Taming Performance Variability

Aleksander Maricq* Dmitry Duplyakin* Ivo Jimenez†
Carlos Maltzahn† Ryan Stutsman* Robert Ricci*

* University of Utah
† University of California Santa Cruz

Outline

Work published at OSDI'18

Current Efforts

Future Directions

Cyber-Physical Systems/Internet of Things

- Original context: Performance metrics on bare-metal compute HW
- Analysis techniques are not specific to this context
- Applicable to environments with more and less control over factors

Taming Performance Variability - OSDI'18

Motivation: Performance Variability

How confident should I be that my results are correct?

How many times do I need to run my experiments?

CloudLab

As a testbed builder, how can I help users figure this out?

11 months
~892,000 data points
835 servers

Memory
Disk
Network

Examine performance variability of testbed hardware

Within servers
Across servers

CloudLab

https://www.cloudlab.us/

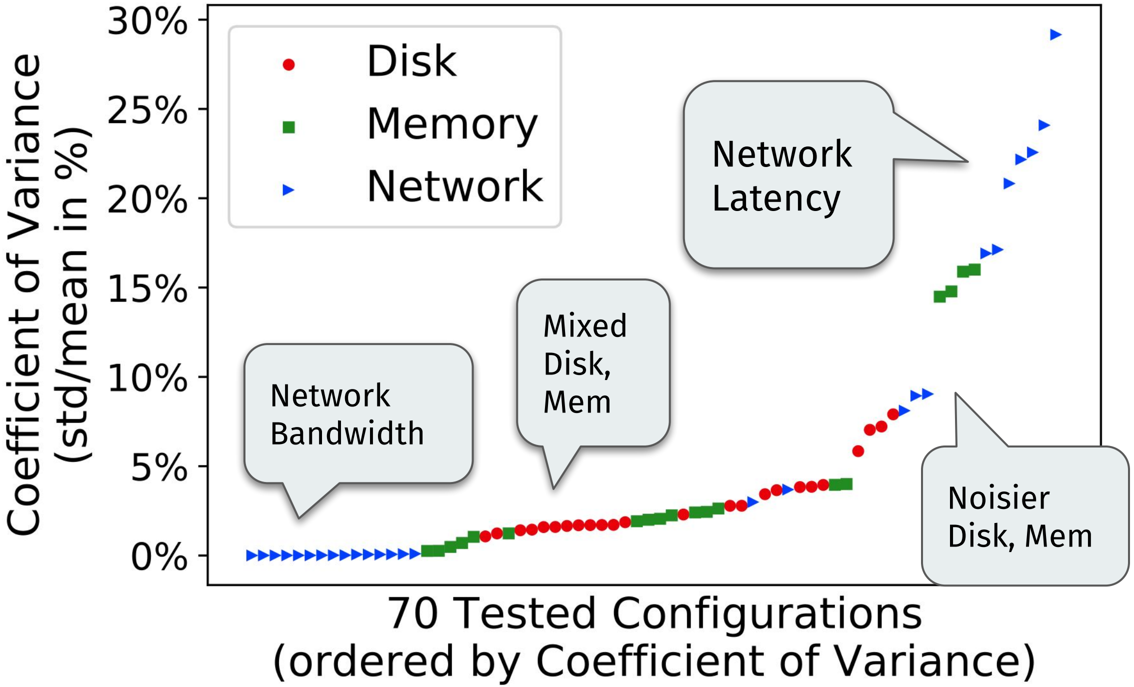
- 1,500 servers at three sites
  - Several distinct ‘types’ of identical servers
- Exclusive, raw access to hardware
  - No interference on servers from simultaneous users
  - Doesn’t add virtualization overhead / variability
- Our experiments were run on servers allocated only to us
- Configuration: Combination of hardware type, workload, parameters

