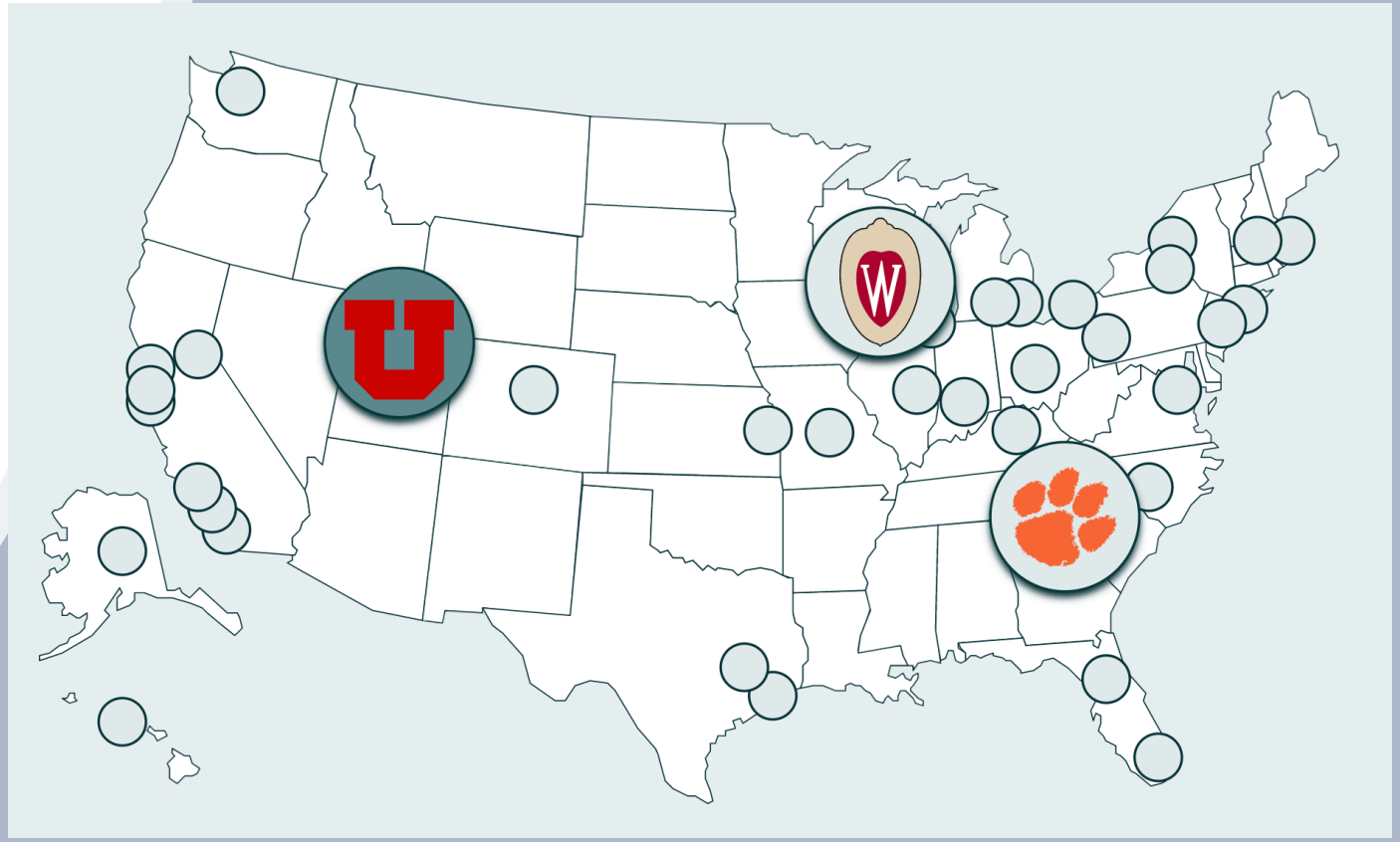
The Design and Operation of CloudLab

Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra

USENIX Annual Technical Conference 2019, July 10
Track 1, Real-World Deployed Systems

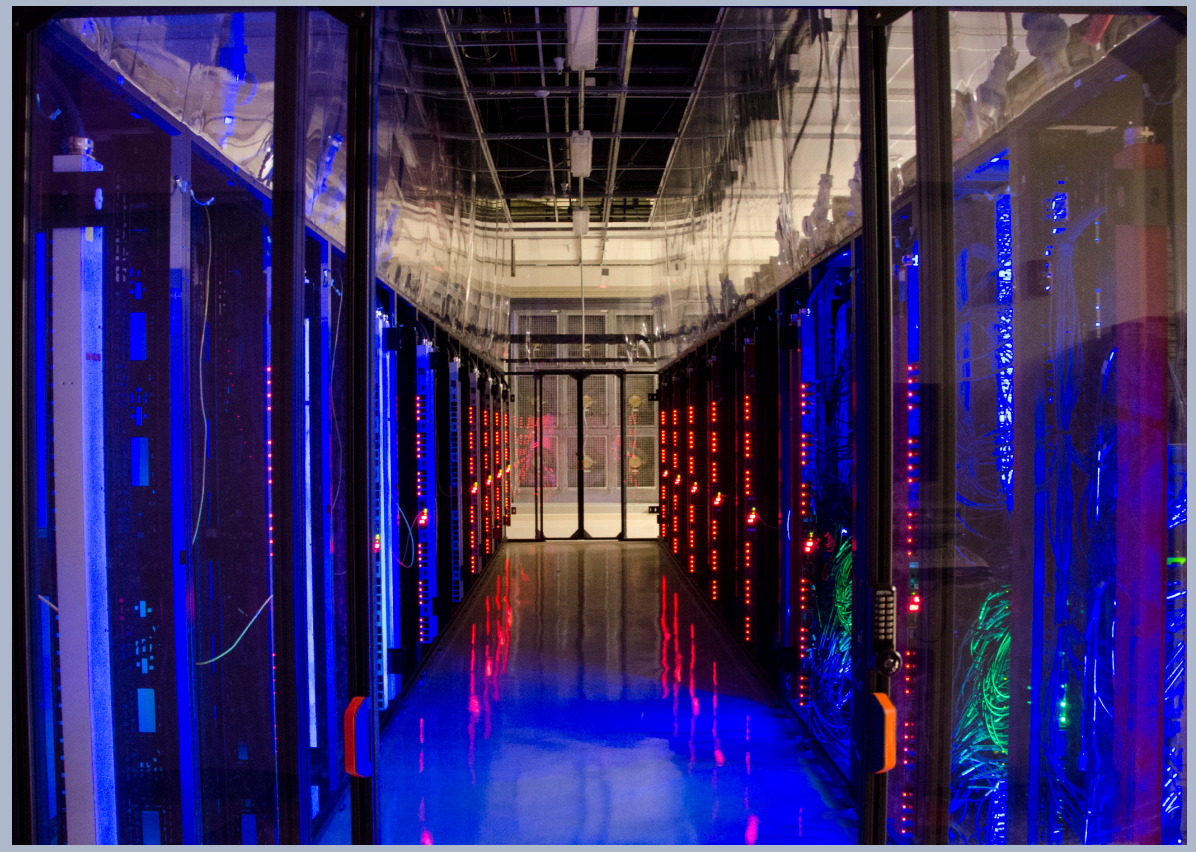
What is CloudLab ?

Distributed testbed for cloud computing and systems research



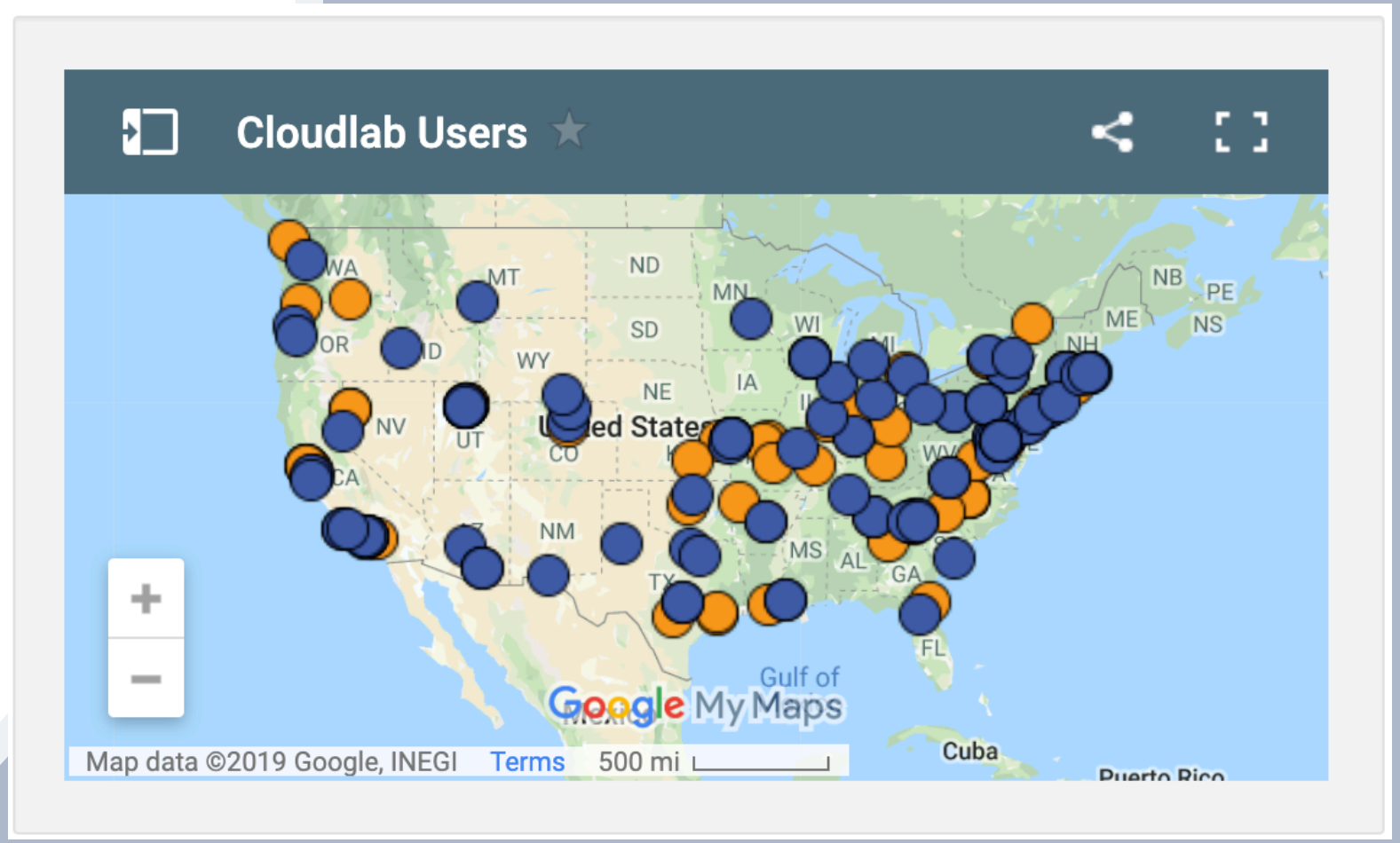
What is CloudLab ?

where experiments are run on **real, physical hardware** (not Virtual Machines)



What is CloudLab ?

with >4K users
from many U.S. institutions



What is CloudLab ?

