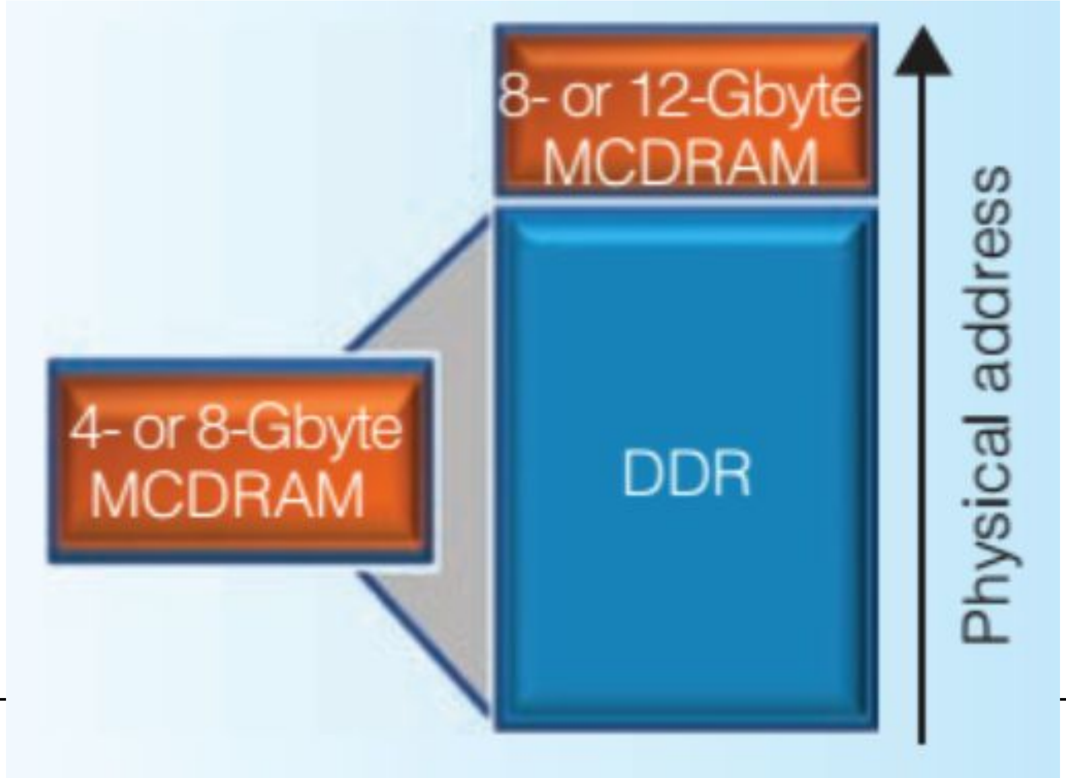


Tile

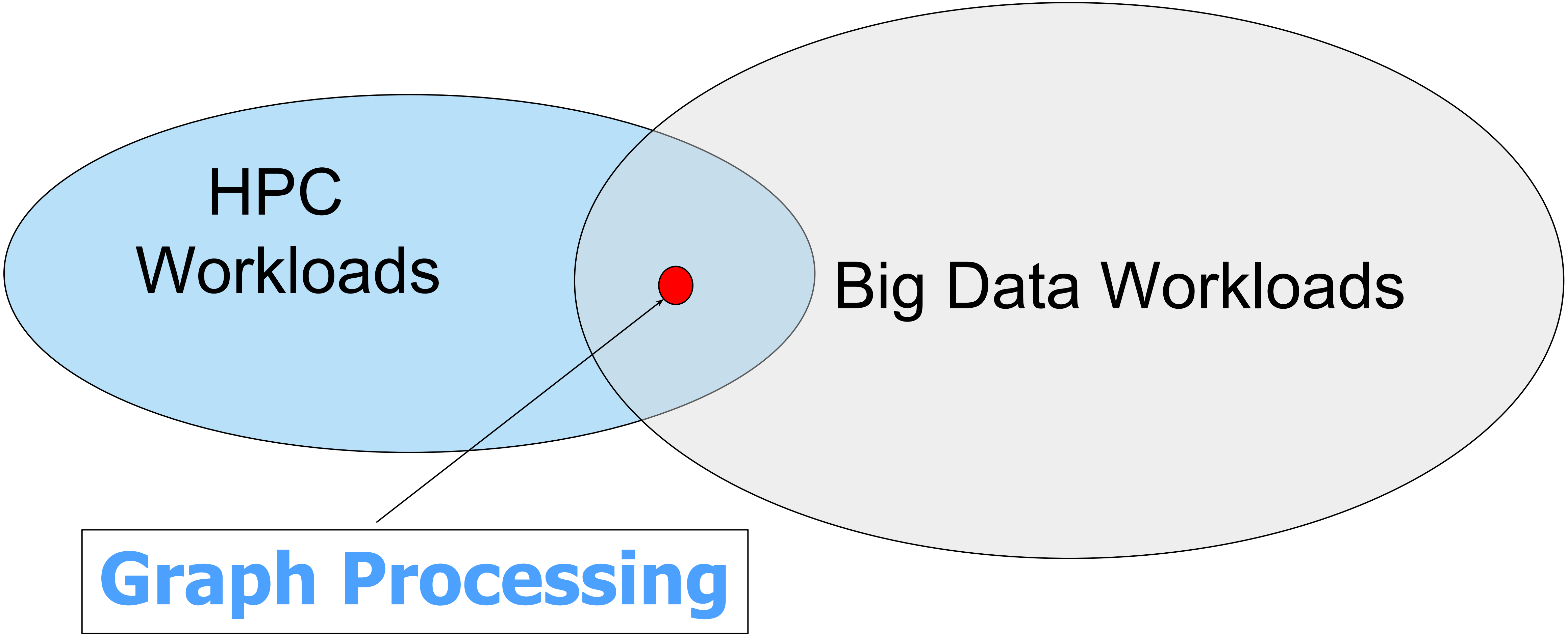
KNL – Hardware Parameter Space

- Clustering modes: (L2 cache miss latency)
  - All2All
  - Quadrant/Hemisphere
  - NUMA