c220g1, single-threaded mem copy, dvfs off

m510, net bw, rack-local

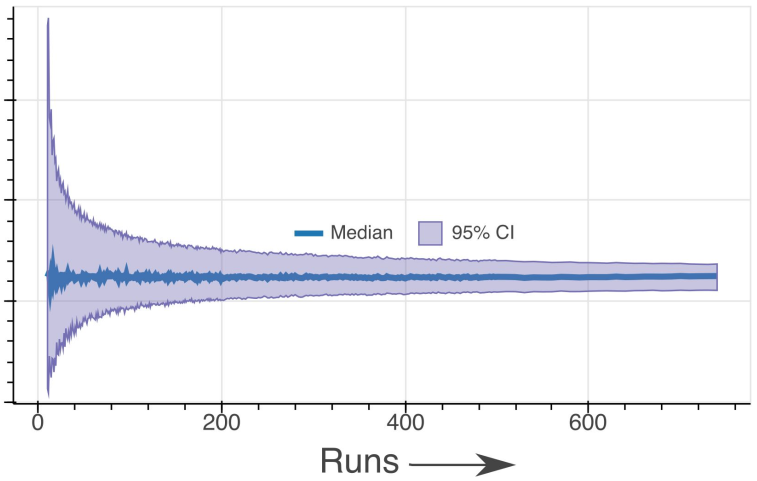
How confident can we be in the correctness of our results?

How much trouble are we in?



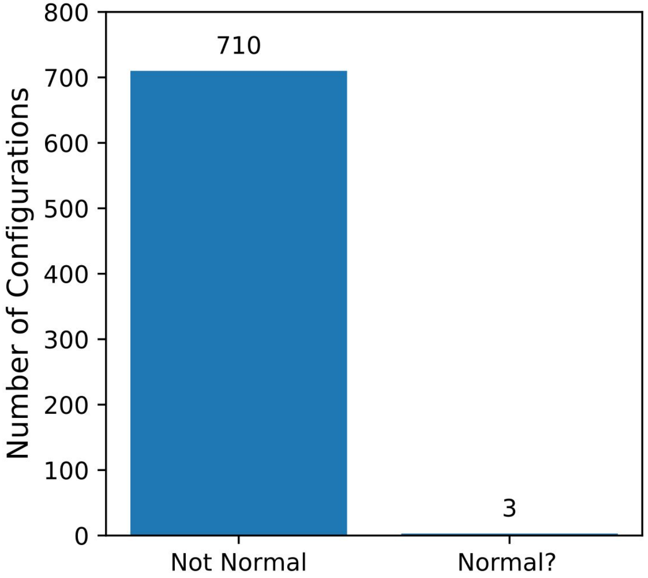
Confidence Intervals

- Range for your mean (different than stdev)
- Represents some % confidence (eg. 95%) the true mean lies between
- More runs -> narrower CI



Testing Normality

- Many statistical models assume normal (Gaussian) bell-curve
- Is our data normal? Shapiro-Wilk test (95% confidence)



Use Non-Parametric Statistics to Avoid Assumptions of Normality

How confident can we be in the correctness of our results?

- Some variation is unavoidable
- Results are often non-normal
- More runs → more confidence

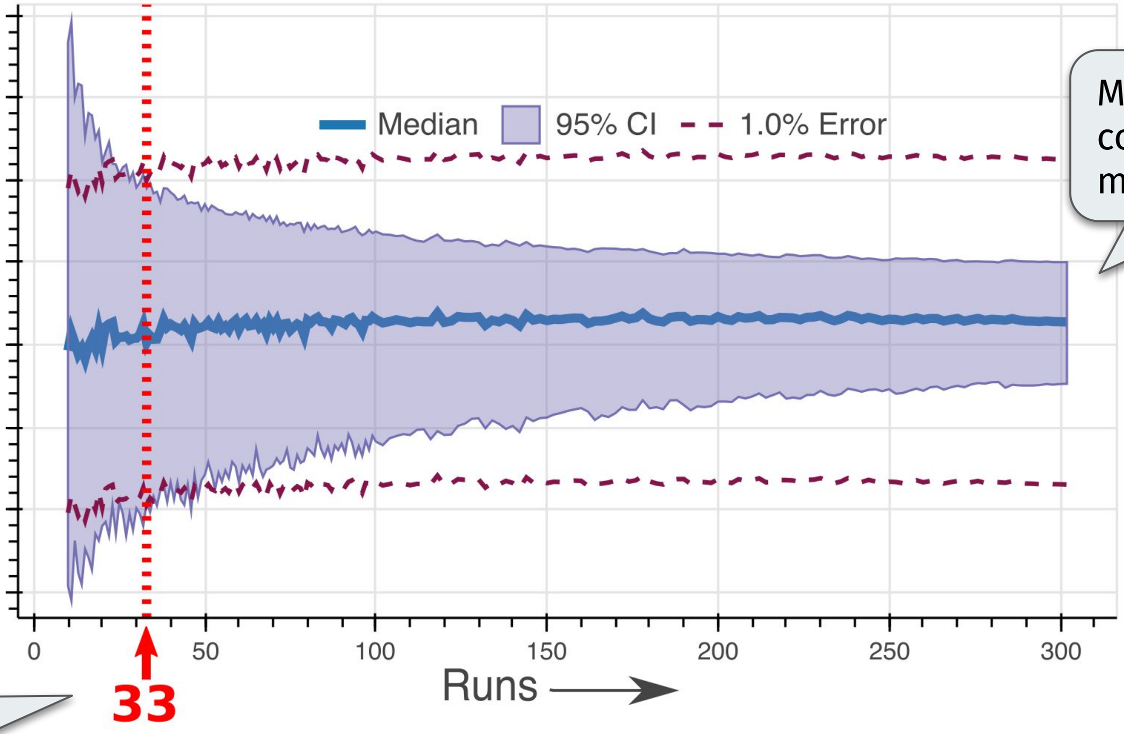
How many times
should we run our experiments?