A place to experiment with your own clouds and distributed systems

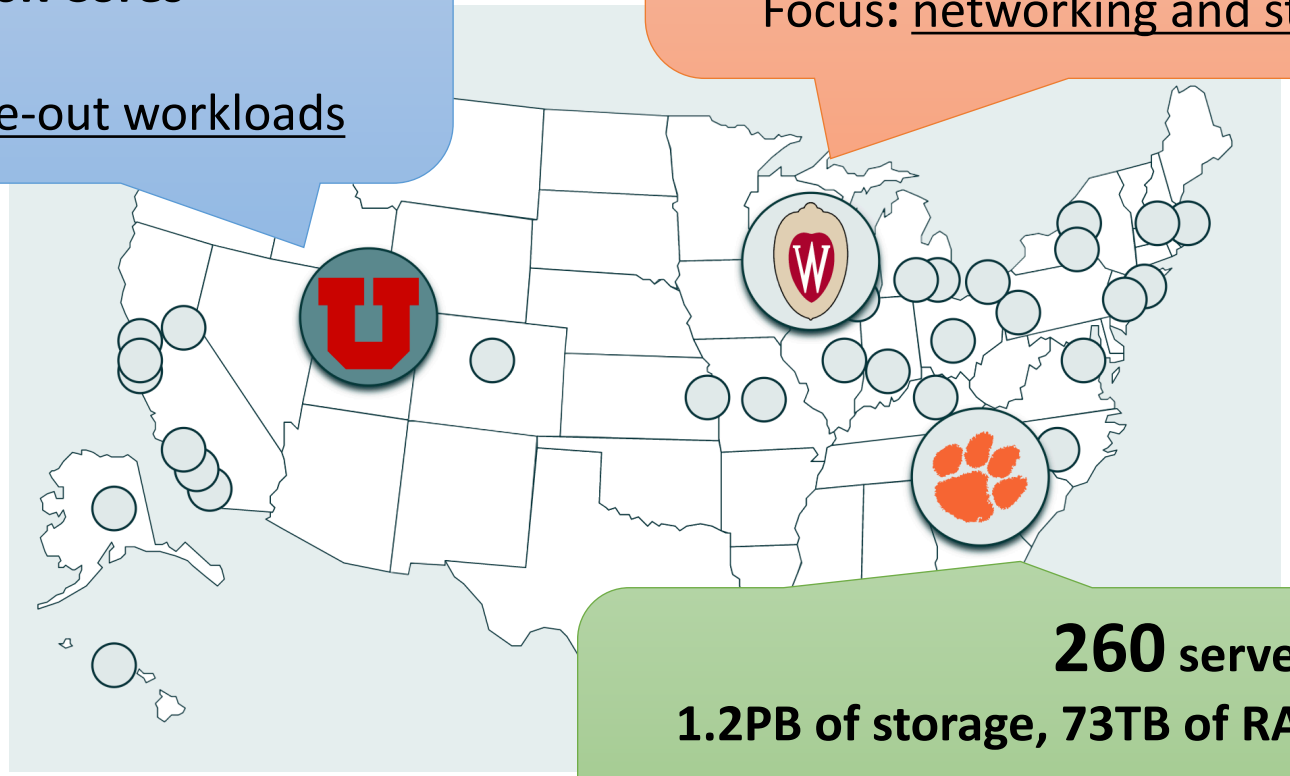
CloudLab Hardware

977 servers
10.6K Cores

Focus: scale-out workloads

527 servers
1PB of storage, SSD on every server

Focus: networking and storage work



260 servers
1.2PB of storage, 73TB of RAM, QDR Infiniband

Focus: analytics and high-performance workloads

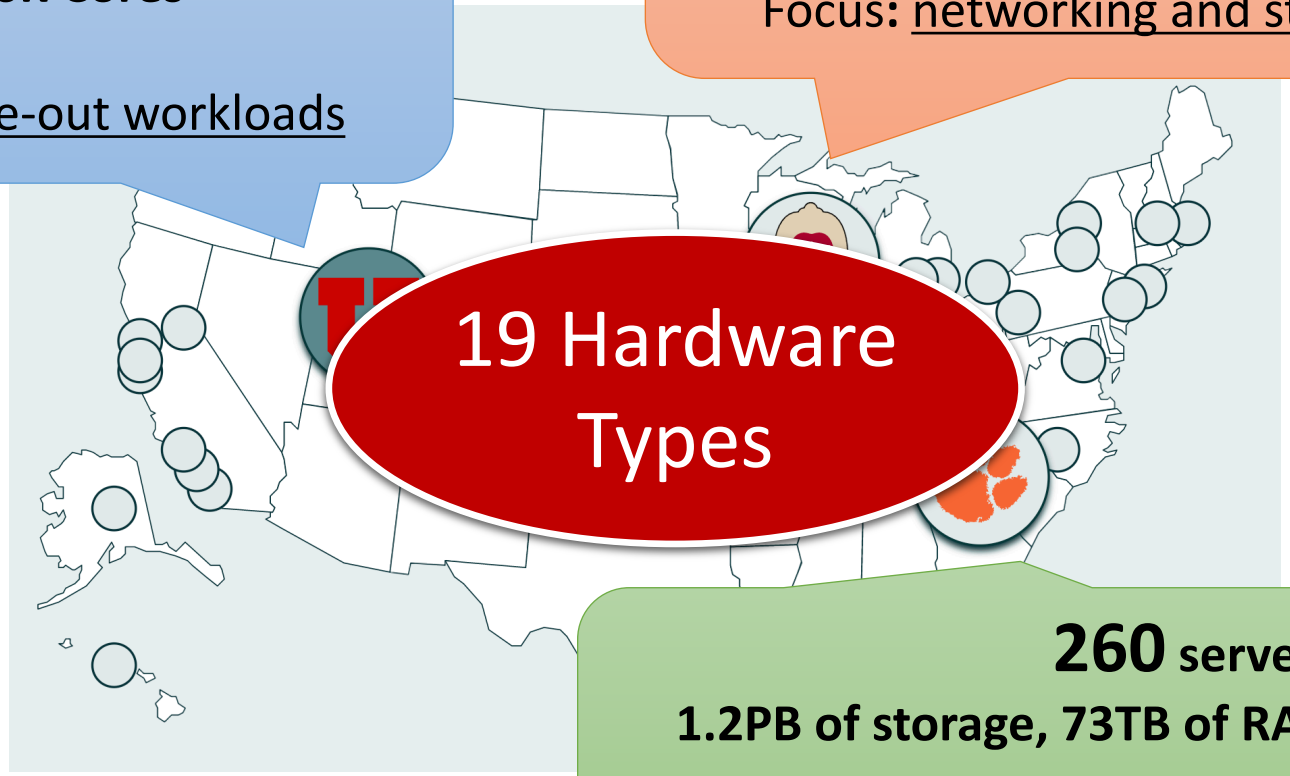
CloudLab Hardware

977 servers
10.6K Cores

Focus: scale-out workloads

527 servers
1PB of storage, SSD on every server

Focus: networking and storage work



260 servers
1.2PB of storage, 73TB of RAM, QDR Infiniband

Focus: analytics and high-performance workloads

CloudLab Hardware

More at: https://cloudlab.us

What do researchers use CloudLab for?

Networking	30%
Security	16%
Storage	11%
Applications	10%
Computing	9%
Virtualization	8%
Databases	7%
Middleware	4%
Energy & Power	2%
Other	15%

* Based on 93 papers from 2017-2018

Low-level access to hardware

Specific features

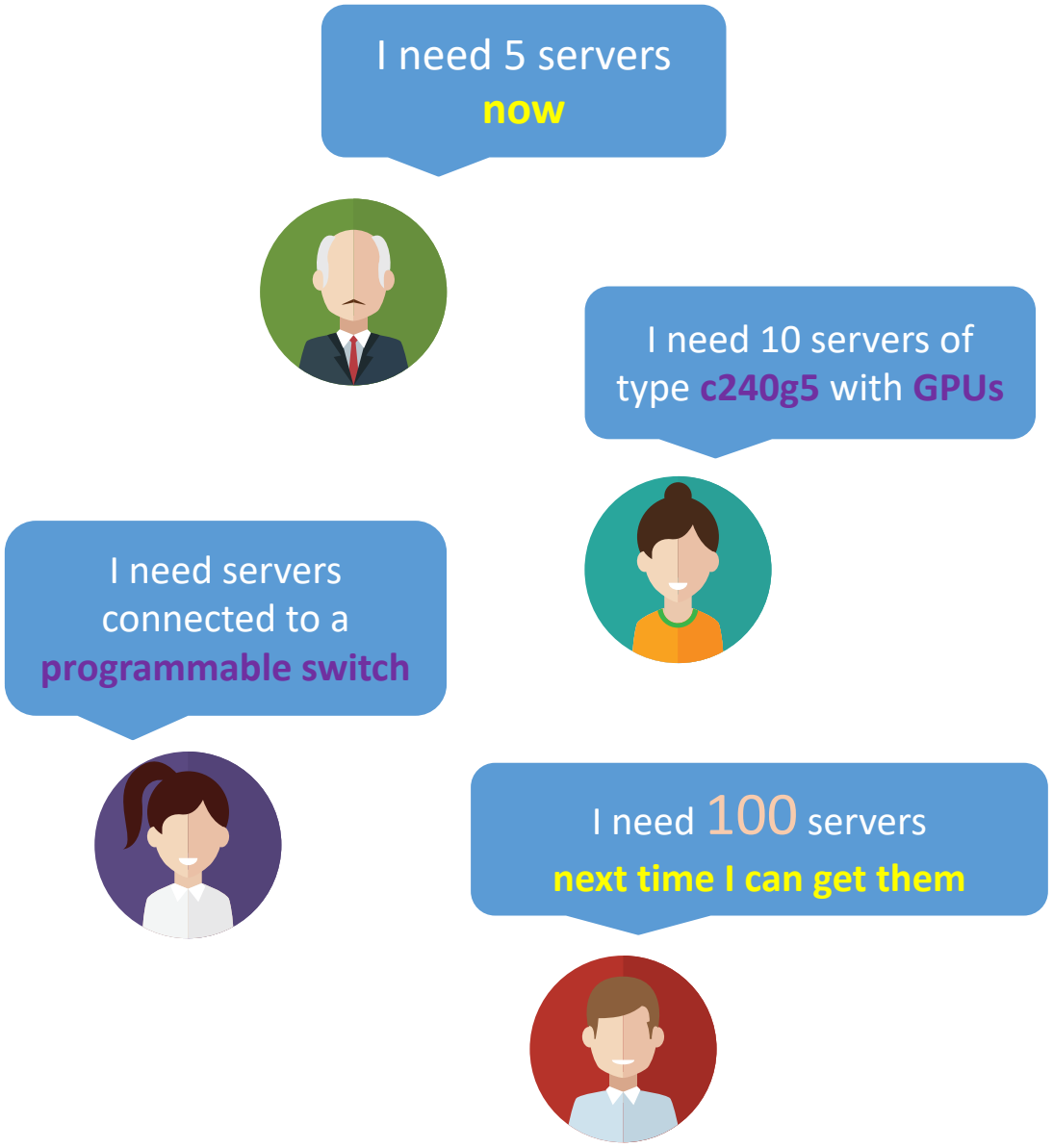
Performance isolation

Why study CloudLab ?

