www. chameleoncloud.org

CHAMELEON: OPERATIONAL LESSONS

Kate Keahey, Jason Anderson (ANL, UC)

Paul Ruth (RENCI), Jacob Colleran (UC, ANL), Cody Hammock (TACC), Joe Stubbs (TACC), Zhuo Zhen (UC, ANL)

{keahey, jasonanderson}@uchicago.edu

July 29, 2019
HARC Workshop, Chicago, IL

CHAMELEON IN A NUTSHELL

- We like to change: testbed that adapts itself to your experimental needs
  - Deep reconfigurability (bare metal) and isolation (CHI) – but also ease of use (KVM)
  - CHI: power on/off, reboot, custom kernel, serial console access, etc.
- We want to be all things to all people: balancing large-scale and diverse
  - Large-scale: ~large homogenous partition (~15,000 cores), 5 PB of storage distributed over 2 sites (now +1!) connected with 100G network…
  - …and diverse: ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
- Cloud on cloud: leveraging mainstream cloud technologies
  - Powered by OpenStack with bare metal reconfiguration (Ironic) + "special sauce"
  - Chameleon team contribution recognized as official OpenStack component
- We live to serve: open, production testbed for Computer Science Research
  - Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  - Currently 3,000+ users, 500+ projects, 100+ institutions

CLOUDS VERSUS HPC RESOURCES

- Traditional HPC resources: interfaces, complexity, efficiency&cost
- Clouds: interfaces, complexity, efficiency&cost
- Differences in complexity:
  - Operational complexity: networking, security, and others
  - Greater sharing of artifacts: appliance management
  - Relative immaturity of the paradigm
- Cloud systems are more complex because they solve a more complex problem