CONFIRM - CONFIdence-based Repetition Meter

- Uses all our collected data to build *estimates* of how many runs are needed
  - For configurations on a single server or group of servers
- Uses random sub-samples of historical data
  - Takes many sub-samples, computes mean and CI
- Calculating observed empirical CIs still necessary
- Integrated into CloudLab, but doesn't have to be specific to it

CONFIRM

From past data, uses random subsets to model median and CI behavior for increasing numbers of runs



Median and CI converge with more runs

33 runs until CI is within 1% of median

CONFIRM Recommendations

	CoV	Recommended Runs
Mem Config A (c8220, ST copy, no dvfs, socket 1)	0.262	
Disk Config B (c8220, /dev/sda4, seqwrite, iodepth 4096)	1.708	
Mem Config C (c220g1, ST copy, dvfs, socket 1)	6.139	
Net Config D (m400, not rack-local, iperf3 (bw), forward)	6.309	
Net Config E (m510, not rack-local, latency, forward)	8.086	
Disk Config F (c8220, /dev/sda4, randread, iodepth 4096)	8.122	

Trend: Higher CoV → More Runs

CoV and recommended runs are not perfectly correlated

Recommended runs rise fast with higher CoV

How many times should we run our experiments?

- Enough for target confidence
- Trend: high CoV → more runs
- Use past data to estimate

Can the facility help?

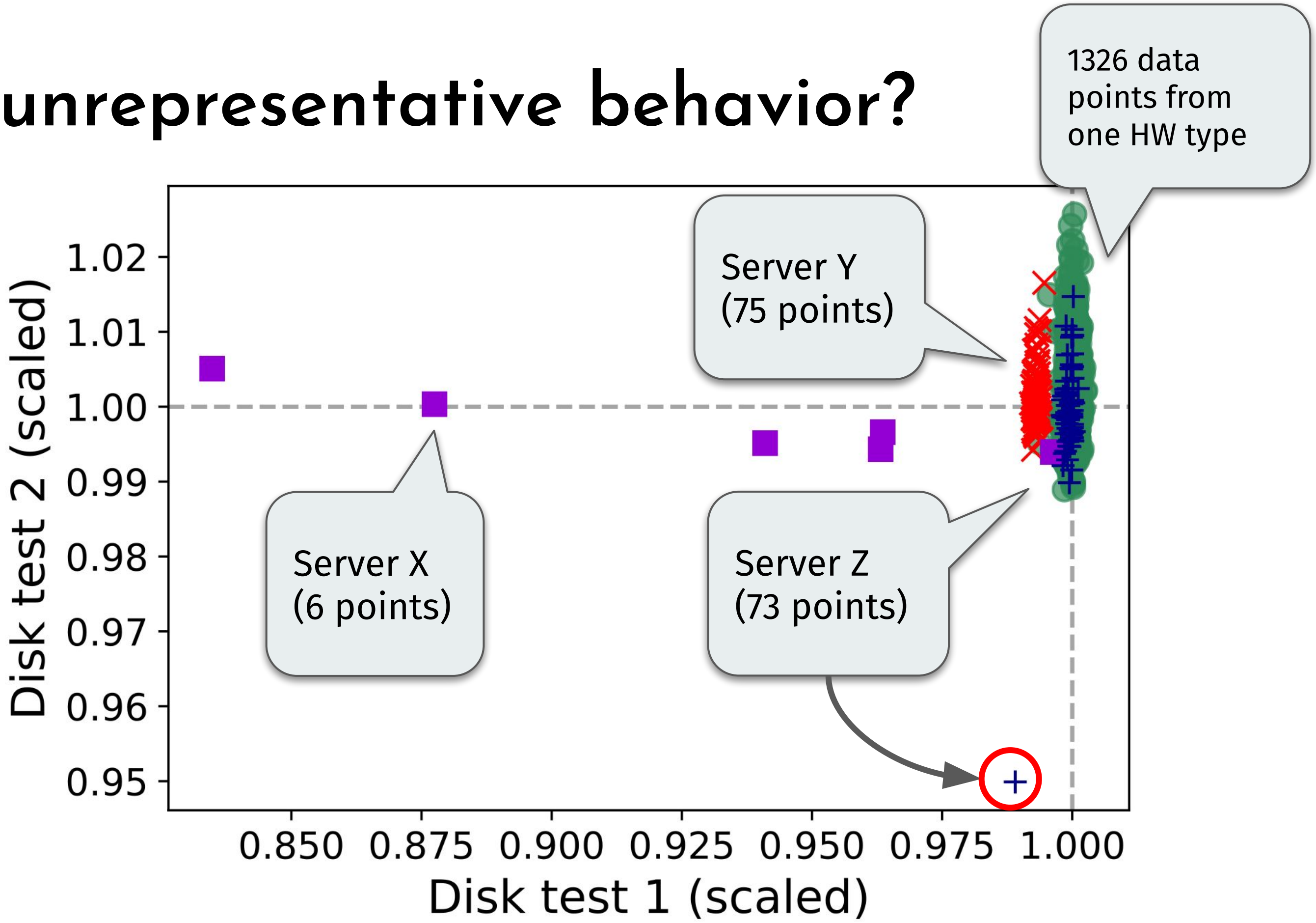
Can The Facility Help?