To better understand how well it serves **diverse researchers' needs**

To discern the **impact** of design decisions and associated **tradeoffs**

To offer the insights that would benefit the **design of other testbeds and IaaS facilities**

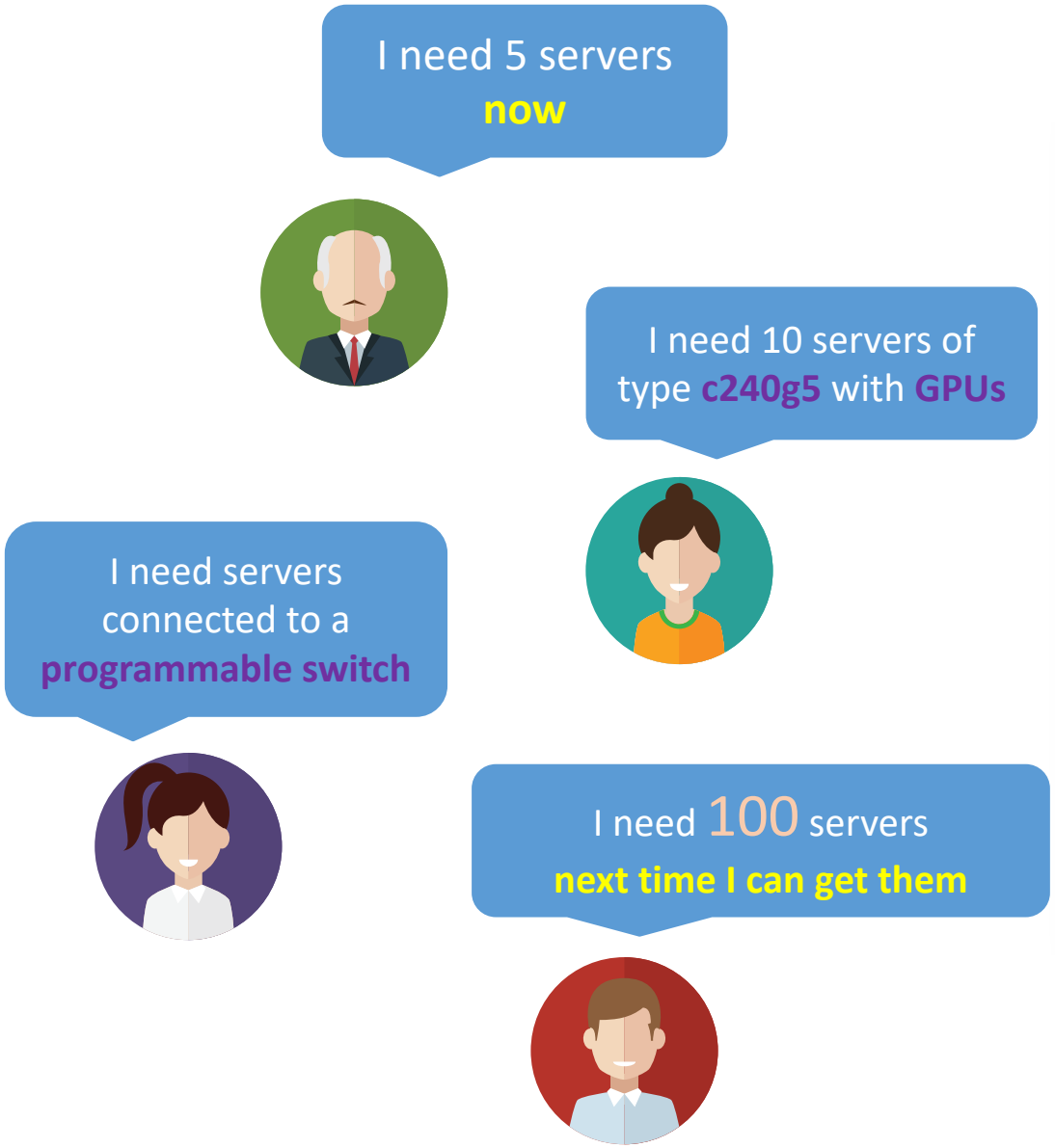


CloudLab needs to:

Satisfy diverse user needs (scale, time, and features)

Help users select feasible configurations

Return meaningful errors when request or facility issues occur



This study's questions:

Does CloudLab fulfill these goals?

How does CloudLab achieve them?

How can we generalize the lessons learned?

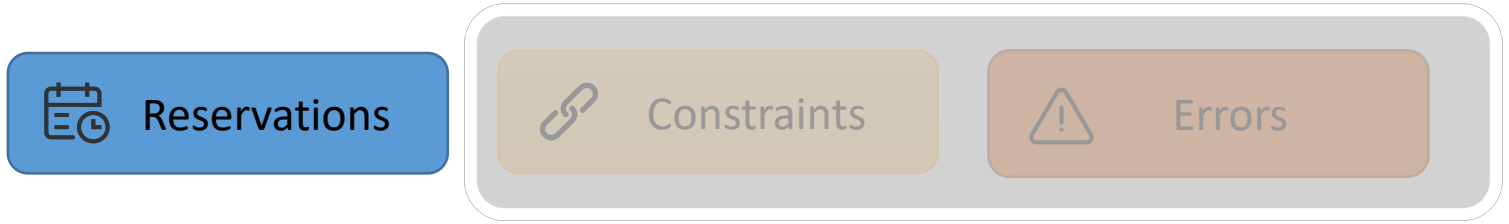
User Perspective

Satisfy diverse user needs (scale, time, and features)

Help users select feasible configurations

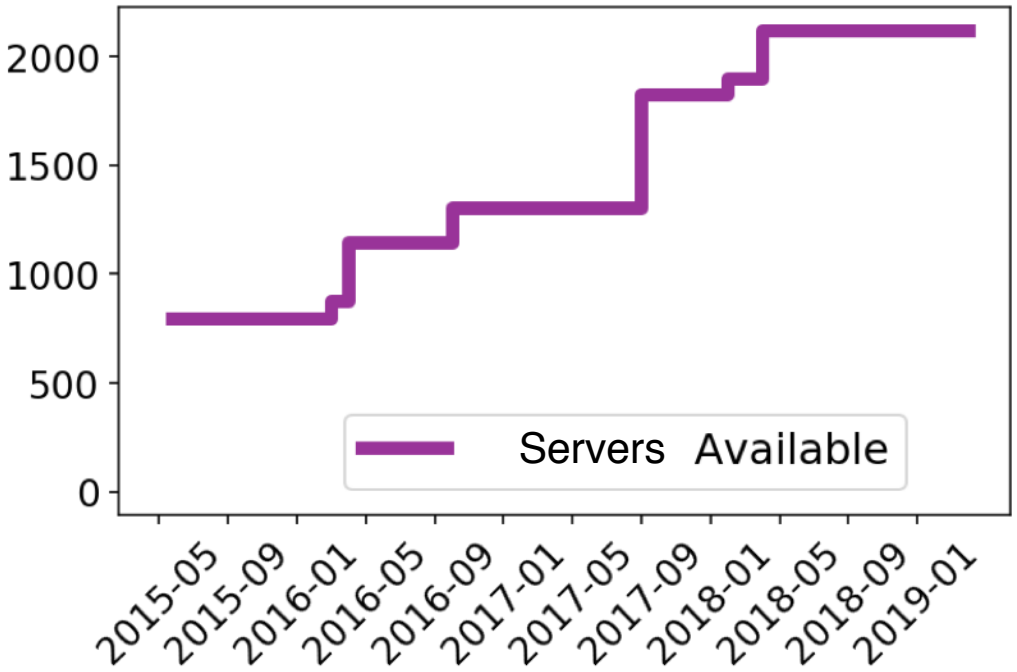
Return meaningful errors when request or facility issues occur

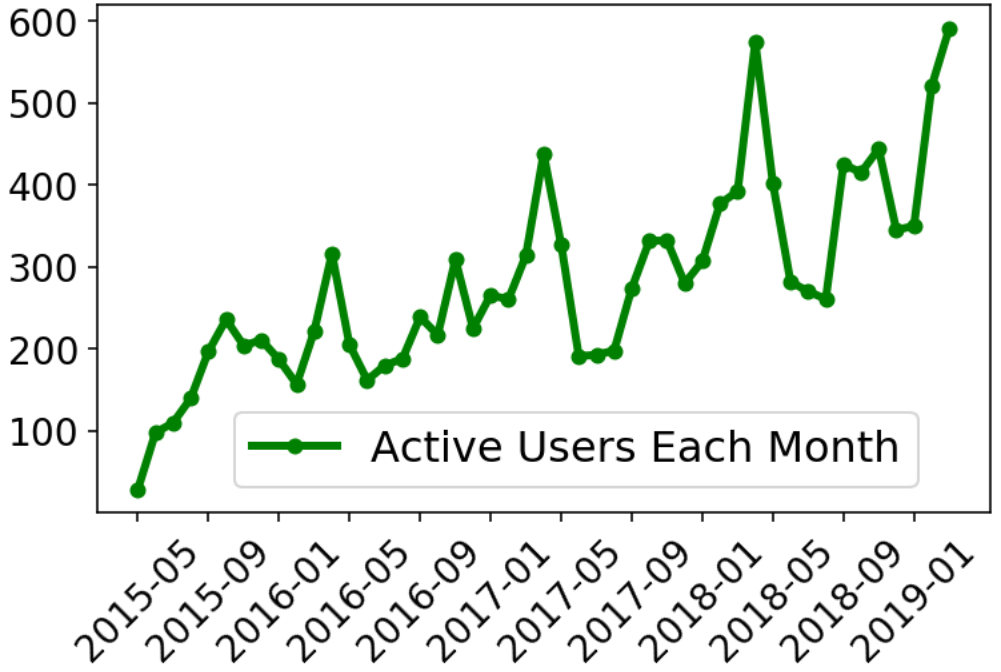
"Under the hood"