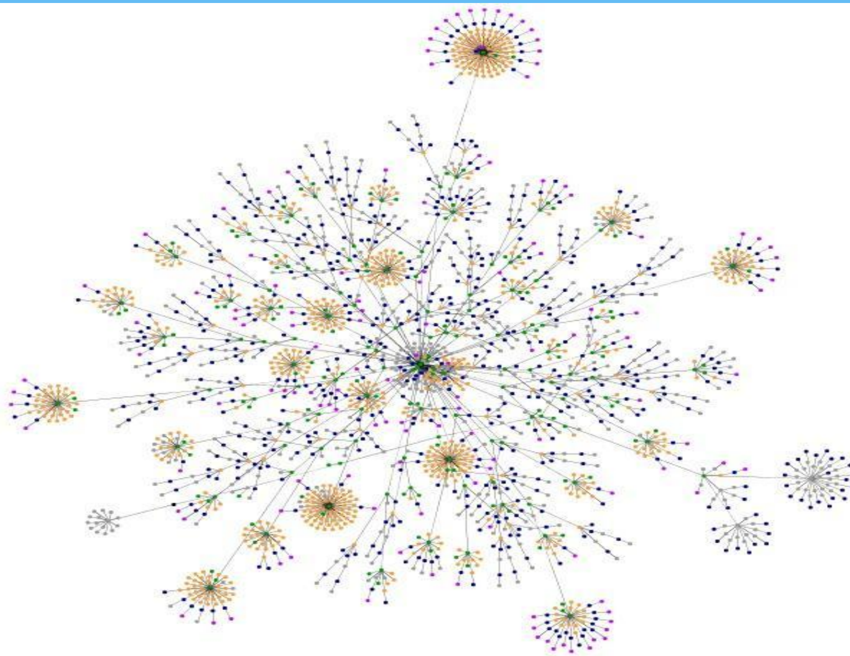


Graph Processing – HPC and Big Data



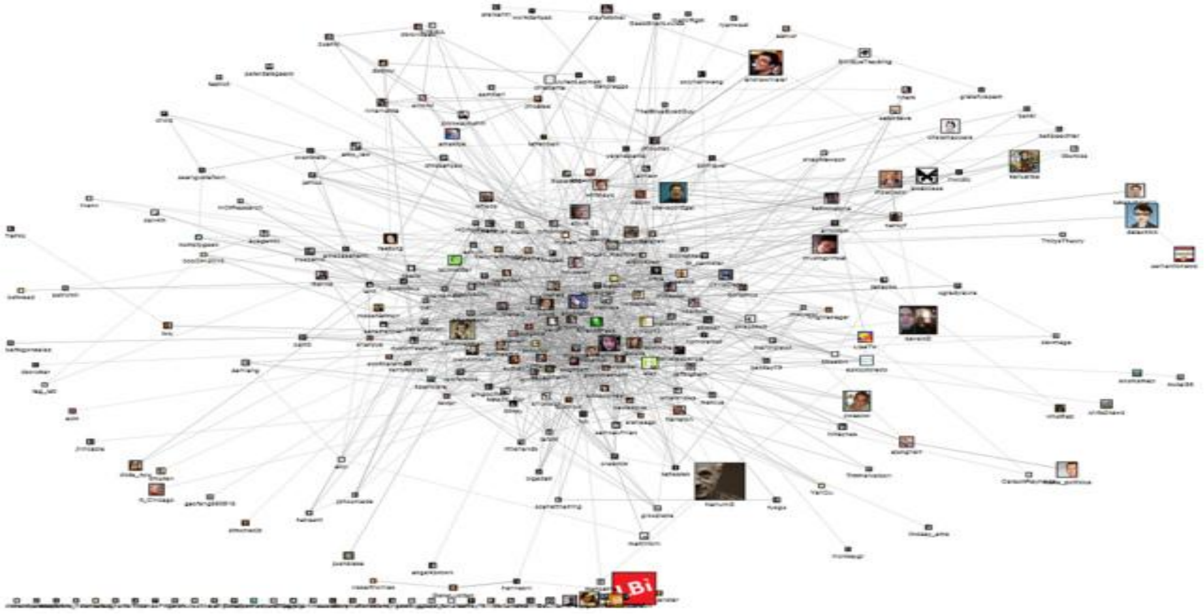
Graph Processing – High-impact Domain

- Social networks
- Drug discovery
- Monitoring wildfires
- Combating human-trafficking
- Studying the human brain



Graph Processing – Highly Challenging

- Mostly traversing links between entities
- Little computation
- Mostly memory bound
- Highly irregular workloads
- Cache misses



Performance = f(platform, algorithm, dataset)