- Provide indistinguishable resources

Indistinguishable:

Performance results gathered on *any* server should be representative of the *population as a whole*.

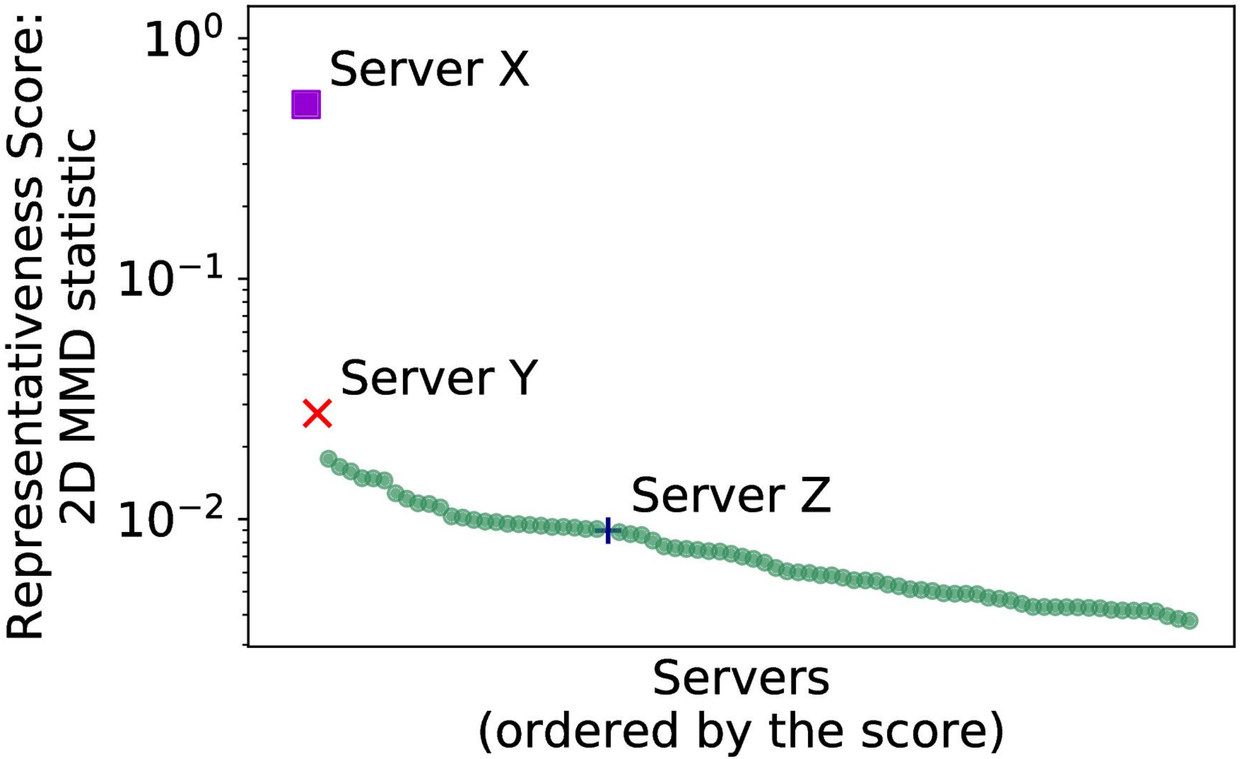
What is unrepresentative behavior?