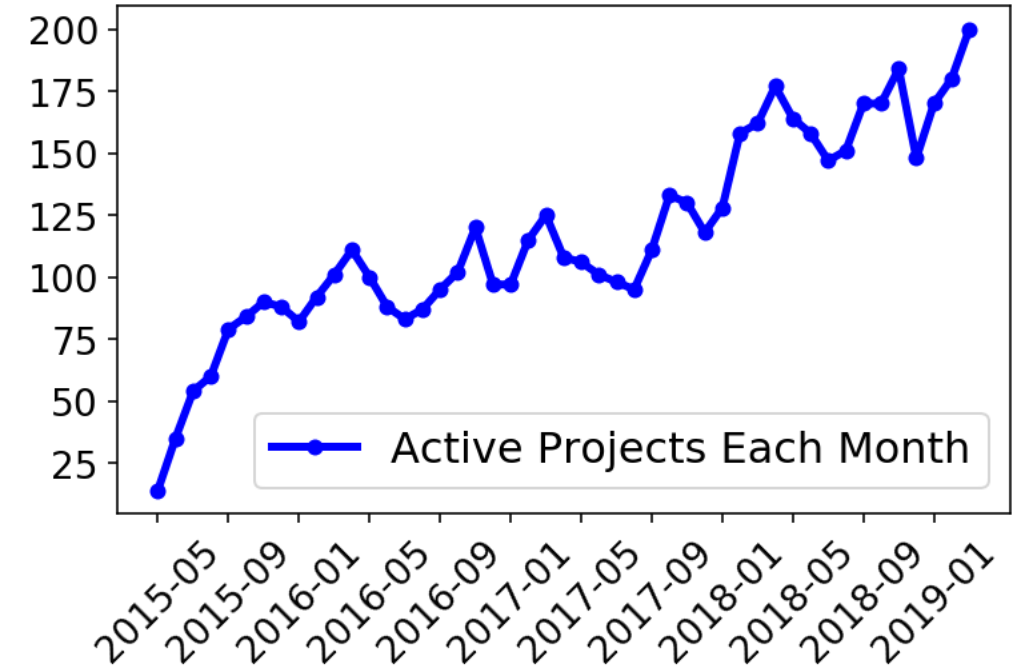


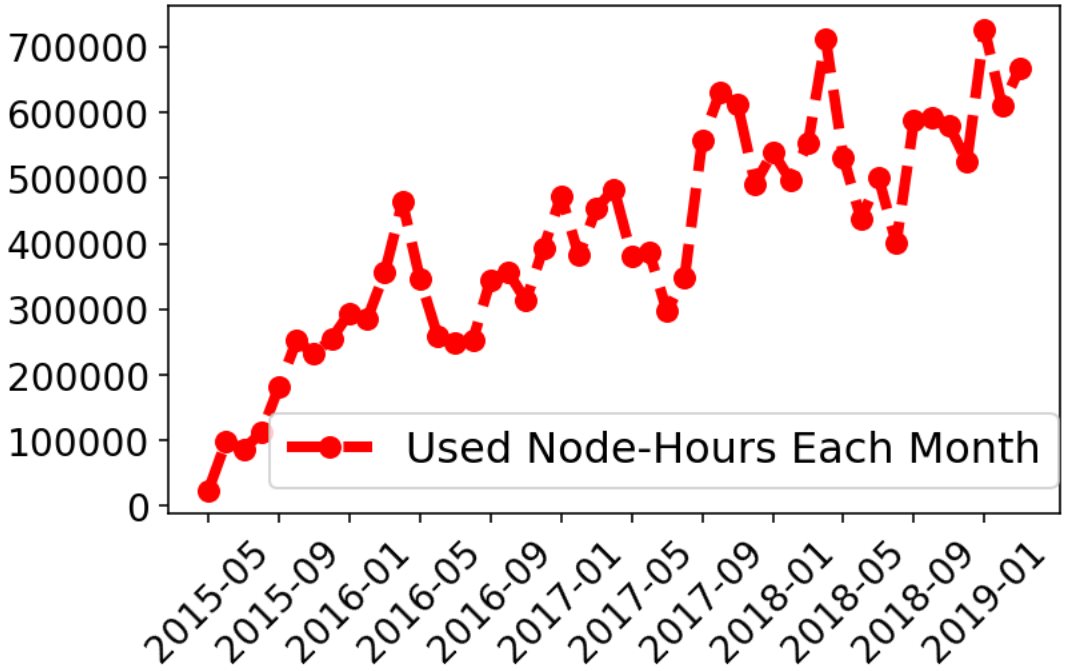
What can we learn from the historic data?