[1] Guo et al., IPDPS ’14 ; [2] Iosup et al., VLDB ‘16

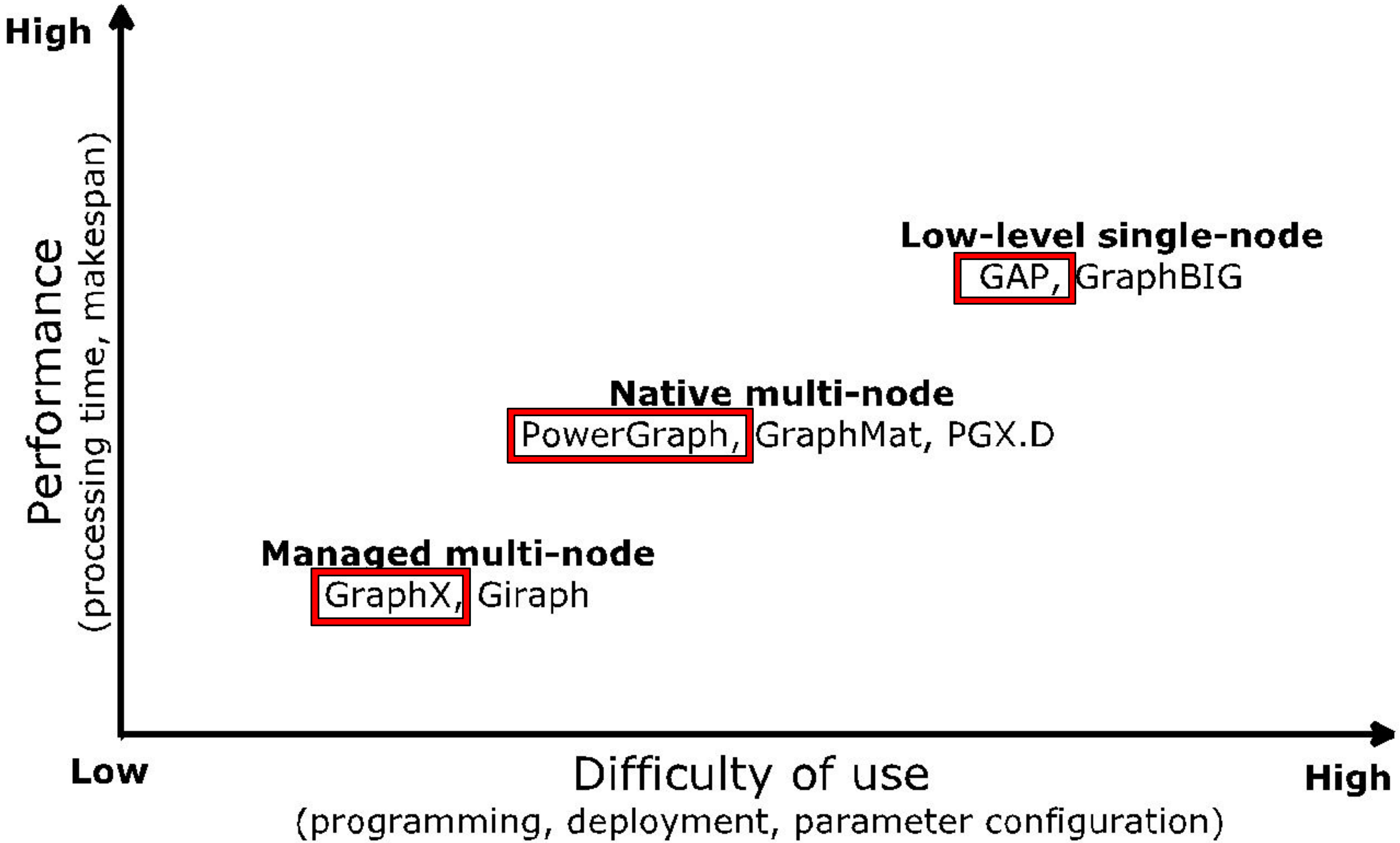
How to study the convergence?

- Benchmark using Graphalytics
- Multiple classes of algorithms
- Multiple datasets (scale-free and non-scale free)
- Multiple classes of graph analytics platforms
- Comparison between KNL and de-facto big data hardware (Intel Xeon family)

Graphalytics

Open-source Graph Processing Benchmark Suite