Detecting Unrepresentative Resources

- Kernel two-sample test based on Maximum Mean Discrepancy (MMD)
    - Provides a measure of similarity between two non-parametric distributions
- We compare:
    - Each server to all others of its type
    - … using many dimensions: disk, memory, and network
- Remove servers that are statistically dissimilar from the rest

Removing Unrepresentative Servers



Can The Facility Help?

- Identify and/or fix anomalous components

Related Work

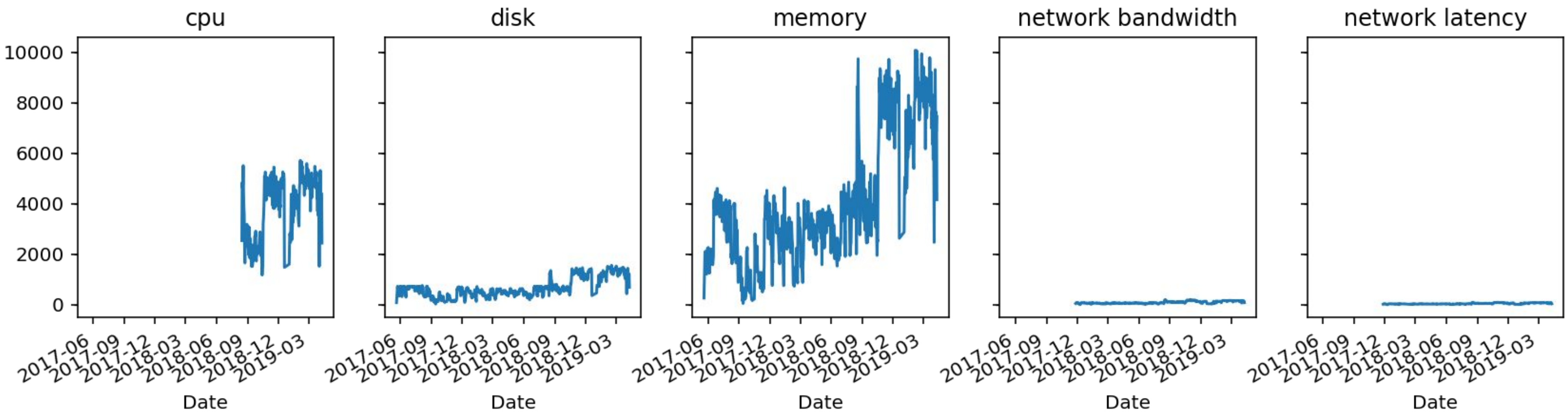
- Profiling
  - Cloud-scale (distributed) (Kanev et al., 2015, [1]) (Kozyrakis et al., 2010, [2])
  - Single-node (VM) applications (Yadwakar et al., 2014, [3])
- Quantifying Variability
  - Virtualized clouds (Iosup et al., 2011, [4])
  - Warehouse-scale computers (Dean and Barroso, 2013, [5])
- Other experimentation platforms
  - Baselining performance for Grid’5000 (Nussbaum, 2017, [6])

Summary of the Original Work

- How confident can we be in the correctness of our results?
  - Measure confidence with (non-parametric) CIs to account for unavoidable variability
- How many times should we run our experiments?
  - CONFIRM - Pick a target CI width, estimate number of runs using past performance data
- Can the facility help?
  - Provide statistically indistinguishable resources
- More results, experiences with pitfalls in the paper