Growth and Usage

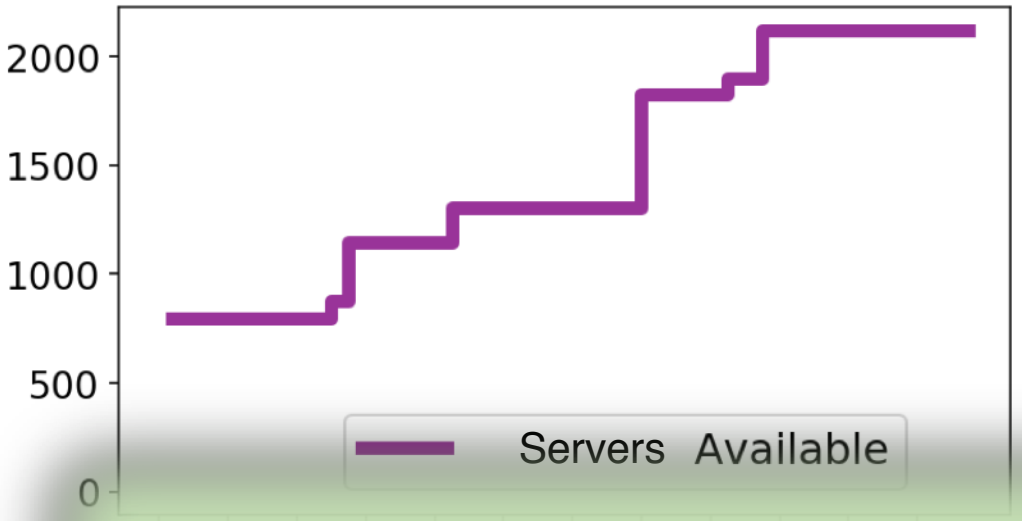


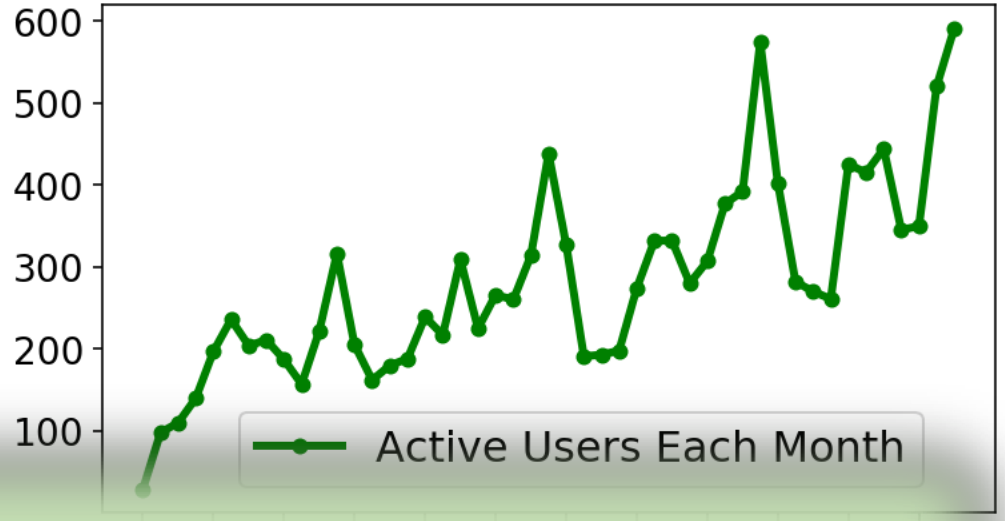




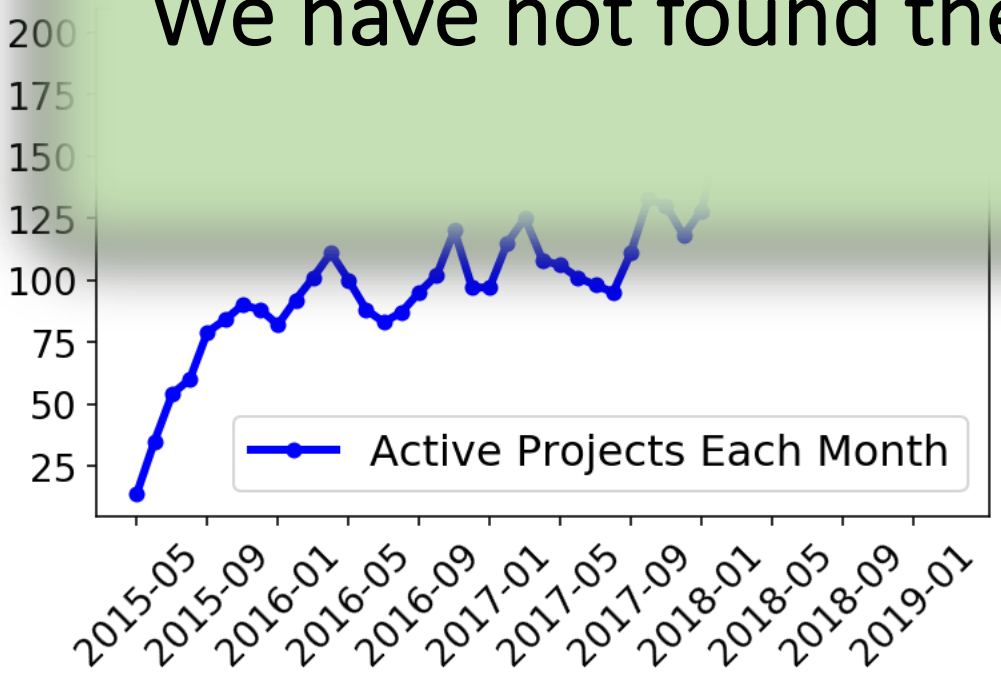


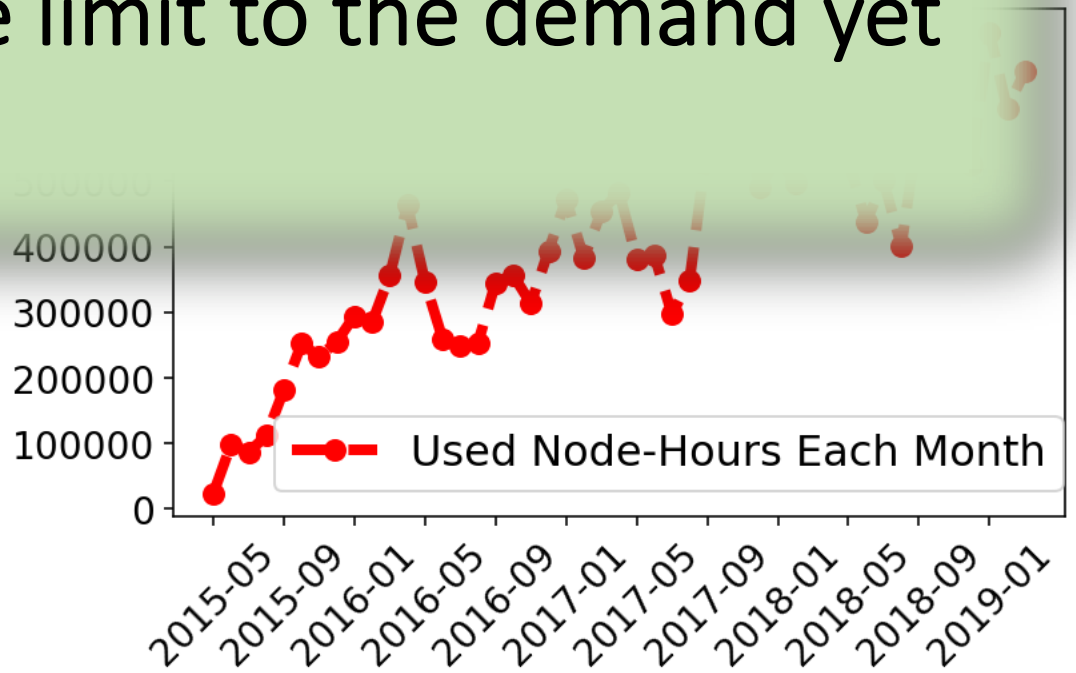
Growth and Usage





We have not found the limit to the demand yet

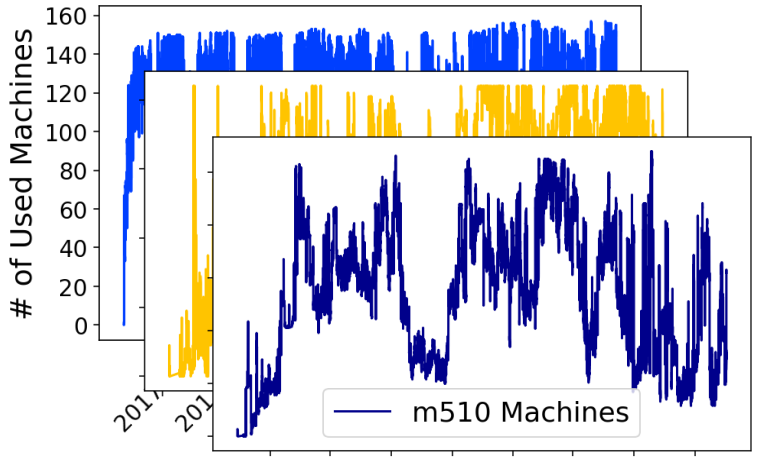


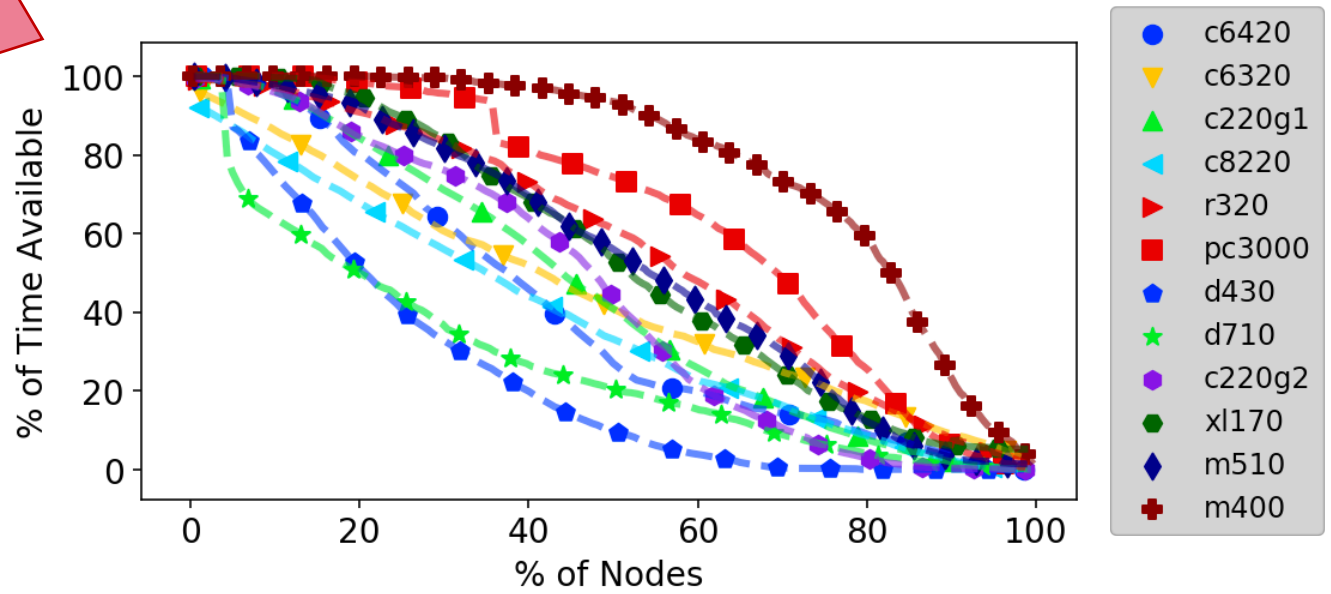


Growth and Usage

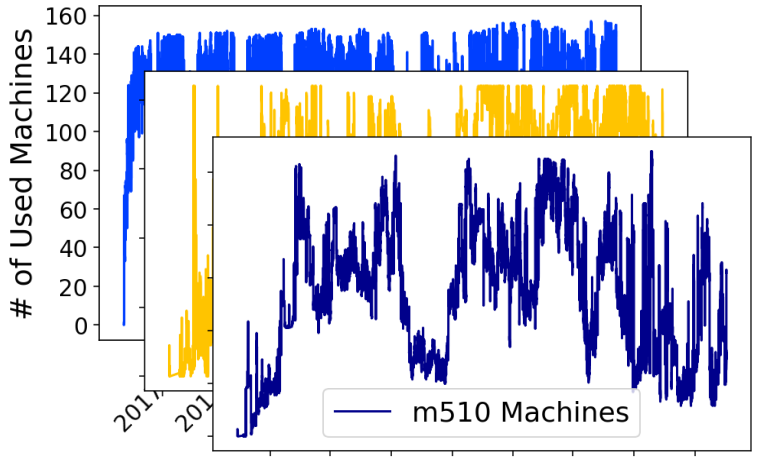
Is there enough hardware for everyone?

Growth and Usage

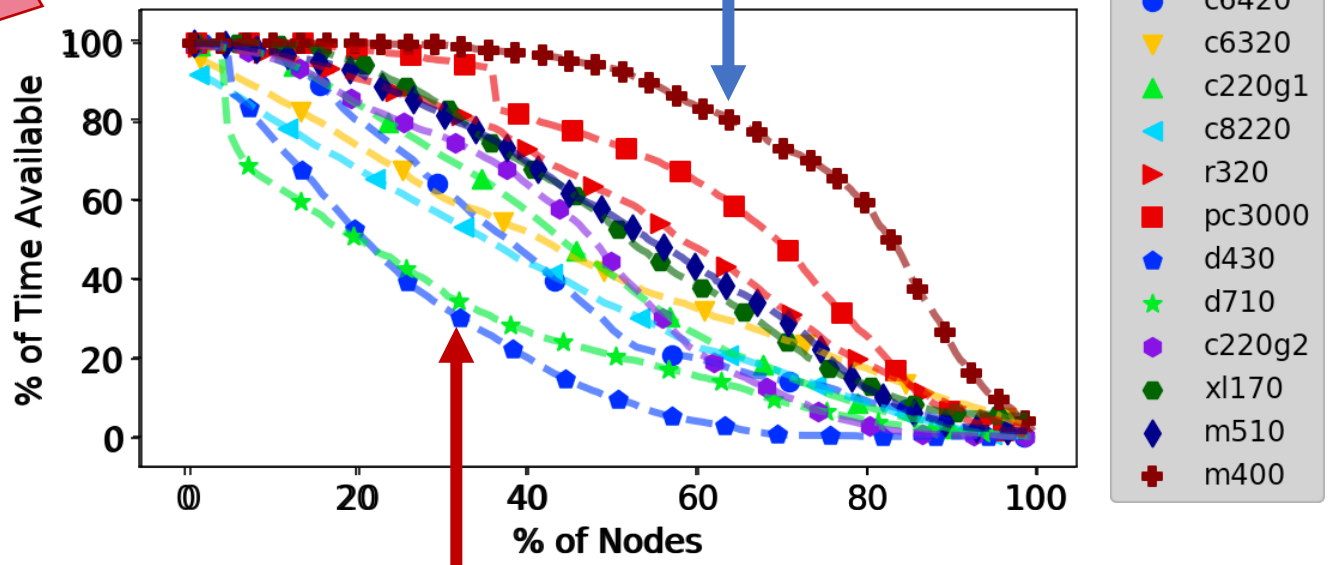




Growth and Usage

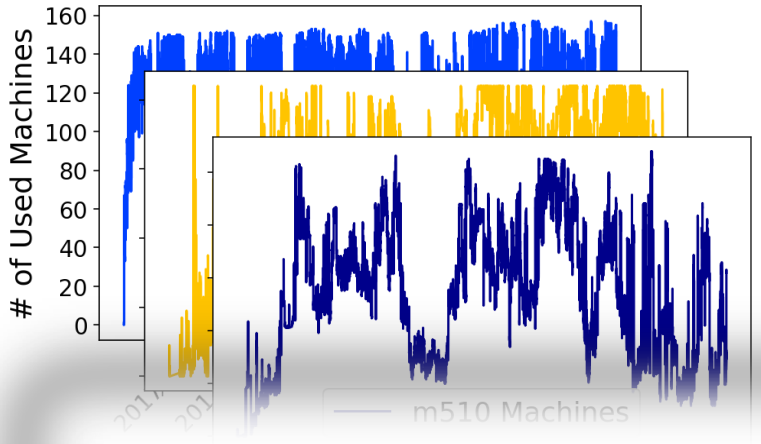


Wow, I can get **200** machines almost **80%** of the time!



