Meterstick: Benchmarking Performance Variability in Cloud and Self-hosted Minecraft-like Games

Jerrit Eickhoff

M.Sc. @ TU Delft, AtLarge Research

jerrit.eickhoff@gmail.com

https://atlarge-research.com/opencraft/

Ir. Jesse Donkervliet 

Prof. dr. ir. Alexandru Iosup 

@Large Research
Massivizing Computer Systems

Source and data available!
Meterstick: https://github.com/atlarge-research/Meterstick
Data: https://zenodo.org/record/7657838

ICPE 2023
Coimbra, Portugal

TU Delft VU Vrije Universiteit Amsterdam

Why Minecraft-like Games?

Massively popular:

- Video games are **the largest entertainment industry**
- Minecraft is the **best-selling video game of all time**
- More than **173 million people** play Minecraft per month
- Thriving industry of **third-party content creation**

YouTube > 1 Trillion views

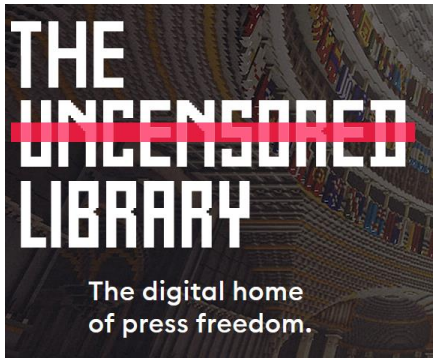
twitch > 2 Billion hours watched

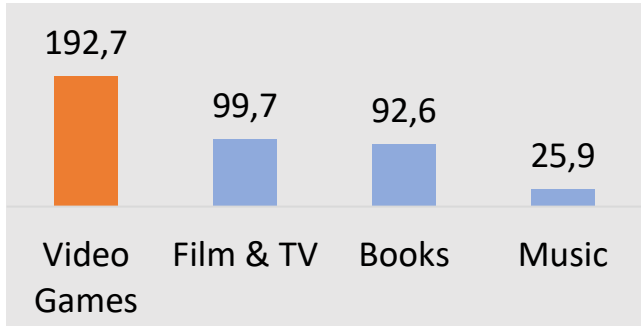
CURSEFORGE > 125 thousand mods

Societally beneficial:

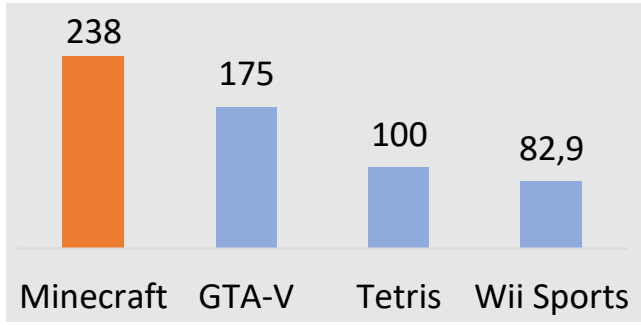
- *(not just)* Entertainment
- Education
- Activism
- Social Interaction

MINECRAFT EDUCATION





Global revenue of entertainment industry sectors 2021, in Billions USD



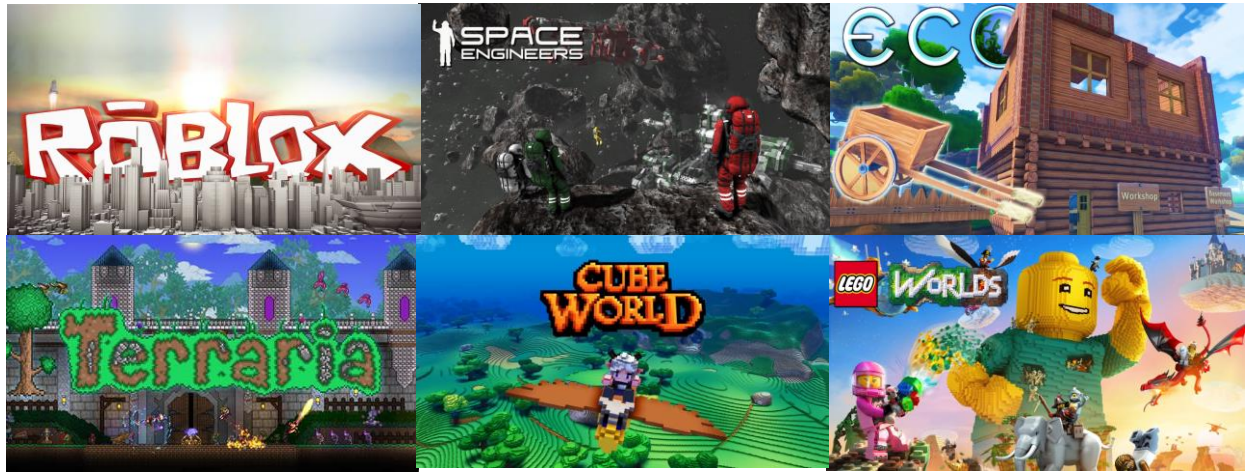
Total sales of highest selling video games, in Millions of copies sold

Minecraft: Connecting More Players Than Ever Before

by Helen Chiang, Studio Head, Mojang Studios • May 18, 2020 @ 6:00am

Sources: Newzoo, Statista, Motion Picture Association, IFPI, The Verge, Dot Esports, CurseForge, Xbox, Uncensored Library, Mojang

Minecraft-like Games



Not just Minecraft! Whole genre, characterized by:

- **Realtime interaction**
- **Dynamic, modifiable environments**
- **Server-client architecture**

Multiplayer services typically not operated by the developer, but instead **community-hosted**.

ServerMiner
GGSERVERS
BISECT HOSTING
ScalaCube
RAMSHARD
STICKY PISTON
MINECRAFT HOSTING
Ferox Hosting
SHOCKBYTE
CubedHost
PebbleHost
APEX HOSTING
Server.pro
NODECRAFT
HOSTHAVOC
skynode
BECAUSE YOU DON'T HAVE TO PLAY ALONE
MCPROHOSTING
MELONCUBE
Aquatis

The booming market of premium Minecraft-like Game cloud services

The Problem

Massively popular == incredibly scalable, right?

Minecraft music festival Block By Blockwest postponed after servers crash

Over 100,000 people logged on to catch virtual performances by Massive Attack and more

By **Patrick Clarke** | 26th April 2020

Isolated instances which do not scale beyond **a few hundred players.**[1]

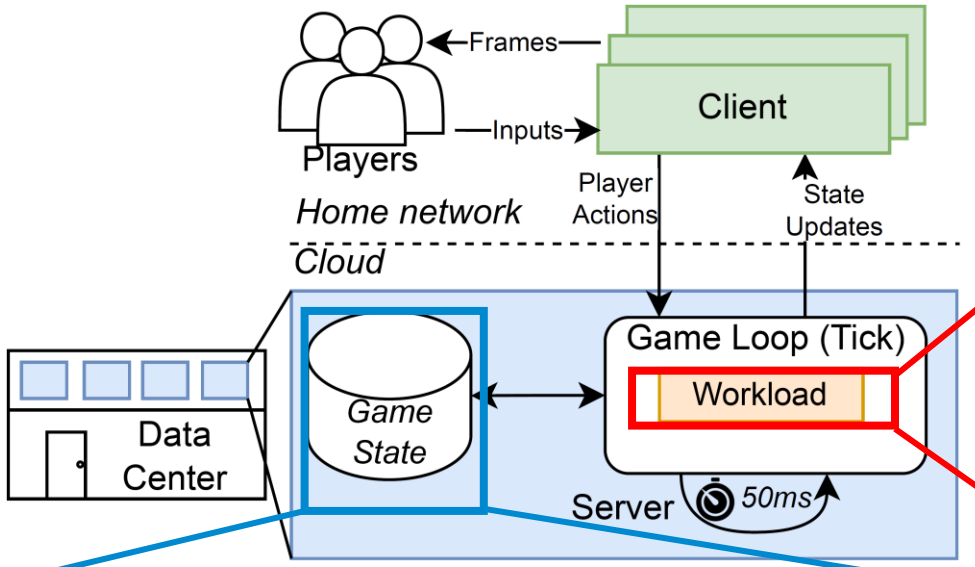
Does not account for performance impact of **cloud-hosting** or **environment-based workloads**.

