The Design and Operation of CloudLab

What is CloudLab?

Distributed testbed for cloud computing and systems research

2 | The Design and Operation of CloudLab
What is CloudLab?

where experiments are run on real, physical hardware (not Virtual Machines)
What is CloudLab?
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

19 Hardware Types

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?

<table>
<thead>
<tr>
<th>Topic</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networking</td>
<td>30%</td>
</tr>
<tr>
<td>Security</td>
<td>16%</td>
</tr>
<tr>
<td>Storage</td>
<td>11%</td>
</tr>
<tr>
<td>Applications</td>
<td>10%</td>
</tr>
<tr>
<td>Computing</td>
<td>9%</td>
</tr>
<tr>
<td>Virtualization</td>
<td>8%</td>
</tr>
<tr>
<td>Databases</td>
<td>7%</td>
</tr>
<tr>
<td>Middleware</td>
<td>4%</td>
</tr>
<tr>
<td>Energy &amp; Power</td>
<td>2%</td>
</tr>
<tr>
<td>Other</td>
<td>15%</td>
</tr>
</tbody>
</table>

* 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.
I need 5 servers now

I need 10 servers of type c240g5 with GPUs

I need servers connected to a programmable switch

I need 100 servers next time I can get them

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
I need 5 servers now.

I need 10 servers of type c240g5 with GPUs.

I need servers connected to a programmable switch.

I need 100 servers next time I can get them.

**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”

Reservation System

Constraints System

Error Reporting
What can we learn from the historic data?
We have not found the limit to the demand yet
Is there enough hardware for everyone?
Growth and Usage
Wow, I can get 200 machines almost 80% of the time!

I can get 50 machines around 30% of the time 😊
Wow, I can get 200 machines almost 80% of the time!

I should submit reservation requests for this hardware.
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
How well do they work in practice?
Reservations

The Design and Operation of CloudLab Reservations

Graph showing daily count of reservation actions.

- Experiments graph:
  - Duration, days vs Number of nodes
  - 95th % = 13.0

- Approved Reservations graph:
  - Number of nodes
  - 95th % = 29.0
Reservations

Experiments

95th % = 16

Approved Reservations

95th % = 75

Daily Count

# of Reservation Actions (Validate/Submit)
Reservations allow users to run longer and larger experiments.
High testbed utilization ~ high use of reservations
High testbed utilization ~ high use of reservations

Reservations allow users to meet **deadlines**.
How can the testbed efficiently assign resources to users?
Approaches to resource mapping:

**General algorithm:**
- Very few assumptions
- Constraint-satisfaction problem and optimization problem

**Specialized algorithm:**
- Knowledge of the facility
- More tailored and actionable feedback
Approaches to resource mapping:

**General algorithm:**
- Very few assumptions
- Constraint-satisfaction problem and optimization problem

**Specialized algorithm:**
- Knowledge of the facility
- More tailored and actionable feedback

Simulated annealing for solving graph isomorphism problem (NP-hard) + Set of deterministic heuristics as a wrapper for improved feedback
## Results of Mapping Errors

### Error Message

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Helpful?</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Resource reservation violation: X nodes of type HW requested, but only Y available</td>
<td>✓</td>
<td>27.79</td>
</tr>
<tr>
<td>2. X nodes of type HW requested, but only Y available nodes of type HW found</td>
<td>✓</td>
<td>21.86</td>
</tr>
<tr>
<td>3. No Possible Mapping for X: Too many links of type Y</td>
<td>✓</td>
<td>6.64</td>
</tr>
<tr>
<td>4. No Connection</td>
<td>✗</td>
<td>5.22</td>
</tr>
<tr>
<td>5. Insufficient Bandwidth</td>
<td>✗</td>
<td>4.88</td>
</tr>
<tr>
<td>6. No Possible Mapping for X: OS ’Y’ does not run on this hardware type</td>
<td>✓</td>
<td>4.74</td>
</tr>
<tr>
<td>7. Not enough nodes because of policy restrictions or existing resource reservations</td>
<td>✓</td>
<td>4.37</td>
</tr>
<tr>
<td>8. No Possible Mapping for X: No physical nodes have feature Y</td>
<td>✓</td>
<td>3.54</td>
</tr>
<tr>
<td>9. Insufficient Nodes: Unexplained</td>
<td>✗</td>
<td>3.39</td>
</tr>
<tr>
<td>10. Fixed physical node X not available.</td>
<td>✓</td>
<td>2.56</td>
</tr>
</tbody>
</table>
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
Candidate: \( x = \{\text{utah}, \text{m400}, \text{pc}, \text{ubuntu16-64-ARM}\} \)

Groups – whitelists of acceptable combinations with \( \geq 2 \) properties

Evaluation – Boolean *product of sums*

<table>
<thead>
<tr>
<th>Group relating site, hardware, and type:</th>
<th>Group relating hardware and image:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1(x) = {\text{utah, m510, xen}} \subseteq x )</td>
<td>( b_1(x) = {\text{m400, ubuntu16-64-ARM}} \subseteq x )</td>
</tr>
<tr>
<td>( a_2(x) = {\text{utah, m400, pc}} \subseteq x )</td>
<td>( b_2(x) = {\text{m510, ubuntu16-64-STD}} \subseteq x )</td>
</tr>
<tr>
<td>( a_n(x) = {\text{wisconsin, c220g2, pc}} \subseteq x )</td>
<td>( b_m(x) = {\text{c220g2, fbsd110-64-STD}} \subseteq x )</td>
</tr>
</tbody>
</table>

\[
A(x) = a_1(x) \lor a_2(x) \lor \ldots \lor a_n(x)
\]

\[
B(x) = b_1(x) \lor b_2(x) \lor \ldots \lor b_m(x)
\]

\[
a_2(x) \land b_1(x) = 1 \rightarrow \text{candidate passes the check}
\]
Candidate: $x = \{\text{utah, m400, pc, ubuntu16-64-ARM}\}$

**Constraints**

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

$$a_2(x) \land b_1(x) = 1 \implies \text{candidate passes the check}$$
Candidate: \( x = \{ \text{utah, m400, pc, ubuntu16-64-ARM} \} \)

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

\[ a_2(x) \land b_1(x) = 1 \rightarrow \text{candidate passes the check} \]
In Conclusion

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

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

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

CloudLab

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
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