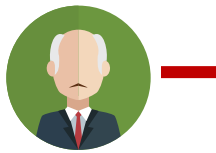
I can get **50** machines around **30%** of the time ☹

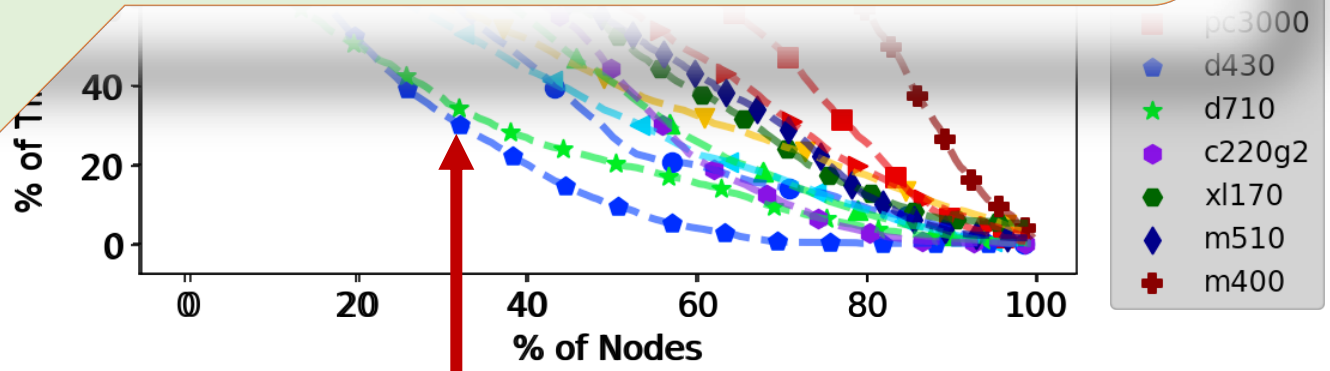
Growth and Usage

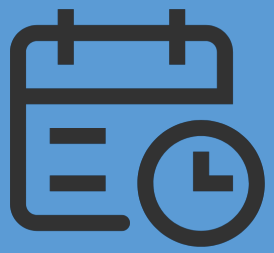


Wow, I can get **200** machines almost **80%** of the time!

I should submit reservation requests for this hardware







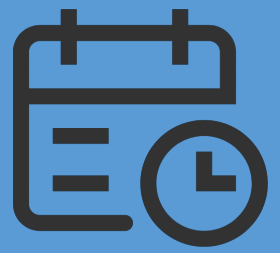
Reservations

I request **100** machines
of type **d430**
between **08/01/19** and **08/15/19**

Submitted **per project** and **per hardware type**

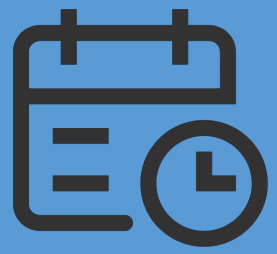
Subject to validation checks

Do not automatically launch experiments –
– only enforce availability

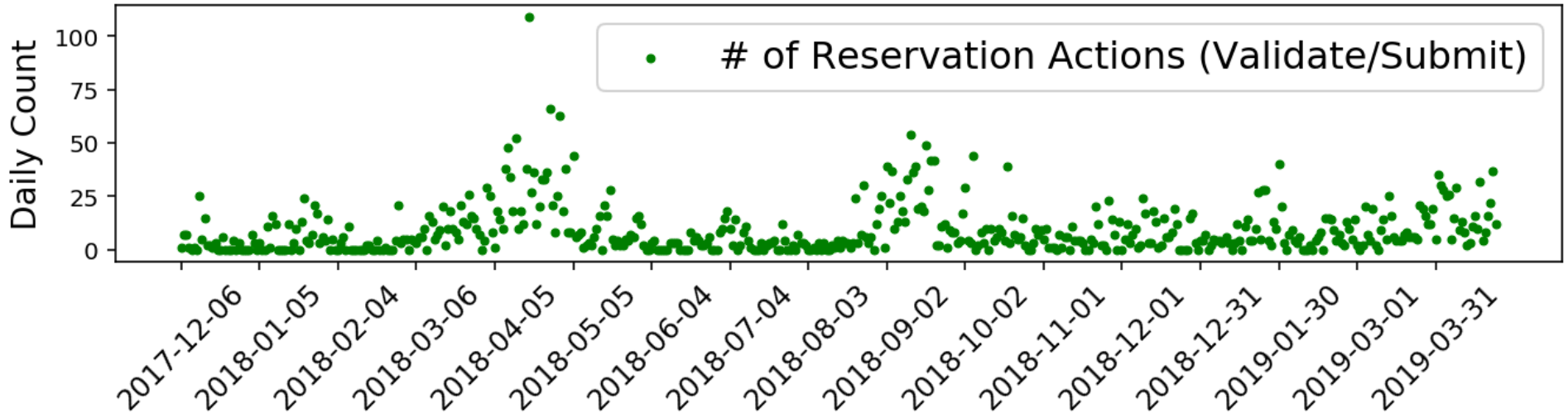


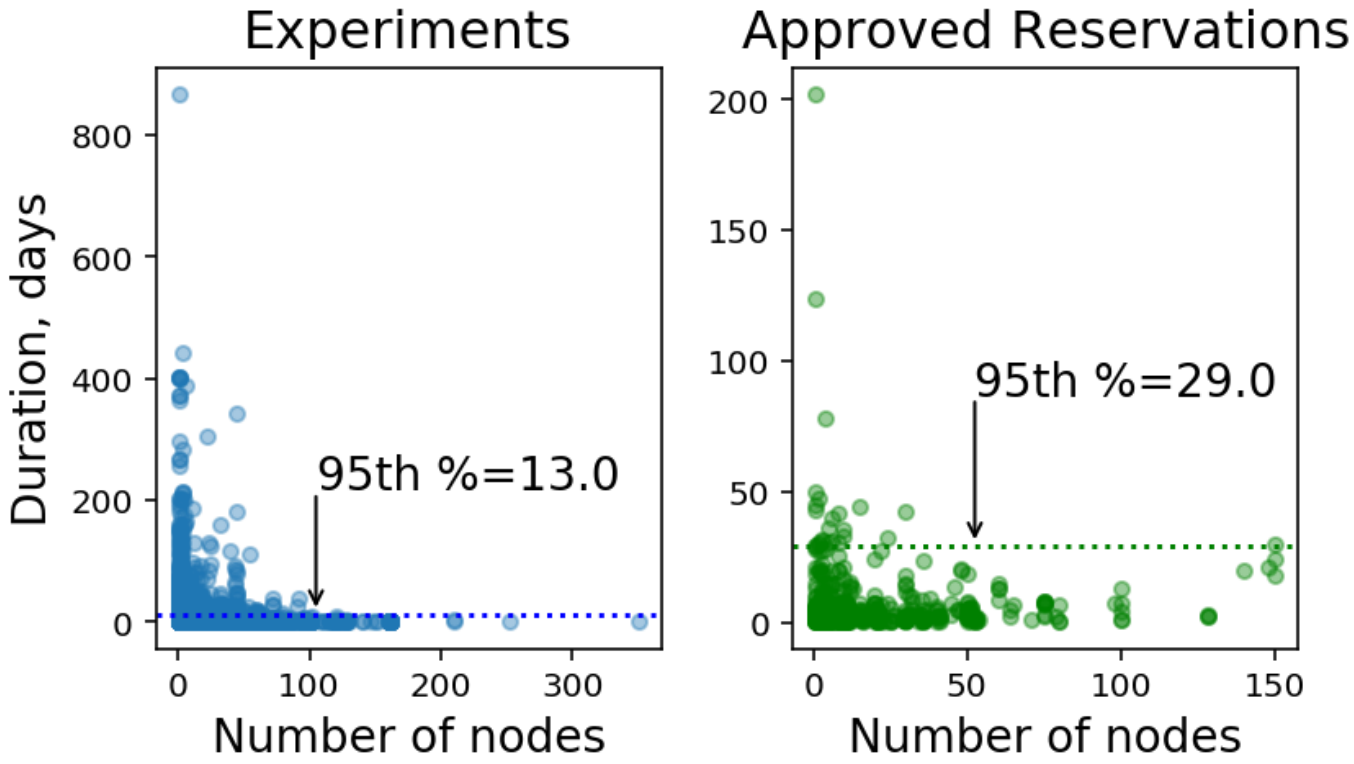
Reservations

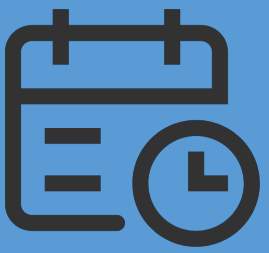
How well do they work in practice?