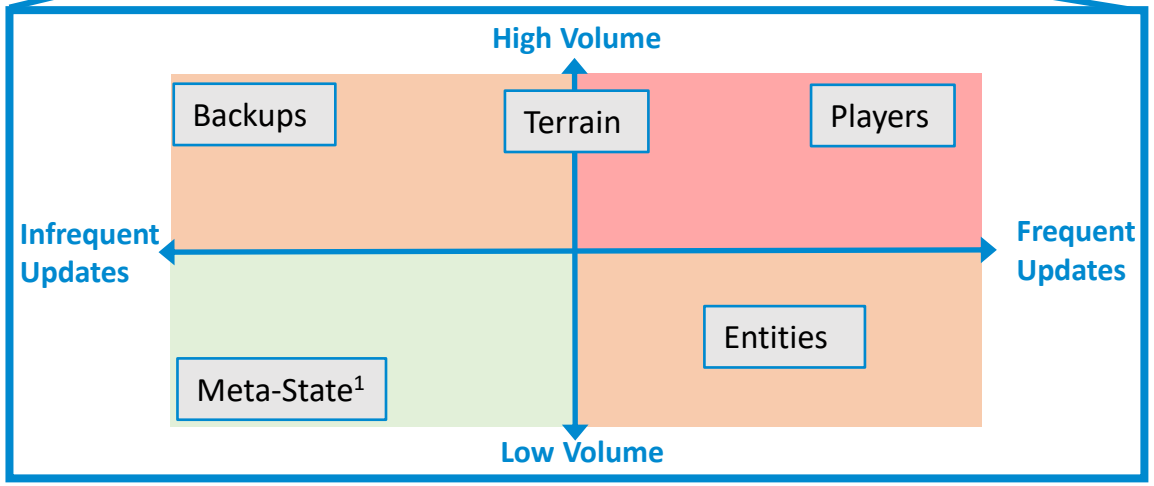
A *single player* can overload or crash Minecraft-like games!

Sources:
1: Yardstick: A benchmark for minecraft-like services, Jerom van der Sar et al. ICPE2019.

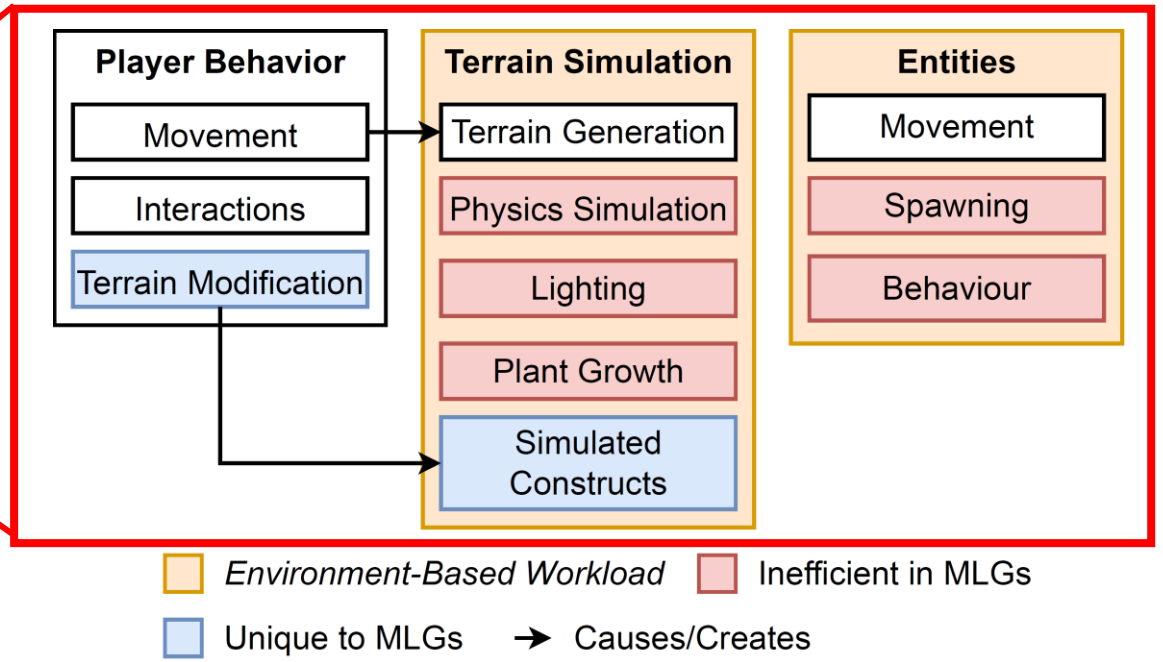
A single player crashing the game!? How can this be?

Server-Client Architecture





Minecraft-like Game Workload Model

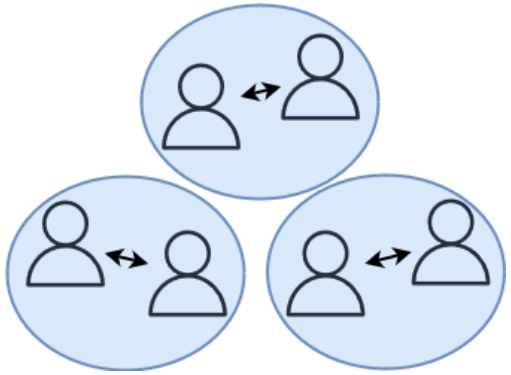


Assumed Game State Volume and Update Frequency

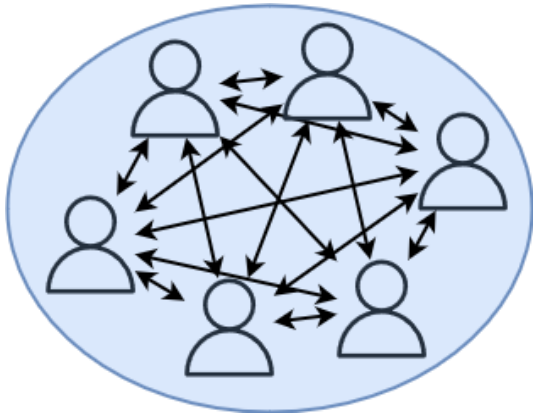
[1]: State concerning functional operation of the game server rather than game features, such as administrative logs or user authentication tokens.