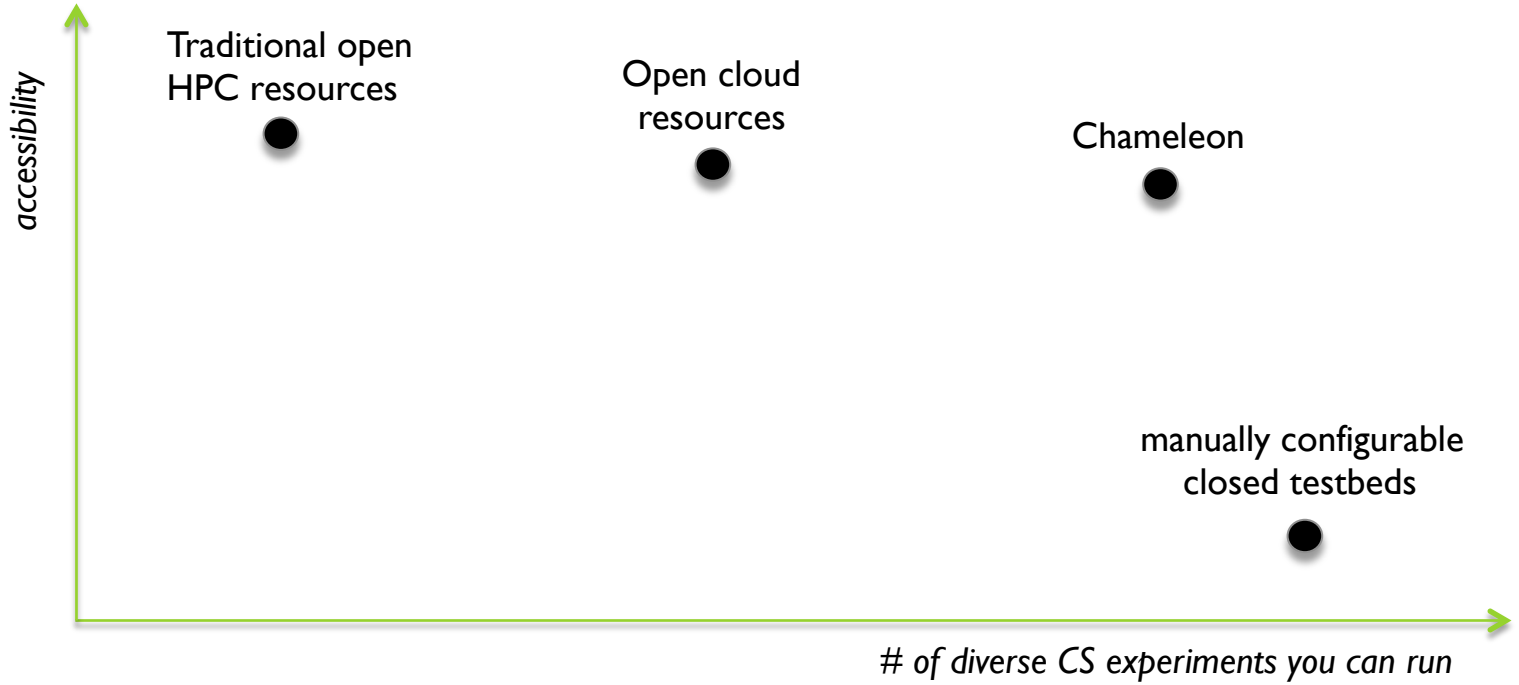
EXPERIMENTAL INSTRUMENTS VERSUS CLOUDS

Chasing the Research Frontier and Adaptation
Emphasis on development/adding new features, closer collaboration with user community

Networking
Access to L2 for all, complexity/automation, integration with commercial offerings

Bare-Metal Infrastructures
security, fewer layers of abstraction, relative immaturity of infrastructure

… AND IT NEEDS TO SCALE



WHAT DOES IT MEAN TO PEOPLE?

- Operators
  - Very high level of skill: more diverse and deeper expertise
  - Significant learning curve
  - Teams of operators with different specialties
  - Development experience is critical
- More effort
  - Many moving parts, immature parts, new parts, unexpected parts
- Close interaction with user community
  - Users are increasingly less **customers** and increasingly more **partners**

HELPING HUMANS IN THE LOOP

- Researchers and instructors (users)
  - Make interfaces to cloud more intuitive (or at least similar to commercial clouds)
  - Facilitate creation of ecosystem for sharing knowledge
  - Direct instruction and guidance
- Host institutions, service providers (operators)
  - Reduce cost of running Chameleon as low as possible
  - Enable plugging in to existing ecosystems
- Ourselves
  - Enable team members of variable expertise to be productive
  - Give insight into usage and health of Chameleon
  - Add force-multipliers to make team have outsized impact

MONITORING: THREE PILLARS

Quantify

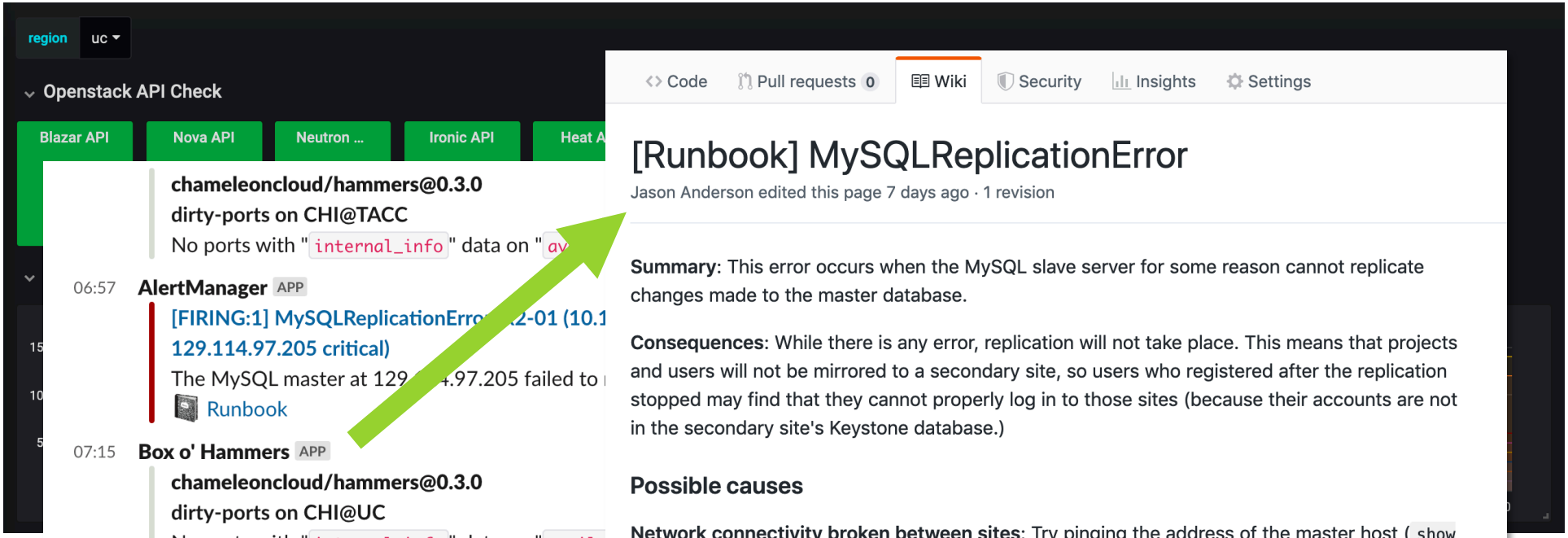
- Symptom-based metrics
  - Prometheus
  - Chameleon-specific metrics
- Log indexing and search
  - Elasticsearch, Fluentd
  - Kolla-Ansible