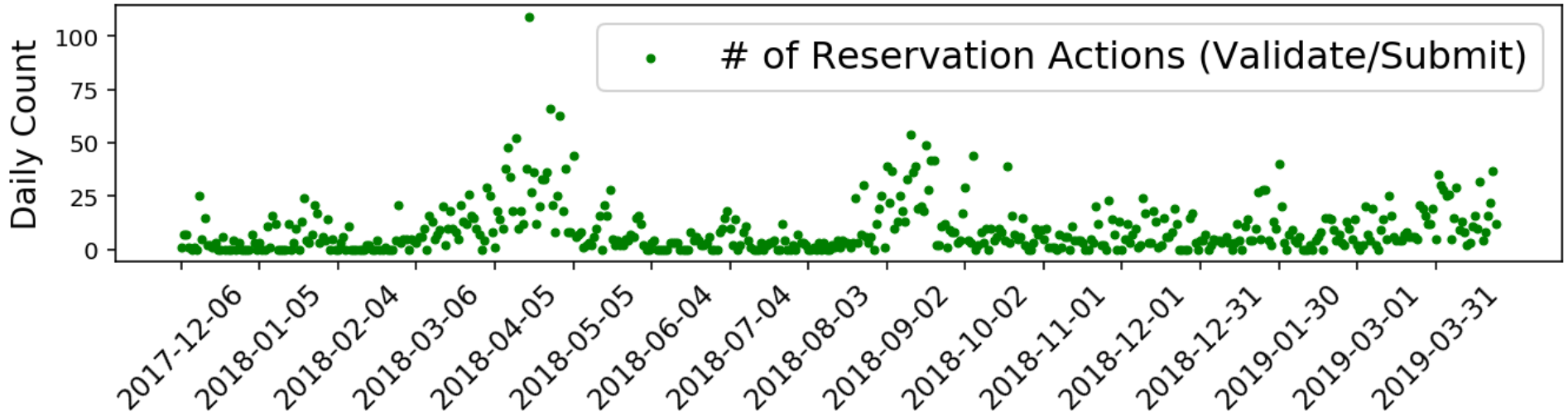
Reservations

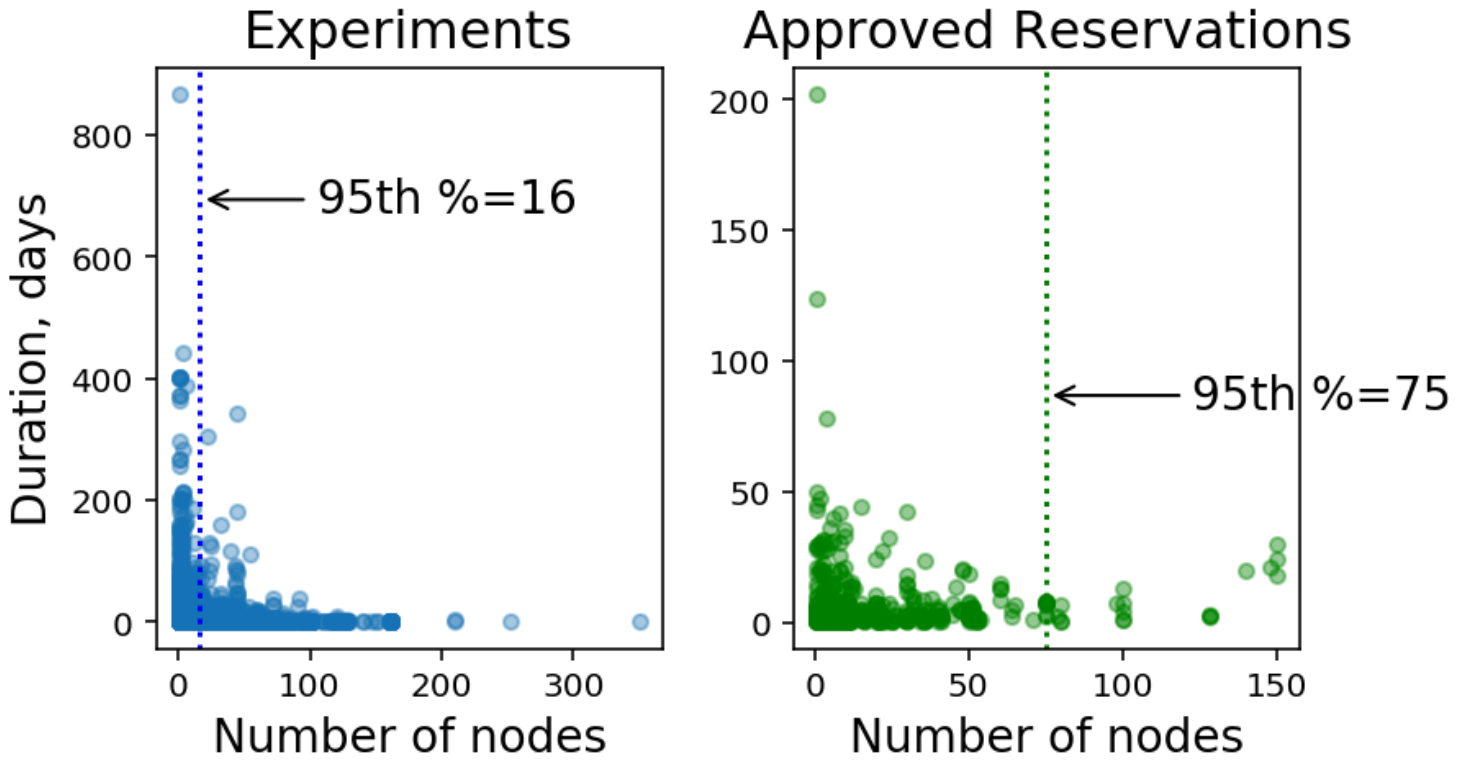


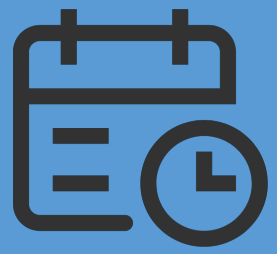




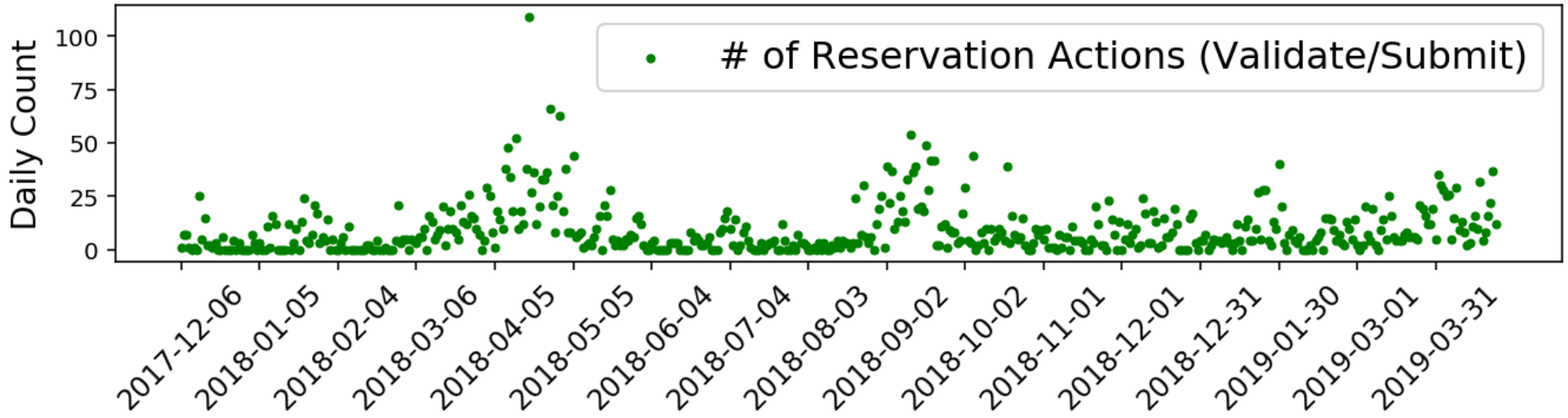
Reservations

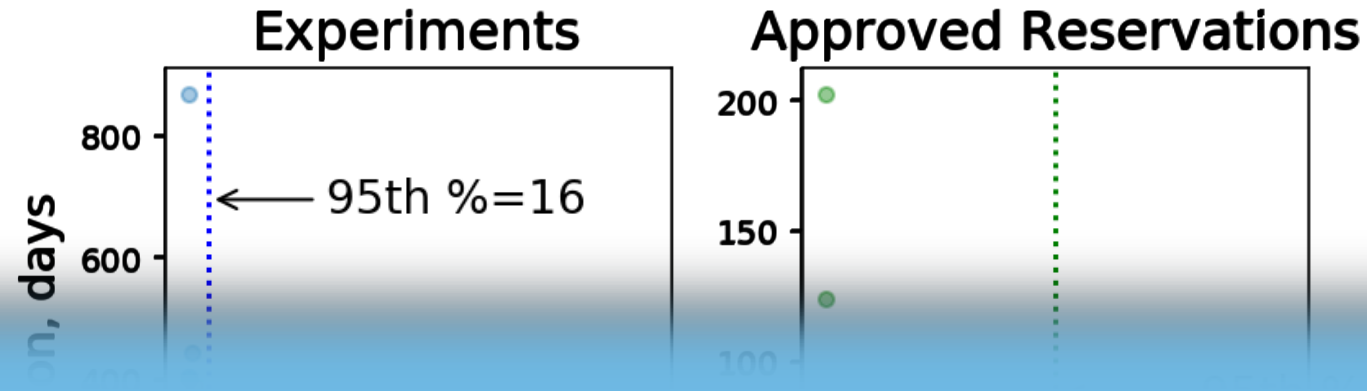




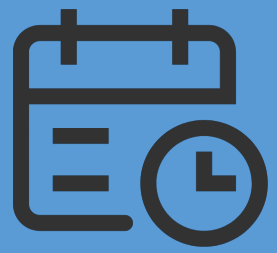


Reservations



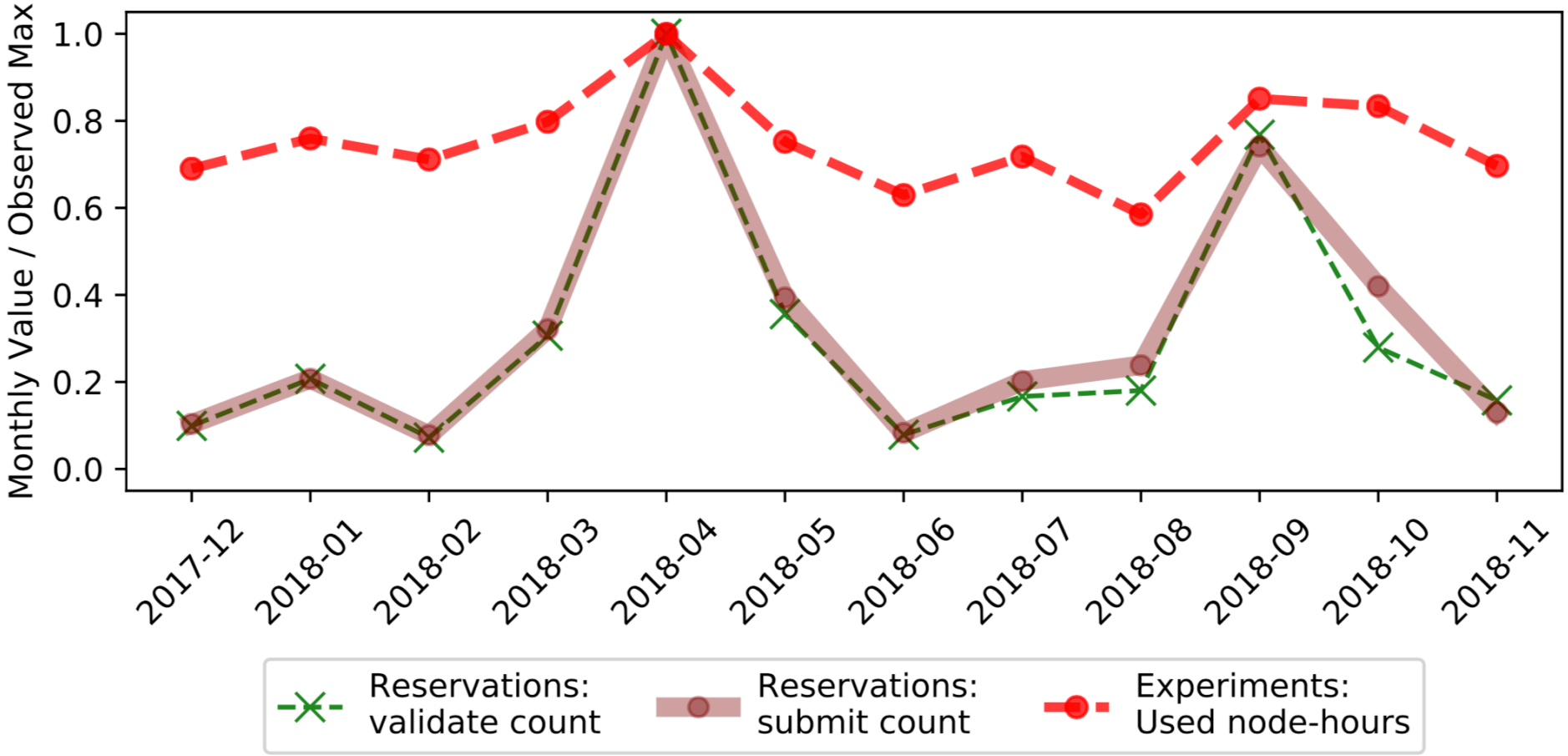


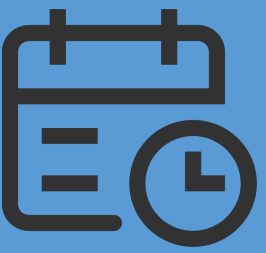
Reservations allow users to run **longer** and **larger** experiments



Reservations

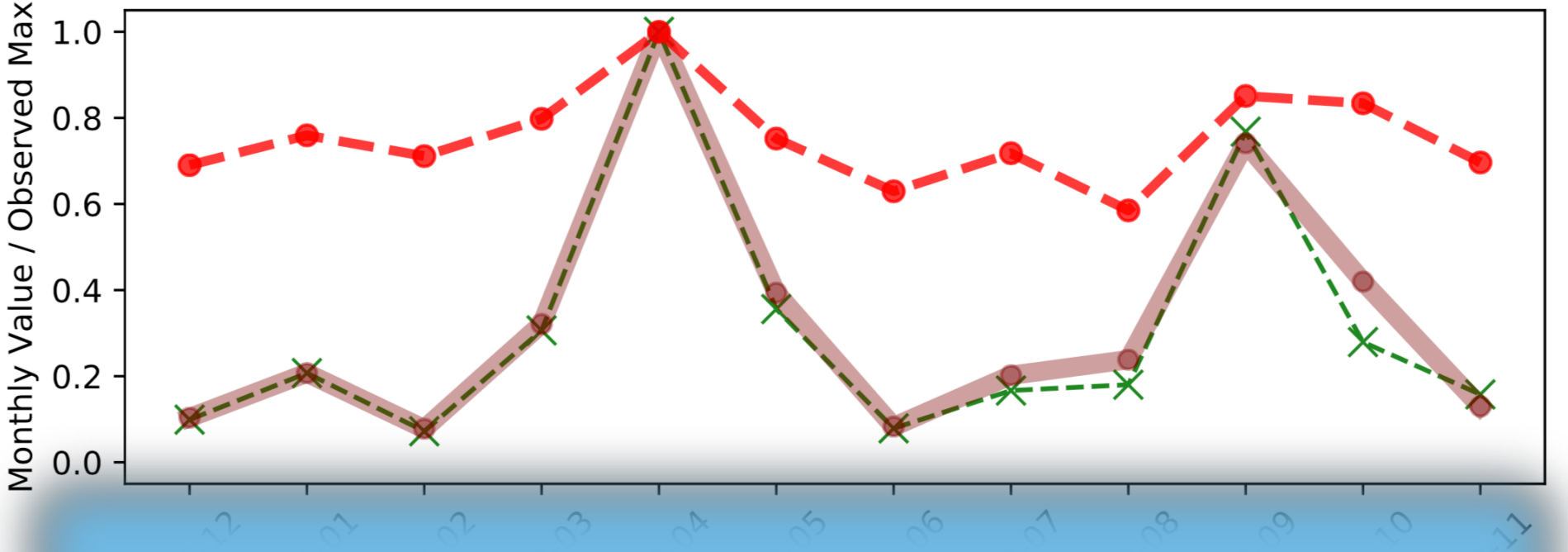
High testbed utilization ~ high use of reservations



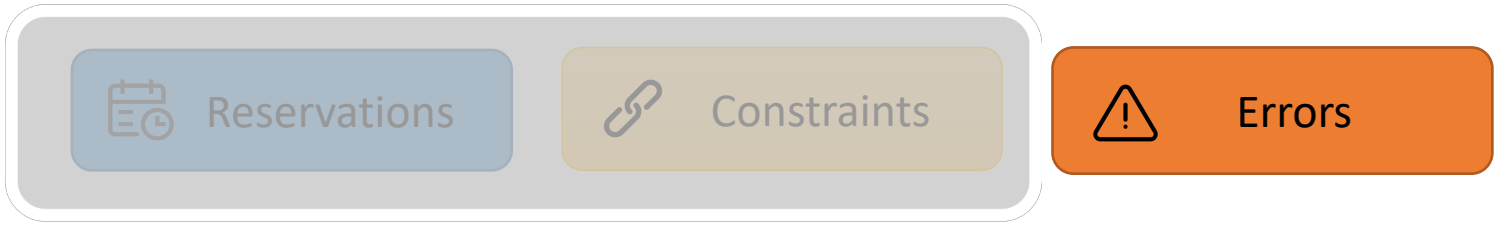


Reservations

High testbed utilization ~ high use of reservations



Reservations allow users to meet **deadlines**



How can the testbed efficiently assign resources to users?

Resource Mapping

Approaches to resource mapping:

General algorithm:	Specialized algorithm:
Very few assumptions	Knowledge of the facility
Constraint-satisfaction problem and optimization problem	More tailored and actionable feedback

Resource Mapping

Approaches to resource mapping:

General algorithm:	Specialized algorithm:
Very few assumptions	Knowledge of the facility
Constraint-satisfaction problem and optimization problem	More tailored and actionable feedback

CloudLab

Simulated annealing for solving graph isomorphism problem (NP-hard)

+

Set of deterministic heuristics as a wrapper for improved feedback

Errors

Results of Mapping Errors

Error Message	Helpful?	%
1. Resource reservation violation: X nodes of type HW requested, but only Y available	✓	27.79
2. X nodes of type HW requested, but only Y available nodes of type HW found	✓	21.86
3. No Possible Mapping for X: Too many links of type Y	✓	6.64
4. No Connection	✗	5.22
5. Insufficient Bandwidth	✗	4.88
6. No Possible Mapping for X: OS 'Y' does not run on this hardware type	✓	4.74
7. Not enough nodes because of policy restrictions or existing resource reservations	✓	4.37
8. No Possible Mapping for X: No physical nodes have feature Y	✓	3.54
9. Insufficient Nodes: Unexplained	✗	3.39
10. Fixed physical node X not available.	✓	2.56

Errors

We have identified common error scenarios and addressed them using custom heuristics

86.5% of last year's errors resulted in helpful error messages

How can the testbed help avoid some of these errors?

 Constraints System