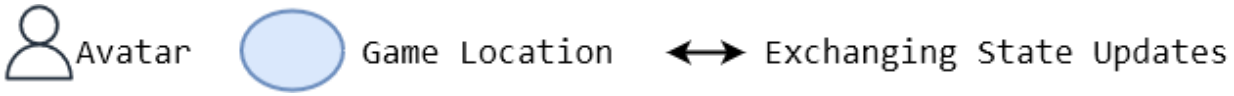
Player Workload

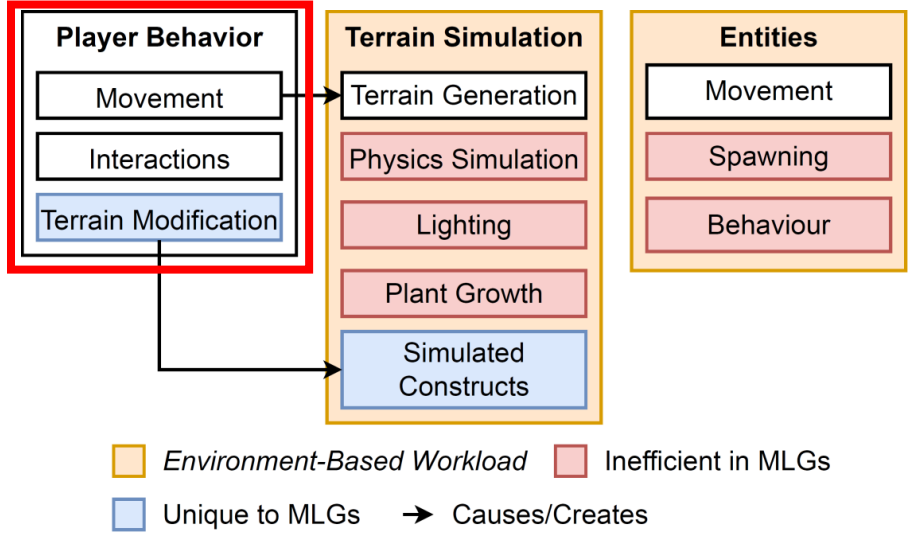
Player Avatars Sparse



Players Avatars Dense







Eve Online 13,700 player battle causes performance disruptions

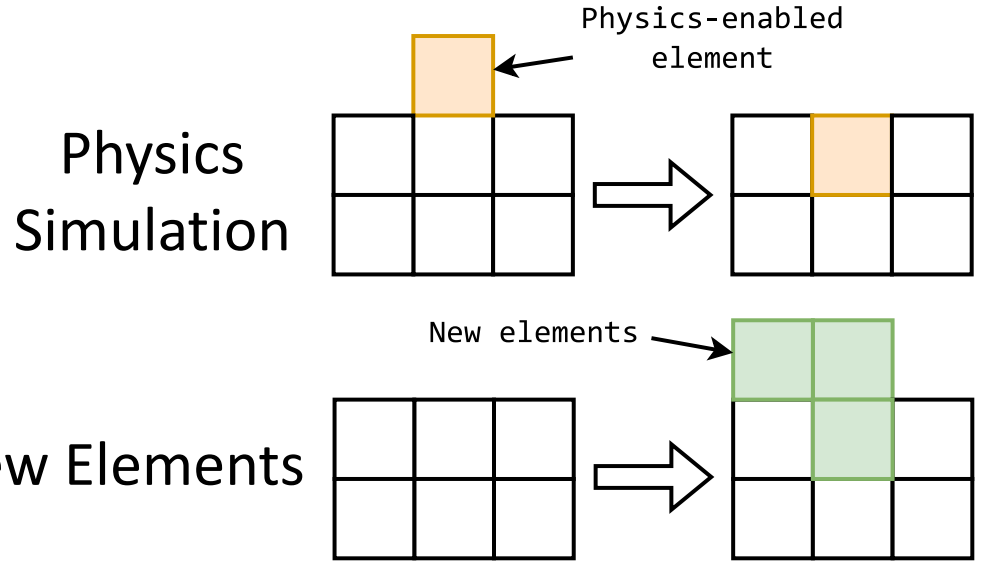
Sources: Polygon

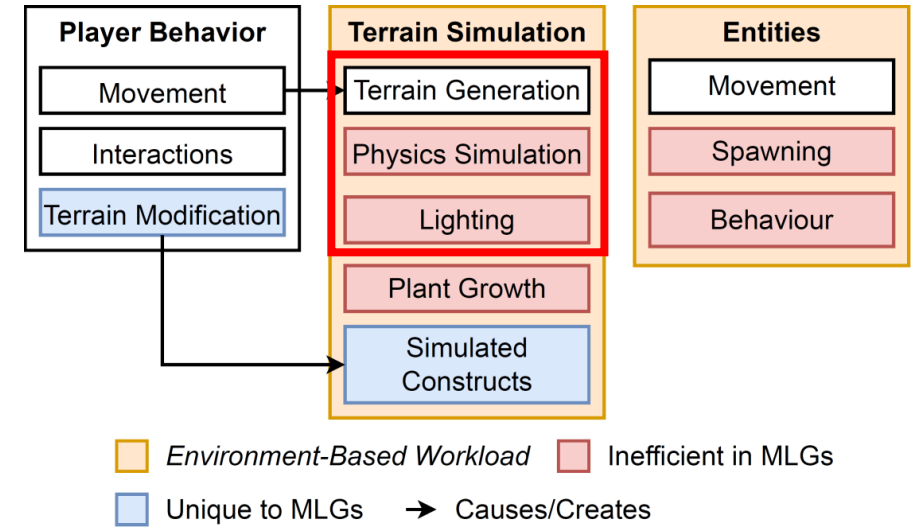
Environment-Based Workloads

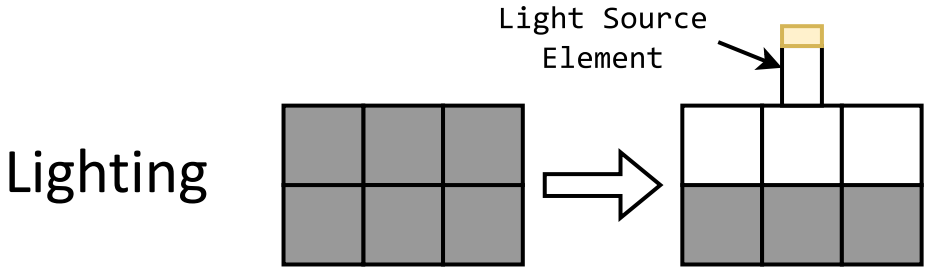
Environment comprised of modifiable *elements,* each with unique properties and individual state

Dynamic, modifiable properties make environment workload in Minecraft-like Games **Inefficient**:

- Require state of neighboring elements
- Recalculation on updated state







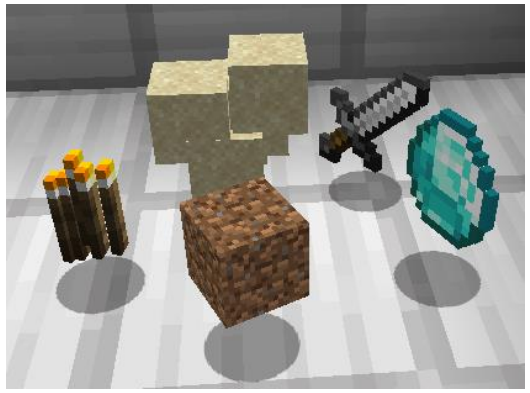
Environment-Based Workloads

Entities

- Exists in environment, but is **not player or terrain**

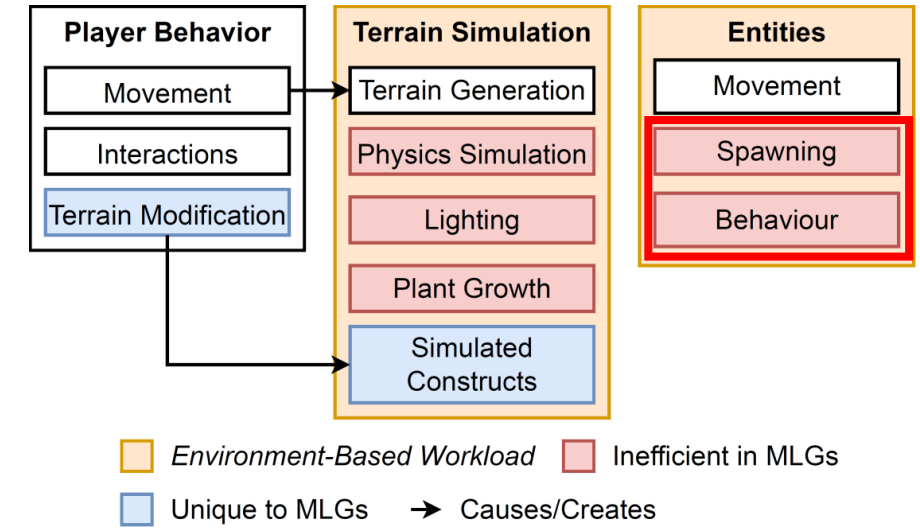
Hostile Mob

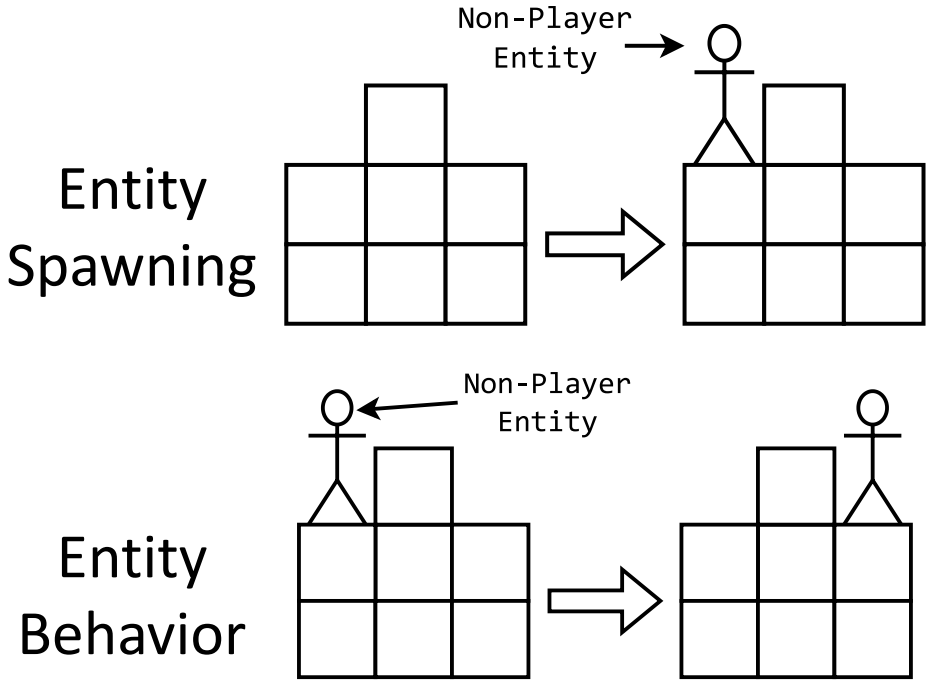
Passive Mob



Item Entities

Cannot be precomputed!