Graph Analytics Platforms



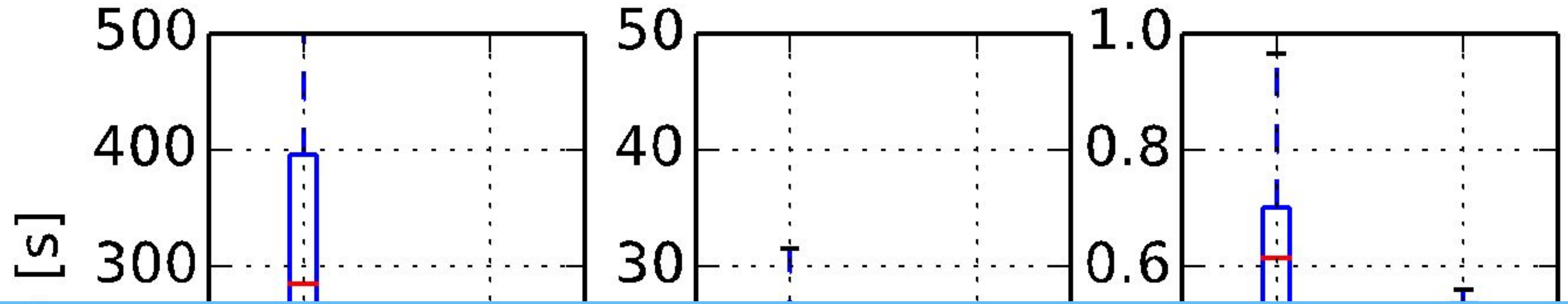
Quantifying the Convergence

- Large-scale study – over 300,000 compute core-hours
- Experiments run in DAS-5, Cartesius cluster*, Intel Academic cluster*
- **Q1: How does the KNL parameter space influence performance?**
- **Q2: How (difficult it is) to tune the platforms on KNL?**
- **Q3: Is KNL faster than Xeon**
- **Q4: Does it scale?**