VS.

 Error Reporting

Analogy

Feedback provided by an IDE

Feedback provided by a compiler

Constraints

Candidate: x={utah, m400, pc, ubuntu16-64-ARM}

Groups – whitelists of acceptable combinations with >=2 properties

Evaluation – Boolean ***product of sums***

Group relating site, hardware, and type:	Group relating hardware and image:
$a_1(x)$ ={utah, m510, xen}$\subseteq x$	$b_1(x)$ ={m400, ubuntu16-64-ARM}$\subseteq x$
$a_2(x)$ ={utah, m400, pc}$\subseteq x$	$b_2(x)$ ={m510, ubuntu16-64-STD}$\subseteq x$
...	...
$a_n(x)$ ={wisconsin, c220g2, pc}$\subseteq x$	$b_m(x)$ ={c220g2, fbsd110-64-STD}$\subseteq x$
$A(x) = a_1(x) \vee a_2(x) \vee \ldots \vee a_n(x)$	$B(x) = b_1(x) \vee b_2(x) \vee \ldots \vee b_m(x)$

$a_2(x) \wedge b_1(x) = 1 \rightarrow$ candidate passes the check

Constraints

Candidate: $x=\{utah, m400, pc, ubuntu16\text{-}64\text{-}ARM\}$

Used in Two Contexts

Interactive Topology Design:

Early feedback/warnings

More permissive

Cluster Selection:

Block instantiation if request is infeasible

Disable selection of incompatible clusters

More conservative

$a_2(x) \wedge b_1(x) = 1 \rightarrow$ candidate passes the check

Constraints

Candidate: x={utah, m400, pc, ubuntu16-64-ARM}

Constraints checker running as a **lightweight** system in front of the **complex** mapper improves user experience

$a_2(x) \wedge b_1(x) = 1 \rightarrow$ candidate passes the check

In Conclusion

Reservation System

Reservations allow users to run **longer** and **larger** experiments and meet **deadlines**

Constraints System

Constraints checker running as a **lightweight** system in front of the mapper improves user experience

Error Reporting

The identified common and addressed error scenarios account for **86.5%** of errors; they lead to helpful messages

In Conclusion

Satisfies diverse researchers' needs, helps them select feasible configurations, and provides helpful feedback

Data & Code

Activity of **4K users**

in **79K experiments**

on over **2K servers**

for over **4 years**

and complete record of testbed events, states, and errors

https://gitlab.flux.utah.edu/emulab/cloudlab-usage

Awards: 1419199 and 1743363

More about CloudLab

Sign up: https://cloudlab.us

Join the BoF later today: 7:30-8:30pm
in Seattle Room

Thank you!

Dmitry Duplyakin
dmdu@cs.utah.edu