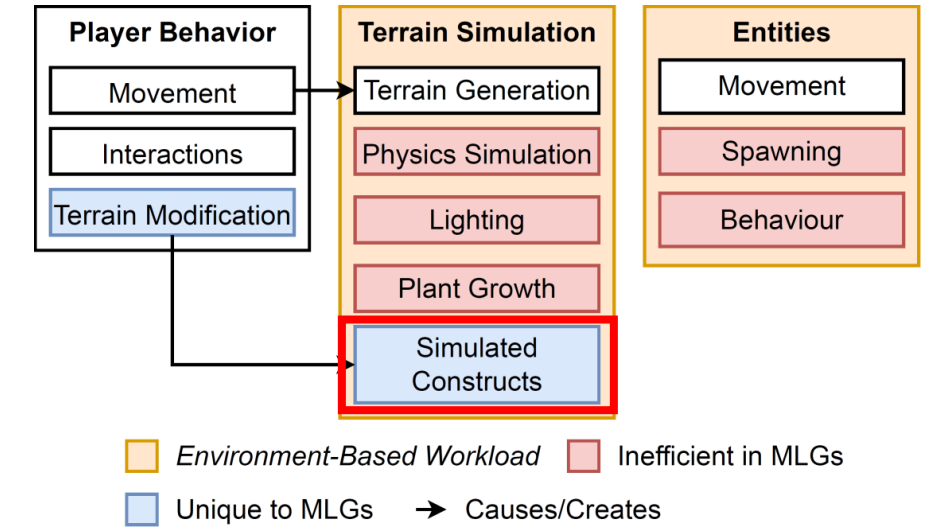


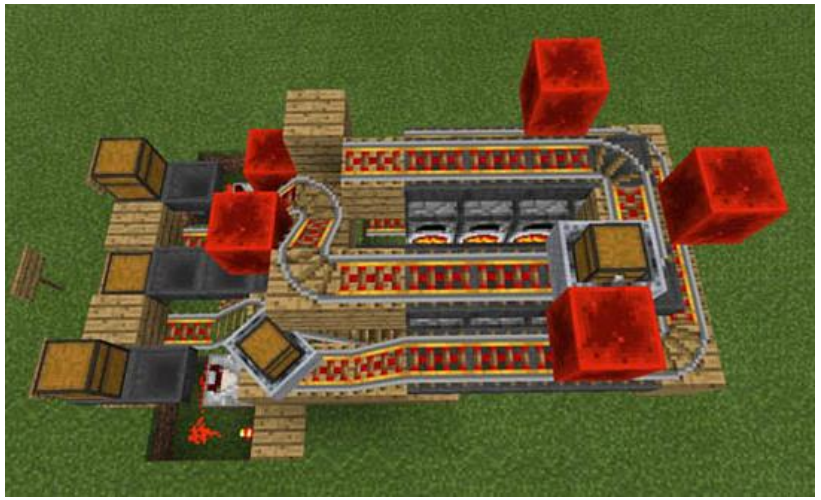


Environment-Based Workloads

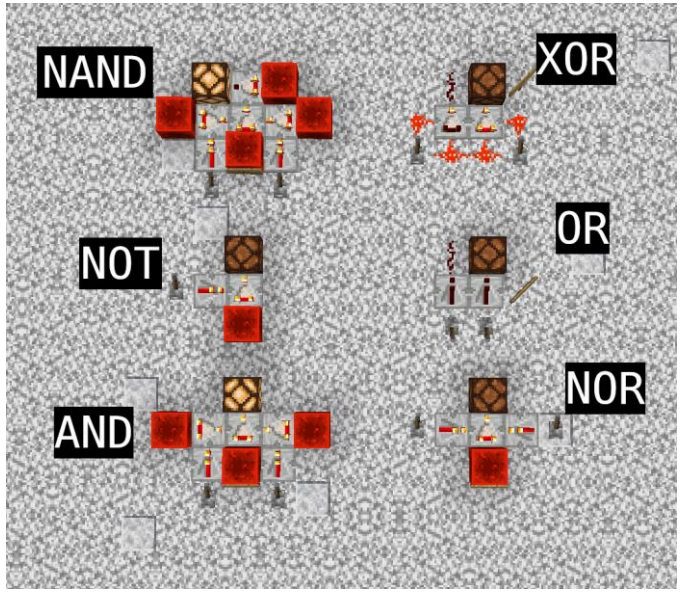
Simulated Constructs

- **Player-constructed** structures consisting of dynamic elements
- "Programmed" to **automatically perform** some in-game task