Current Efforts

Continuously Collecting Performance Data

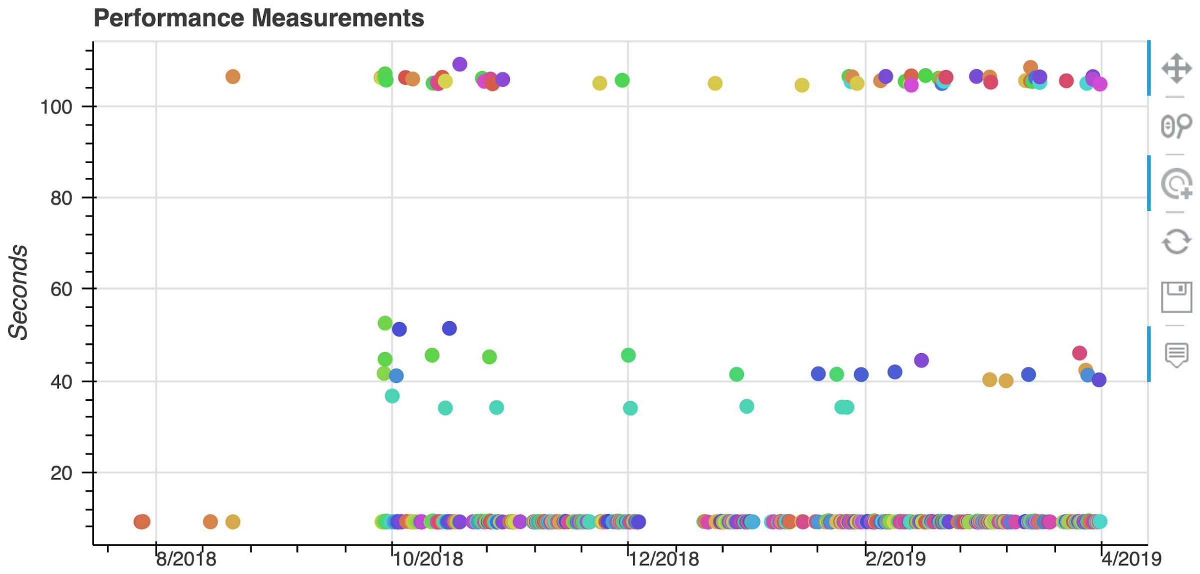


877 K | **452 K** | **2.7 M** | **47 K** | **24 K**

4 M, 1.3GB

As of Apr 9, 2019