	Xeon E5-2630v3	Xeon Phi 7230
Cores	16 (32 hyperthreads)	64 (256 hyperthreads)
Frequency (GHz)	2.4	1.3
Network	56Gbit FDR InfiniBand	56Gbit FDR InfiniBand
Memory	64GB DDR4	96GB DDR4
OS	Linux 3.10.0	Linux 3.10.0

* Thanks to grants from NWO and Intel

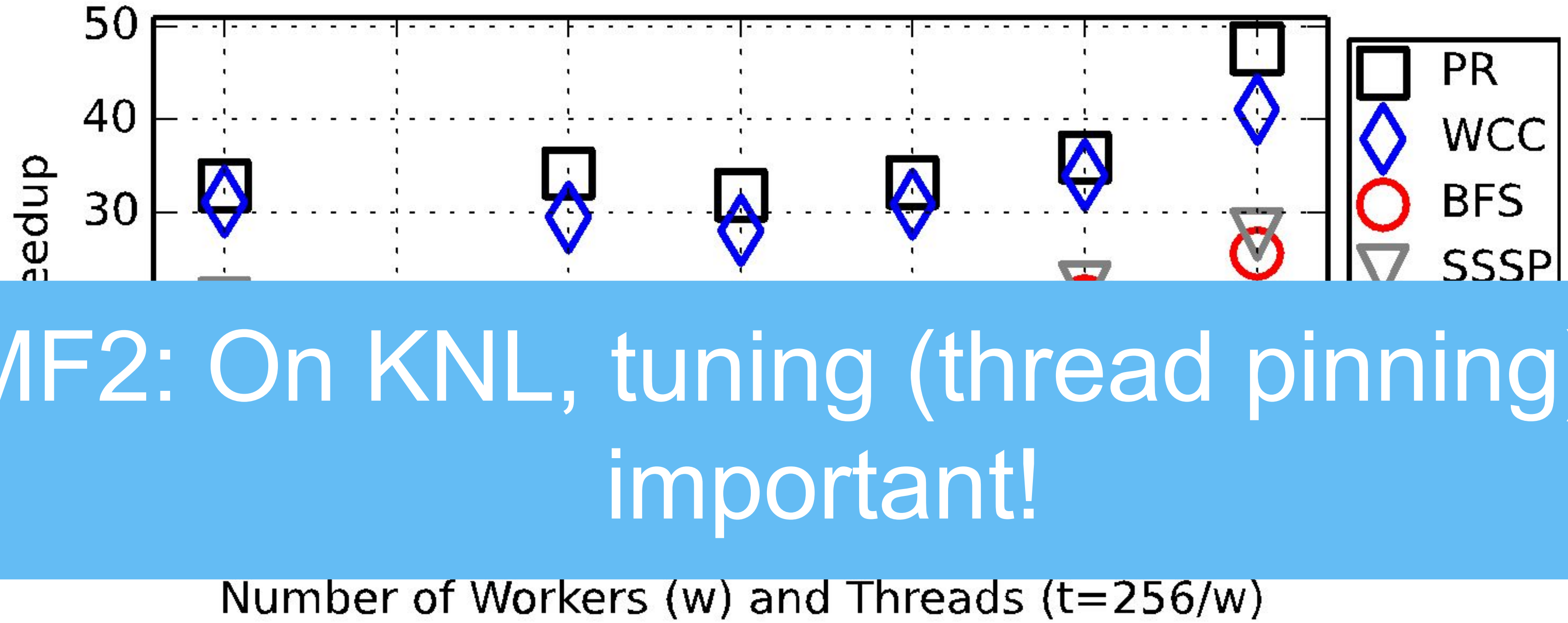
Hardware + Software Parameters



MF1: Much larger performance range due to KNL configurability and interactions with software!