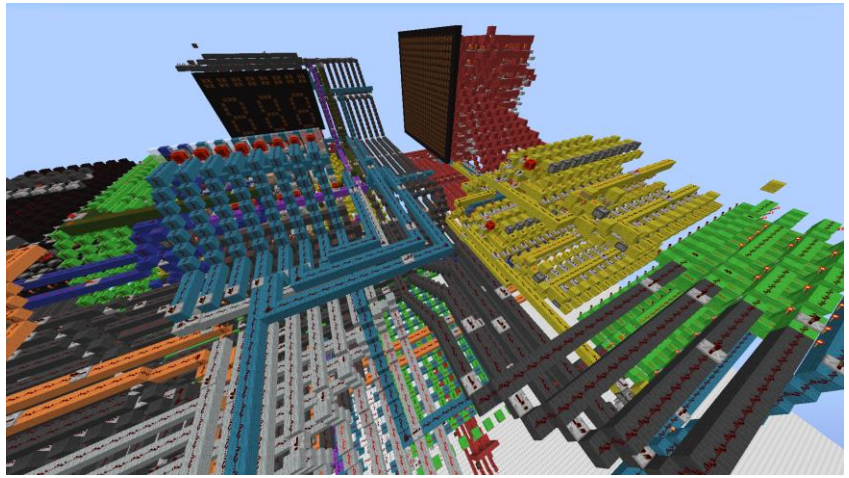




Automatic resource processing



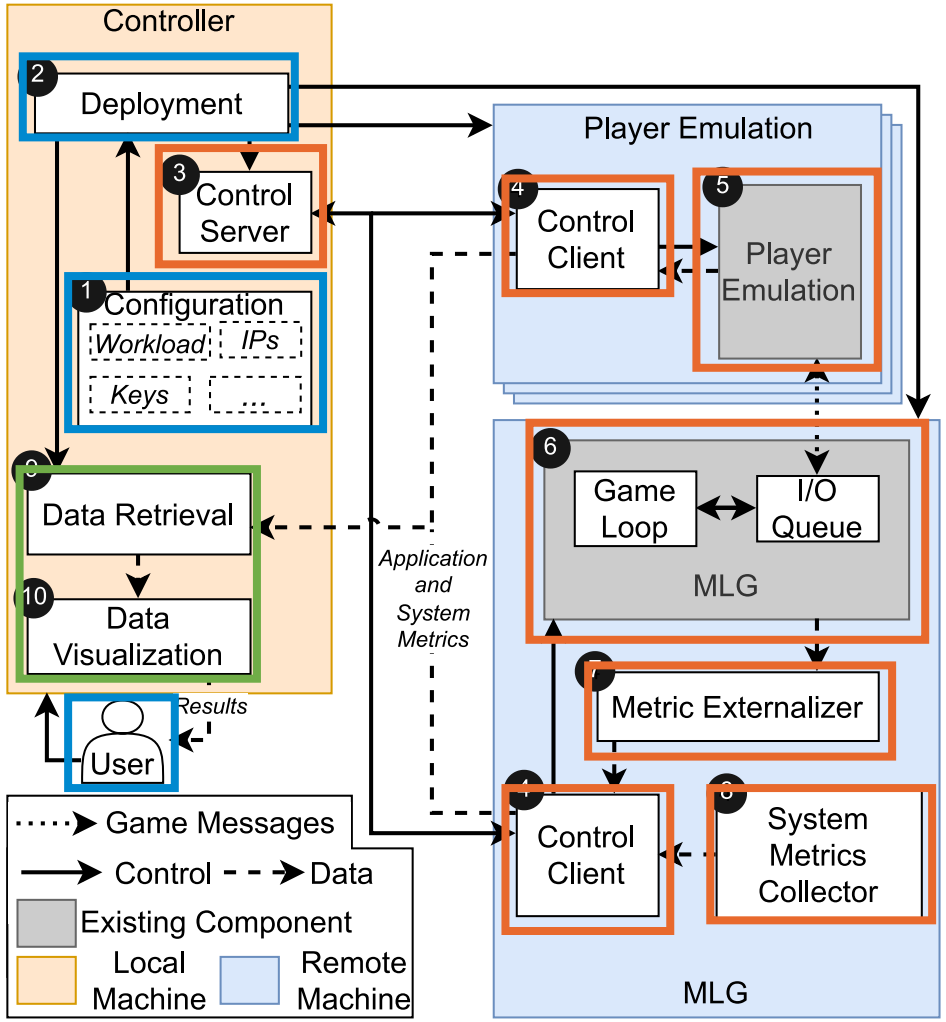
Logic Gates



Operational 16-bit, 1Hz computer

Sources: Reddit, Reddit

Meterstick Benchmark: Design

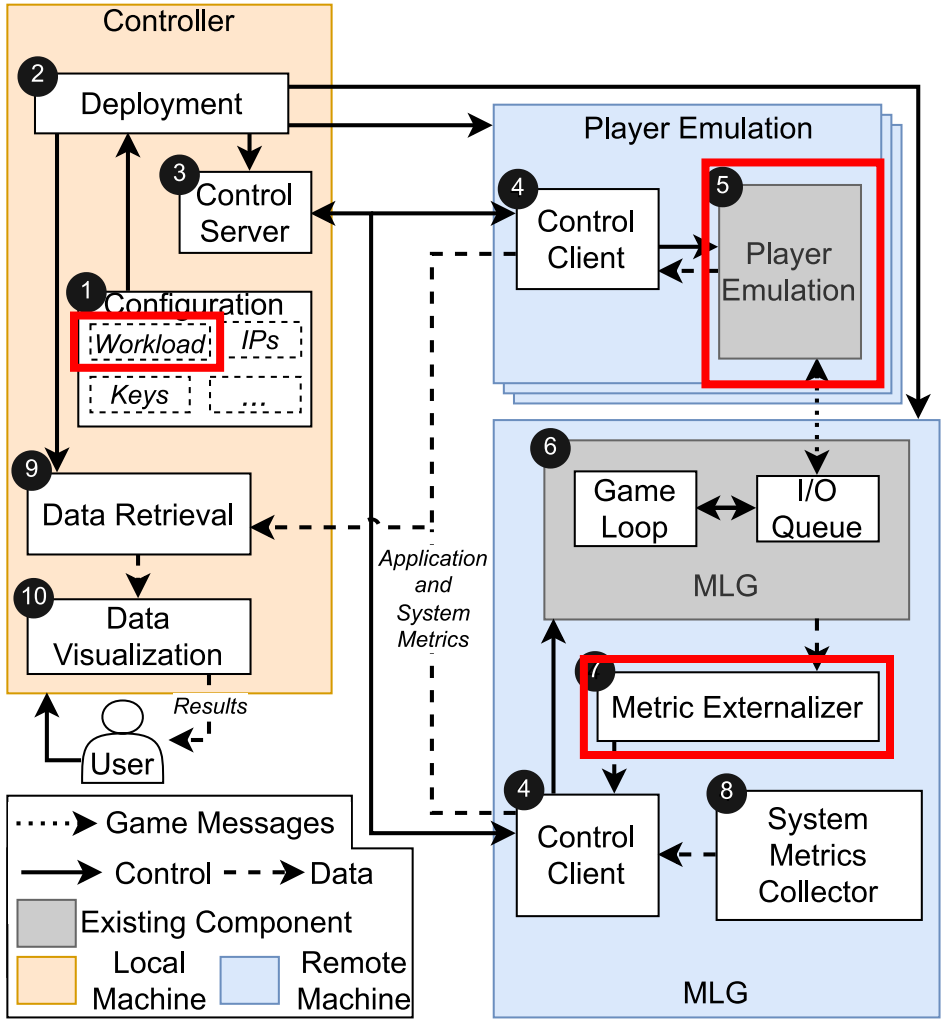


- Supports **environment-based workloads**
- Uses **player-emulation** for player contribution to workload
- Deploys Minecraft-like Games experiments on **commercial clouds**
- Collects relevant **application and system metrics**

Steps:

1. Deployment
2. Experiments
3. Data retrieval

Meterstick Benchmark: Design



- Workloads, Player Emulation, and Metric Externalization tied, directly or indirectly, to application protocol
- Currently supports Minecraft-like games utilizing the **Minecraft protocol**

Minecraft server implementations

Popular[1] mod packs

Same server technology for different games!

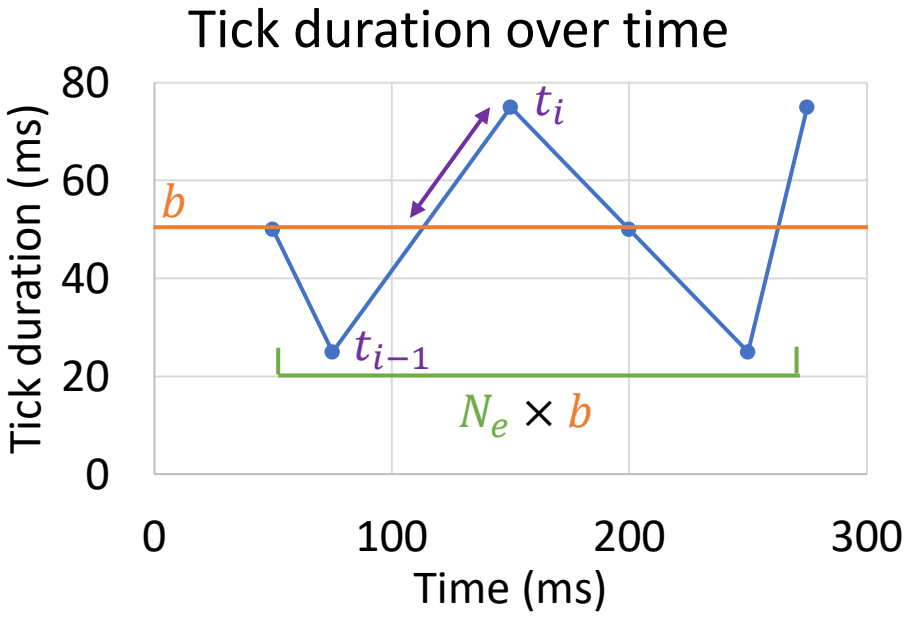
[1]: Ranges from 150 thousand to 2 million downloads, with some individual mods reaching 223 million downloads. See TechicPack and CurseForge

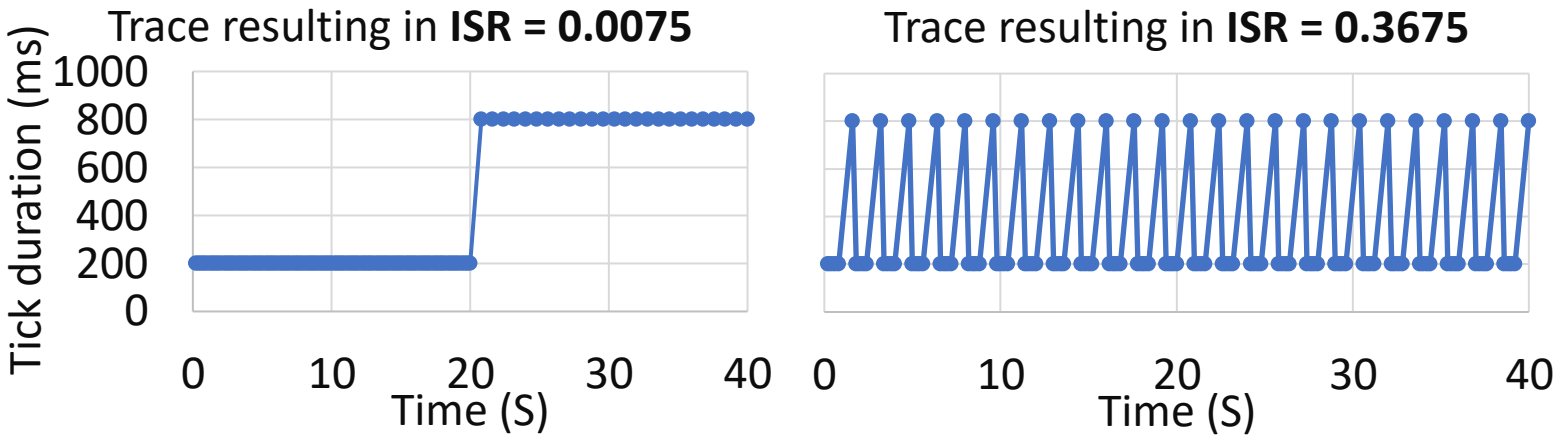
Instability Ratio (ISR)

$$ISR = \frac{\sum_{i=1}^{N_a} |\max(b, t_i) - \max(b, t_{i-1})|}{N_e \times 2b}$$

b = minimum delay between ticks
t_i = duration of i^{th} tick
N_a = actual number of ticks
N_e = expected number of ticks

- **Stability > lowest latency** for online gaming [1-3]
- **Normalized** measure of instability given a trace of tick durations, based on cycle-to-cycle jitter.
- **Order dependent**





Same tick durations, different order

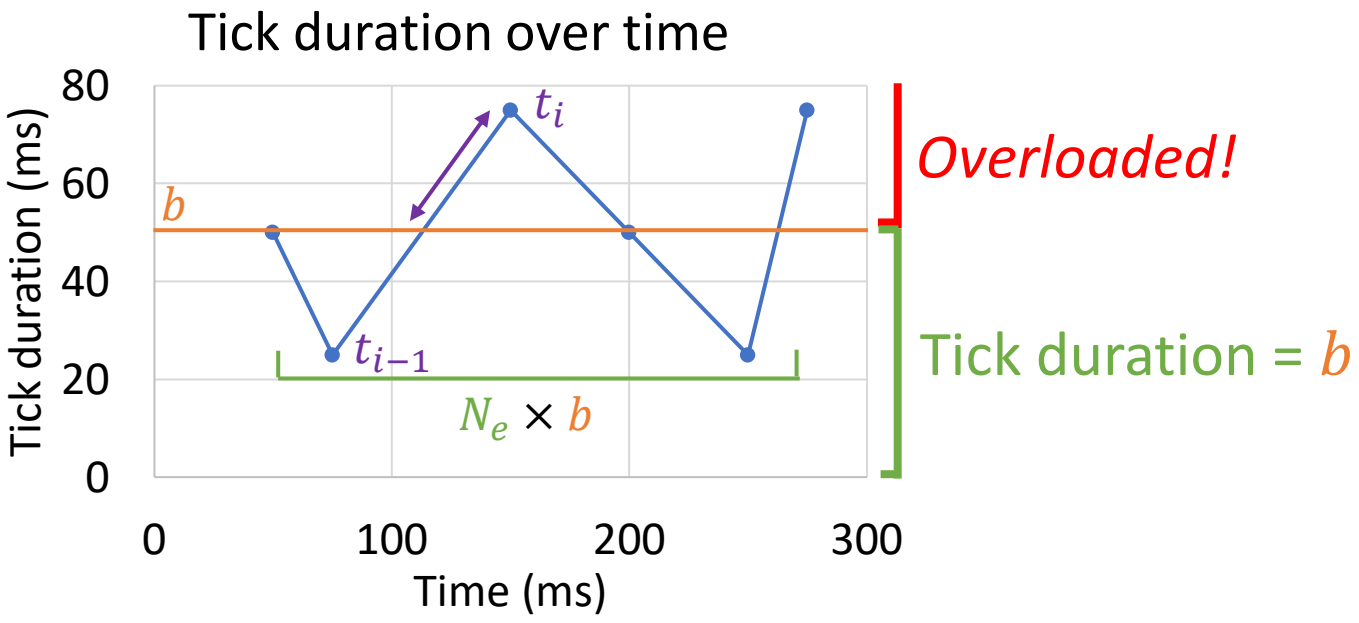
Sources:
1: How sensitive are online gamers to network quality? Chen et al. Commun. ACM 49, 11 (2006)
2: Player Perception of Delays and Jitter in Character Responsiveness, Normoyle et al. SAP2014
3: Empirical study of subjective quality for Massive Multiplayer Games, Ries & Rupp, IEEE (2008)