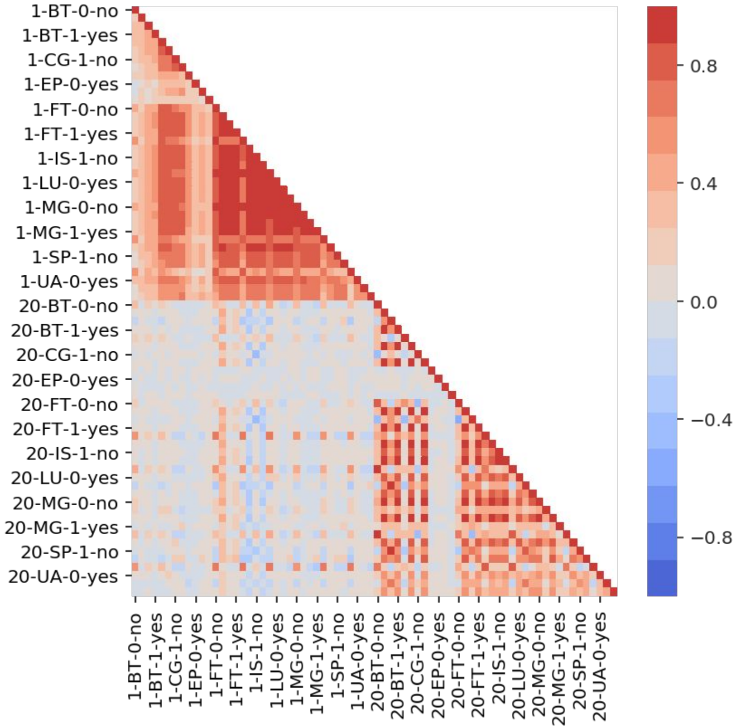
Highly Variable CPU Performance



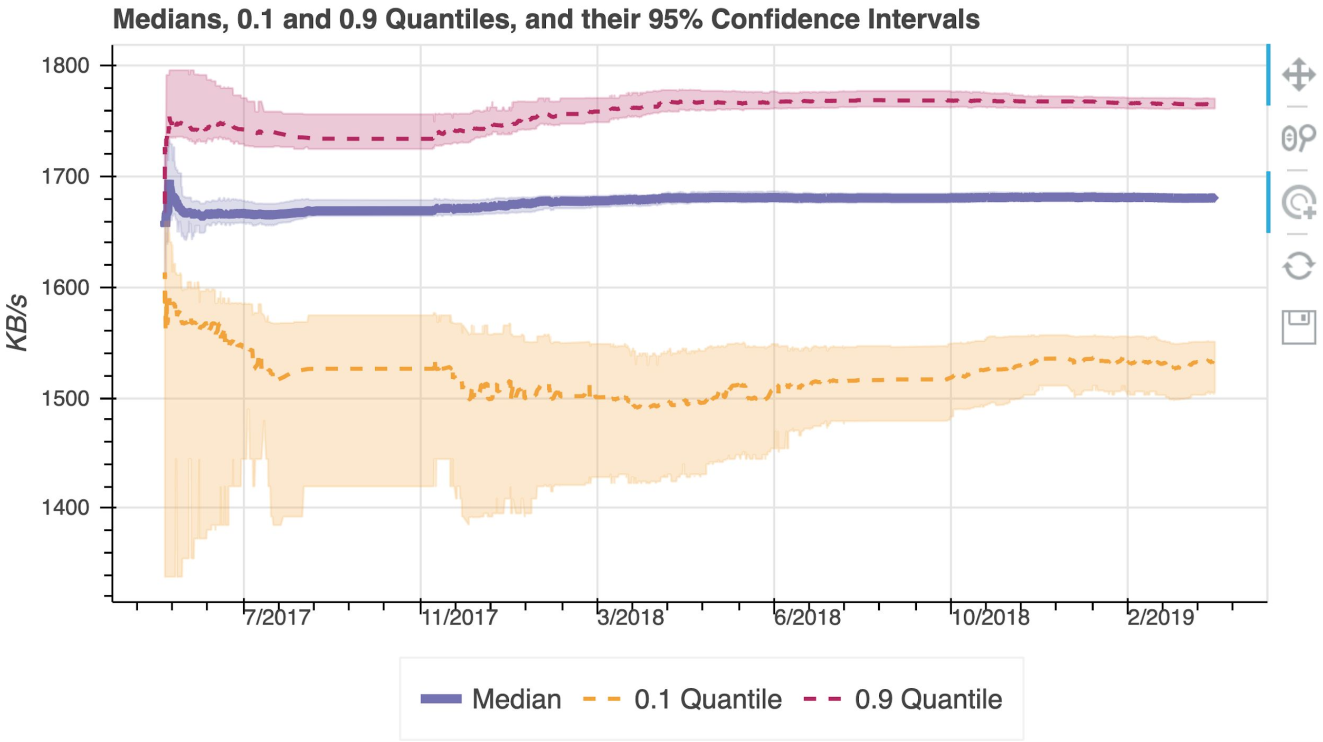
(Clemson, c6320, NPB Multi-Grid solver, Socket 0, DVFS on)

CoV = 152%!