(a) GraphX (b) Powergraph (c) GAP

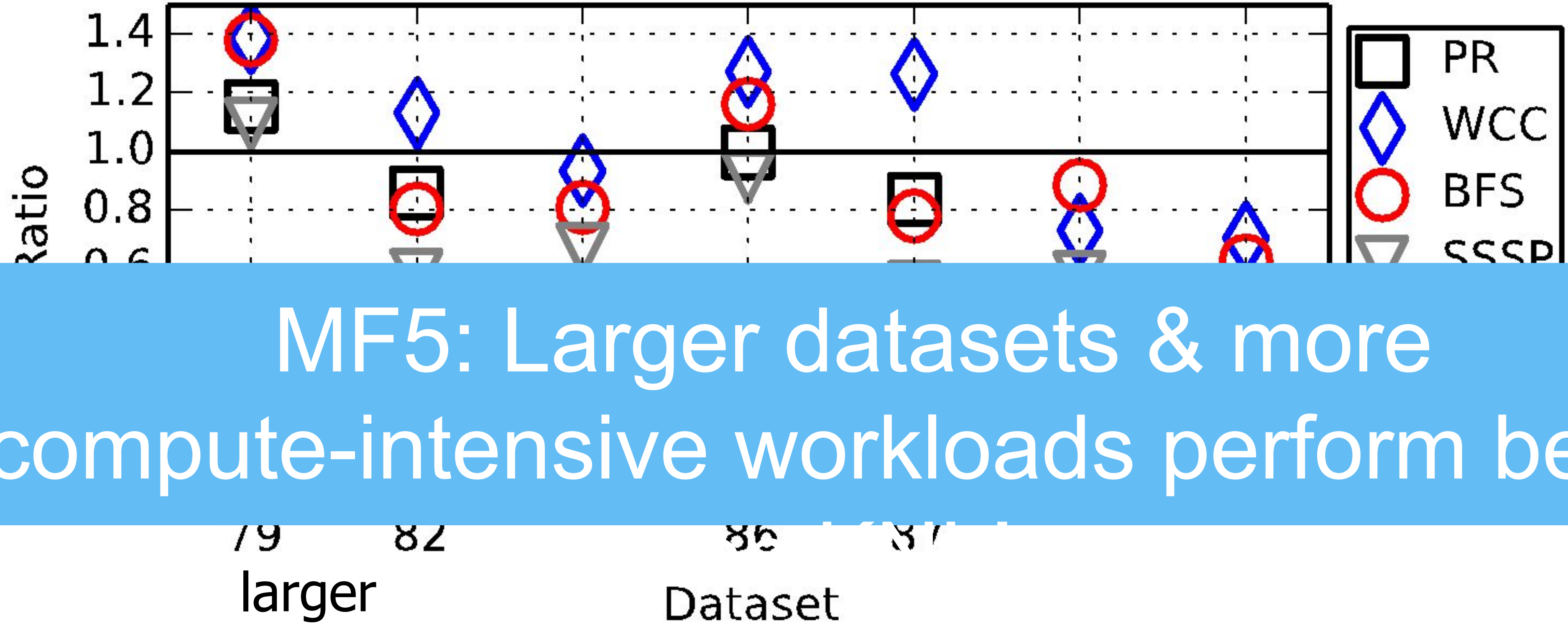
KNL Hardware + Platform Interaction and Tuning



MF2: On KNL, tuning (thread pinning) is important!

Powergraph, Datagen_7-9 – thread pinning speedup
(pinning on Xeon – 5% improvement)

KNL outperforms Xeon



MF5: Larger datasets & more compute-intensive workloads perform better

larger

GAP, KNL vs. Xeon Speedup

Take-home Message: Main Findings

- **HPC & Big Data can converge at a hardware level! But…**
- MF1: **HPAD** – hardware adds an extra complexity layer
- MF2: **Tuning** – good performance entails significant tuning for KNL
- MF3: **Scaling** – KNL scales well vertically, but cannot scale horizontally
- MF4: **H-P interaction** – platforms closer to hardware perform better on KNL
- MF5: **Convergence** – KNL outperforms Xeon
- Future work: adapt software to KNL
  - Use wide vectors
  - Use the on-chip memory
  - Multithreaded I/O and networking

Further Reading

- A. Uta et al., A Performance Study of Big Data Workloads in Cloud Datacenters with Network Performance Variability

- A. Uta et al., Exploring HPC and Big Data Convergence: a Graph Processing study on the Intel KNL