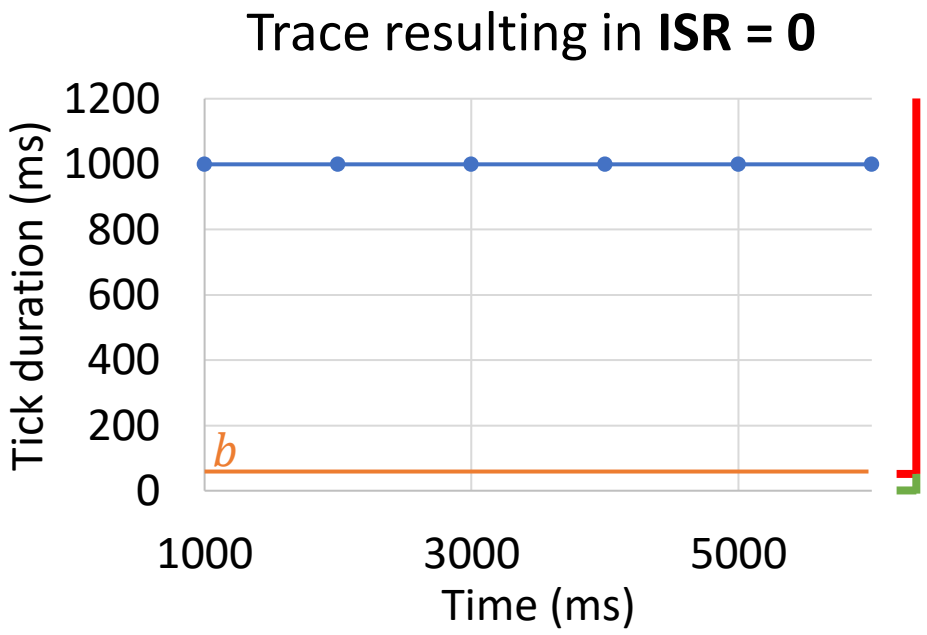
Instability Ratio (ISR)

$$\boldsymbol{ISR} = \frac{\sum_{i=1}^{N_a} |\max(b, t_i) - \max(b, t_{i-1})|}{N_e \times 2b}$$

- b = minimum delay between ticks
- t_i = duration of i^{th} tick
- N_a = actual number of ticks
- N_e = expected number of ticks

- $\boldsymbol{ISR}$ = 0 if all ticks below b!
- $\boldsymbol{ISR}$ = 0 if all ticks are **the same!**
- **Not meant to be used as standalone performance metric!**





Experiment - Setup

Minecraft-like Games

Minecraft
Default

Forge
Mods

PaperMC
Performance

Environments

Amazon Web Services

Azure

DAS5 Cluster

2vCPU: AWS: *T3.Large,* Azure*: Standard_D2_v3*

Workloads:

Workload Name	Description
Control*	Freshly generated world
TNT*	Fast entity actions, terrain updates
Farm*	Many simulated constructs
Lag*	Simulated construct stress test
Players	25 moving players in small area

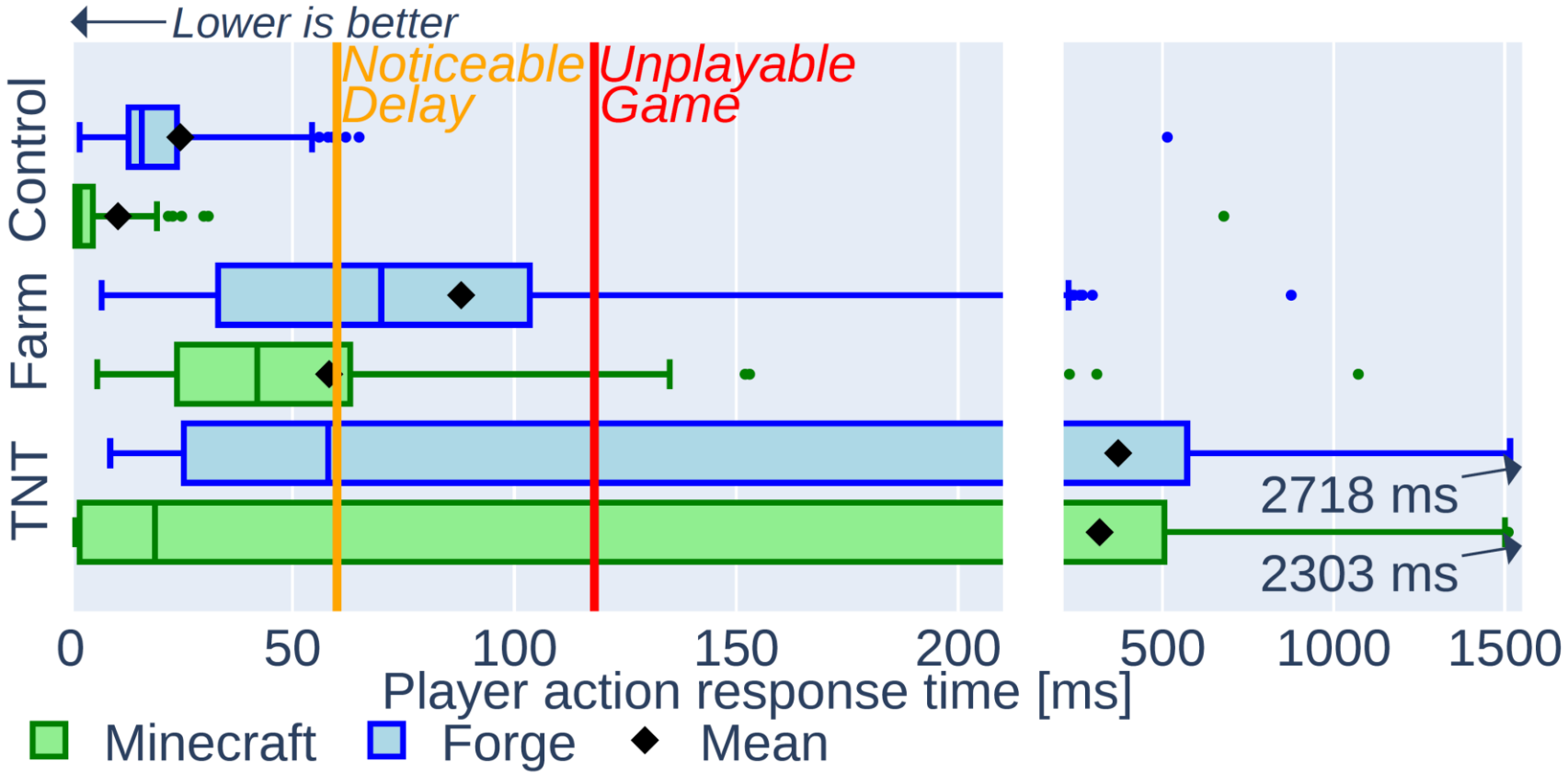
**Only one player, stationary*

Hardware Guidelines:

Service	vCPU[#]	CPU Speed [GHz]
Server.pro	2	2.4
Skynode	2	3.6
Hostinger	3	NP
Ferox Hosting	Not reported	Not reported
MelonCube	Not reported	3.4
Azure	2	Variable
AWS	1	Variable

Full list of cloud service Minecraft-like game hosting recommendations, and community simulated constructs, available in technical report: https://arxiv.org/abs/2112.06963