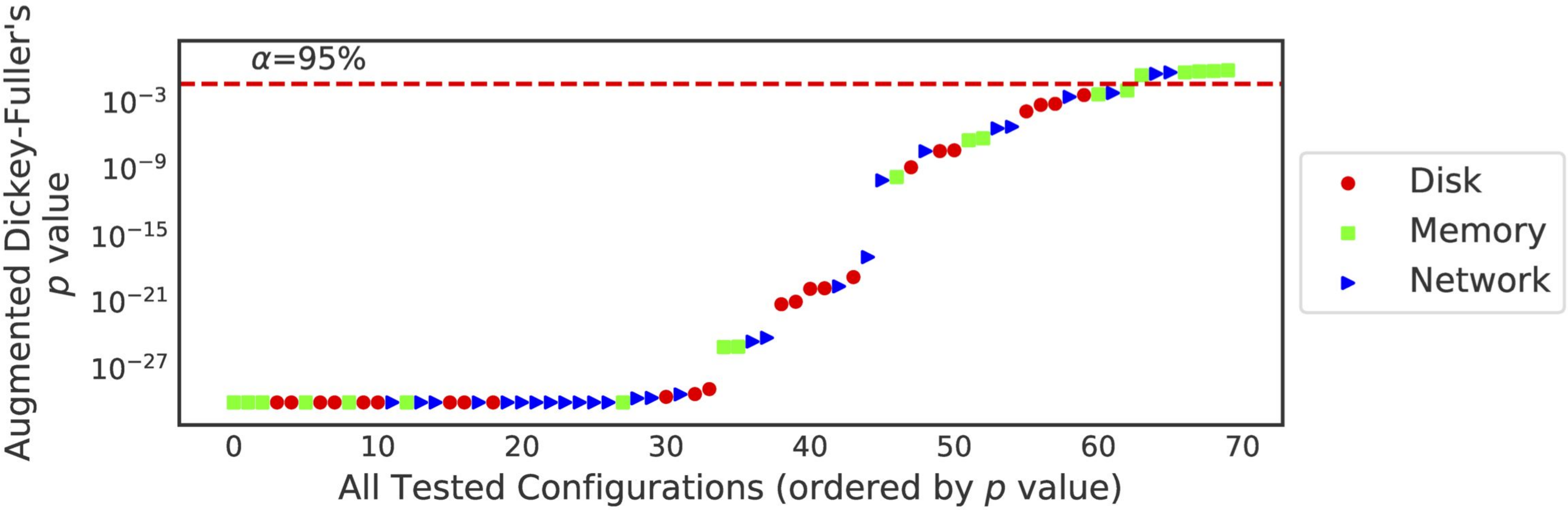
Exploring Correlations



Zooming into Performance *Tails*



Stationarity



Future Directions

Future Directions

- Randomization of benchmark order
- Change-point detection in gathered measurements
- Additional hardware and architectures
- Expand to other clouds and facilities

POWDER

https://www.powderwireless.net/

- **P**latform for **O**pen **W**ireless **D**ata-driven **E**xperimental **R**esearch
  - Flux Research Group - University of Utah
  - RENEW - Rice University
- Multiple Deployment areas
  - Encompases Campus, Downtown area, and a Residential neighborhood
- Fixed and Mobile endpoints

Summary: IoT and CPS

- Compute/Storage/Networking: Evaluate fine-grained ***performance variability***
- Sensory data: Explore and find patterns in ***environment variability***
- Modeling and Prediction: Establish and enforce QoS for ***learning variability***

Summary: IoT and CPS

- **Shapiro Wilks Test:** Check for normality
- **Non-Parametric Statistics:** Analyze non-Gaussian data
- **CONFIRM:** Change in CIs and Median over repeated measurements
- **Kernel Two-Sample Test:** How “representative” is a subset?
- **Augmented Fuller-Dickey Test:** Check for stationarity

References

[1]: Kanev et al., Profiling a warehouse-scale computer. ACM SIGARCH News, 2015.

[2]: Kozyrakis et al., Server engineering insights for large-scale online services. IEEE micro, 2010.

[3]: Yadwadkar et al., Predictable and faster jobs using fewer resources. SOCC'14.

[4]: Iosup et al, On the performance variability of production cloud services. CCGrid'11.

[5]: Dean and Barroso. The tail at scale. Communications of the ACM, 2013.

[6]: Nussbaum. Towards trustworthy testbeds thanks to throughout testing, IPDPSW’17.

CONFIRM — CONFIdence-based Repetition Meter

Designed by The Flux Research Group

CONFIRM is a tool for analysis of performance data that is continuously collected on CloudLab, an NSF-funded distributed testbed for cloud computing research and education. This data includes diverse measurements of CPU, memory, disk, and network performance for a wide range of hardware available on Cloudlab. CONFIRM's analysis provides statistical summaries and insights that help judge hardware performance from both user and facility perspectives.

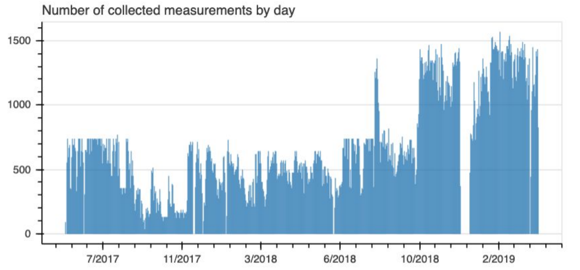
For more info, refer to the **"Taming Performance Varibility"** OSDI'18 paper.

Select ***type of benchmark***:

disk | memory | cpu

network bandwidth

network latency



Earliest measurement: 2017-05-20

Latest measurements: 2019-04-12

Number of collected measurements: 455,864

https://confirm.fyi