Environment-based workloads cause significant performance instability



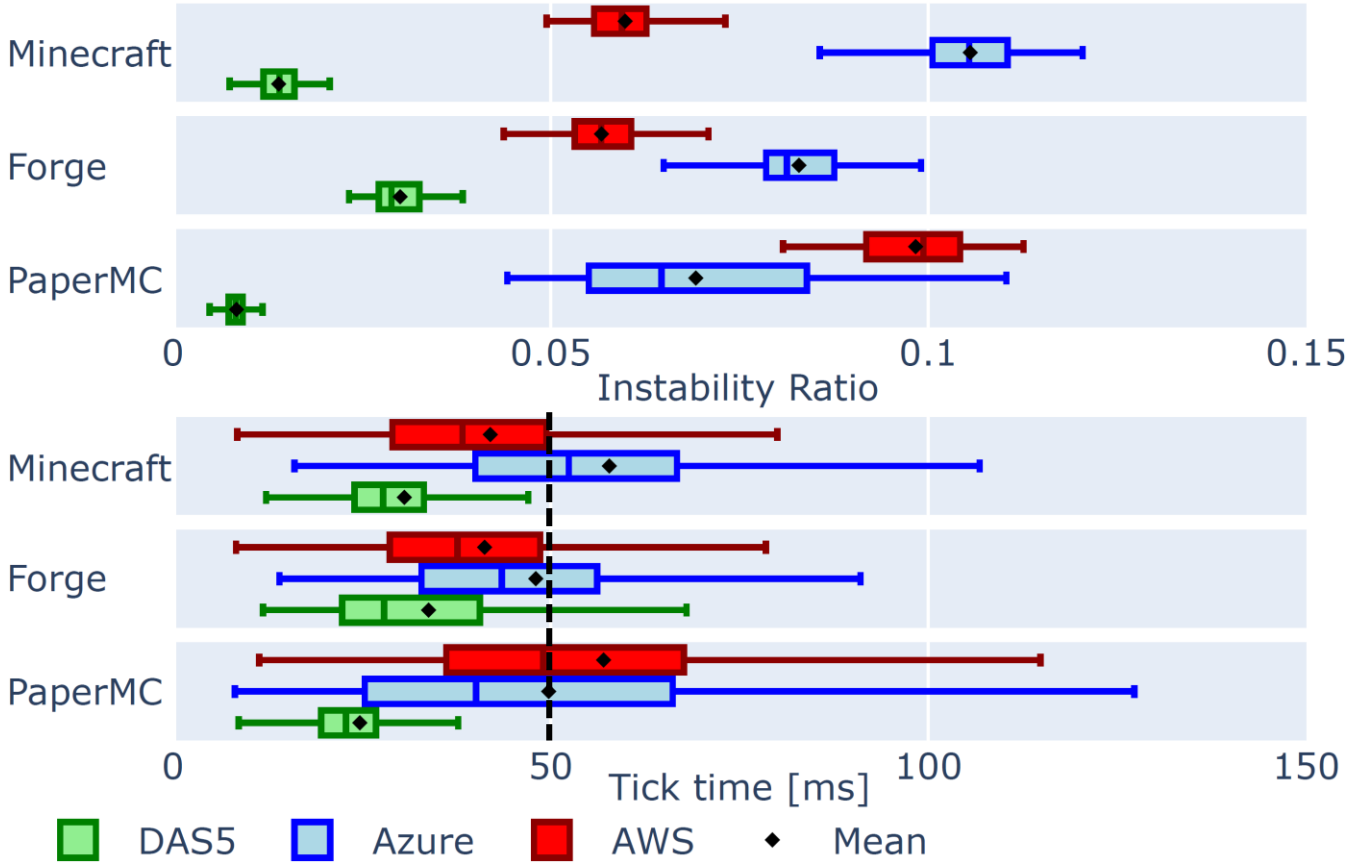
Lag workload missing, crashed!

Player action response time on AWS

Whiskers to 5th, 95th percentiles

Sources for Noticeable, Unplayable thresholds:
1: Analysis of factors affecting players' performance and perception in multiplayer games, Dick et al. Netgames 2005
2: Are 100 ms Fast Enough? Characterizing Latency Perception Thresholds in Mouse-Based Interaction, Forch et al. EPCE 2017

Cloud environments cause significant performance variability



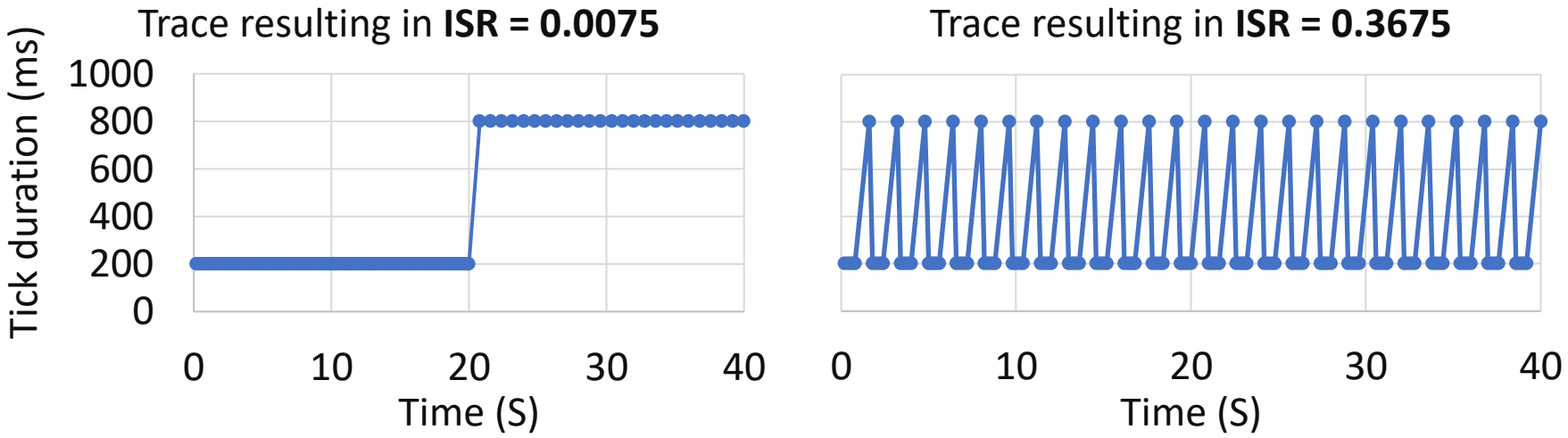
Variation of **Instability Ratio** and **Tick time** over 50 iterations of Players workload

Whiskers to 1.5 x IQR

Actionable Insights

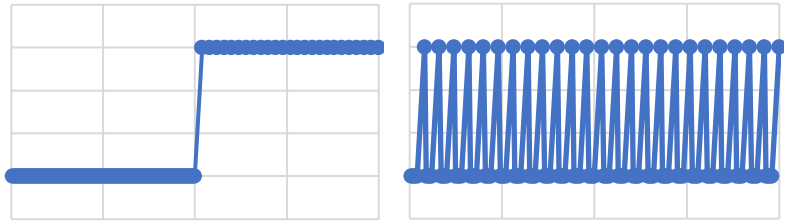
Researchers:

- Performance analysis of online games should include stability analysis! Common statistical measures can hide performance problems



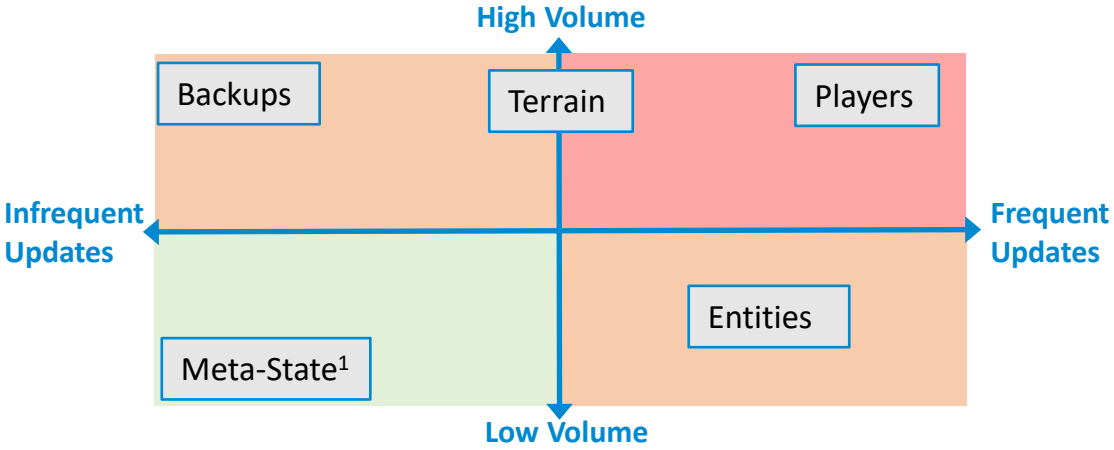
Same mean, median, deviation, quantiles, etc., but vastly different instability!

Actionable Insights

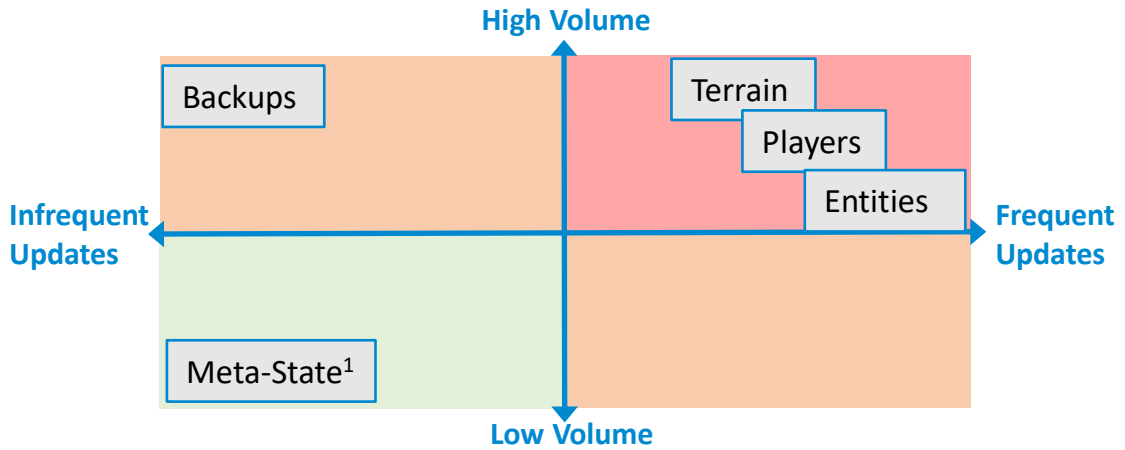


Researchers:
- Performance analysis of online games should include stability analysis! Common statistical measures can hide performance problems

Game Developers:
- Environment-based workloads severely impact scalability



Assumed Game State Volume and Update Frequency

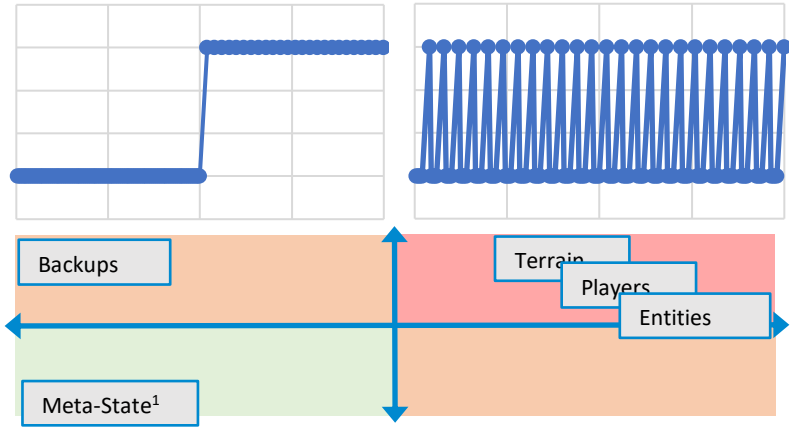


Revised Game State Volume and Update Frequency

Actionable Insights

Researchers:
- Performance analysis of online games should include stability analysis! Common statistical measures can hide performance problems

Game Developers:
- Environment based workloads even more of a scalability concern in Minecraft-like games than previously thought
- Situation improvable by performance engineering, much to be done



Case Study: PaperMC

Simulation quality vs. performance tradeoff

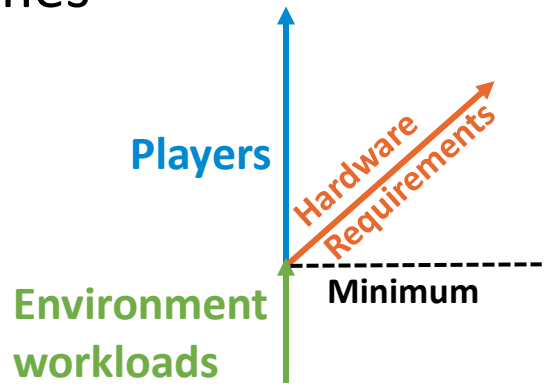
Asynchronous threading + heuristics

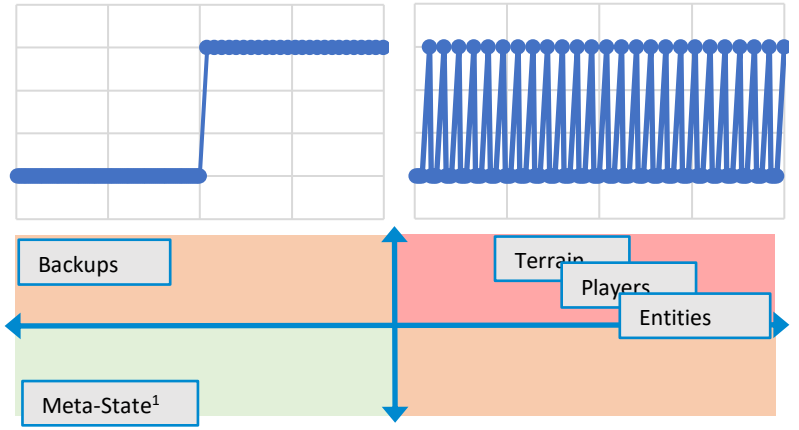
Actionable Insights

Researchers:
- Performance analysis of online games should include stability analysis! Common statistical measures can hide performance problems

Game Developers:
- Environment based workloads even more of a scalability concern in Minecraft-like games than previously thought
- Situation improvable by performance engineering, much to be done

Cloud Providers:
- Revise hardware recommendations for Minecraft-like games





Simulation quality vs. performance tradeoff

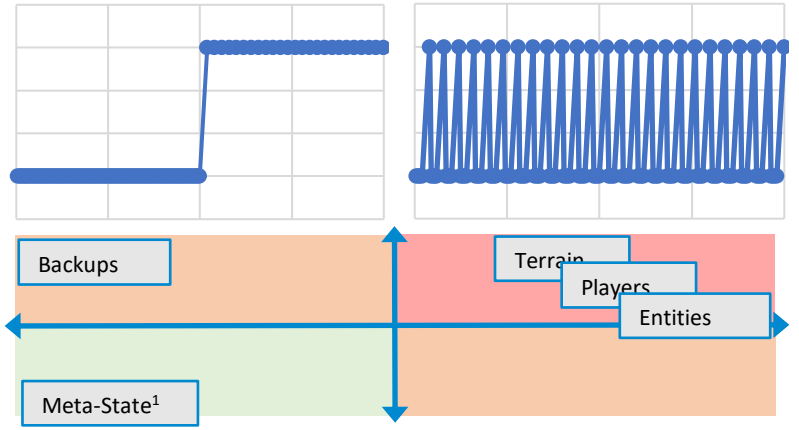
Actionable Insights

Researchers:
- Performance analysis of online games should include stability analysis! Common statistical measures can hide performance problems

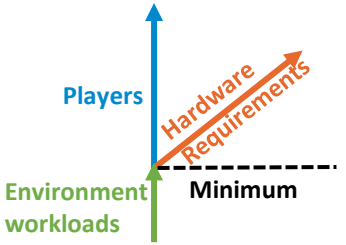
Game Developers:
- Environment based workloads even more of a scalability concern in Minecraft-like games than previously thought
- Situation improvable by performance engineering, much to be done

Cloud Providers:
- Revise hardware recommendations for Minecraft-like games

Server Hosts:
- Compare cloud providers for your Minecraft-like game, consider self hosting



Simulation quality vs. performance tradeoff



Future Work

- User studies to directly link our **Instability Ratio (ISR)** values to player-perceived quality of experience
- Public **leaderboard of Meterstick scores**, allow players, game designers, and cloud platforms to compare results!

Source and Data Available!

Meterstick:

https://github.com/atlarge-research/Meterstick

Data:

https://zenodo.org/record/7657838

Selected **Opencraft** Articles

Serverless gaming

Servo: A Use-Case for Serverless Computing in Online Gaming
Jesse Donkervliet, Javier Ron, Junyan Li, Tiberiu Iancu, Cristina L. Abad and Alexandru Iosup. **ICDCS 2023**

Dynamic **consistency**

Dyconits: Scaling Minecraft-like Services through Dynamically Managed Inconsistency
Jesse Donkervliet, Jim Cuijpers, and Alexandru Iosup. **ICDCS 2021**

Benchmarking online games

Meterstick: Benchmarking Performance Variability in Cloud and Self-hosted Minecraft-like Games
Jerrit Eickhoff, Jesse Donkervliet, and Alexandru Iosup. **ICPE 2023**

Yardstick: A Benchmark for Minecraft-like Services
Jerom van der Sar, Jesse Donkervliet, and Alexandru Iosup. **ICPE 2019**

Jesse Donkervliet
Tech Lead

Jerrit Eickhoff
MSc student

Alexandru Iosup
Project Lead