Detect

- Metric-based alerts
  - Prometheus, Alertmanager
- ”Black-box” probes
  - Periodic checks for external connectivity to public APIs
- “Smoke tests”
  - Suite of Jenkins tests, run nightly (expensive) or hourly (cheap)
  - Checks “happy path” through system

React

- Runbooks
  - Documentation of known errors and mitigations (for operators)
  - Helps new team members be productive
- Hammers
  - Automated solutions for known errors



Code | Pull requests 0 | Wiki | Security | Insights | Settings

[Runbook] MySQLReplicationError

Jason Anderson edited this page 7 days ago · 1 revision

Summary: This error occurs when the MySQL slave server for some reason cannot replicate changes made to the master database.

Consequences: While there is any error, replication will not take place. This means that projects and users will not be mirrored to a secondary site, so users who registered after the replication stopped may find that they cannot properly log in to those sites (because their accounts are not in the secondary site's Keystone database.)

Possible causes

Network connectivity broken between sites: Try pinging the address of the master host (`show slave status\G`). If network works, it could have been broken temporarily. This has in the past caused the slave to stop and not recover. Restarting the slave process can often fix this: `stop slave; start slave;` . Use `show slave status\G` to verify any fix.

Data mistakenly written to slave Keystone database: In this case, data was accidentally added to the Keystone database, which can often cause a duplicate key problem when the master database tries to replicate over a row that already exists. To solve this problem you must reset the replication. You will need access to both the MySQL master and the slave to accomplish this.

IS IT WORTH THE TIME TO AUTOMATE?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

HOW MUCH TIME YOU SHAVE OFF	HOW OFTEN YOU DO THE TASK: 50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS

AUTOMATION

- New appliance releases
  - No “snowflake” images: expressed in code and built with diskimage-builder
- New system releases
  - Patches are tested, built in to a new (Docker) container, then pushed to a local registry for release
  - A job is triggered to deploy new container version using controlled process (Kolla-Ansible)
  - Nobody has to learn how to build/install packages! Downside: somebody should know how to fix problems with pipeline.
- Maintenance processes
  - Taking node out of production, attaching metadata for operators

PACKAGING CHAMELEON

- What is CHI-in-a-Box?
  - Install Chameleon on your own infrastructure with set of provisioning scripts + software bundles

Traditional software

OpenStack

CHI-IN-A-BOX USE CASES

- Chameleon Associate
  - Resources added directly to Chameleon, while retaining project identity
  - Chameleon provides user management (and user support!), resource discovery and appliance catalog
  - Jointly maintained by Chameleon staff and associate site partnership
- Chameleon Part-time Associate
  - Similar to above, but all resources are expected to be taken offline at times
- Independent Testbed
  - Associate site deploys Chameleon, but operates user management and support themselves.
  - First site already deployed at NU

PACKAGING: OUR APPROACH

- Distribute as set of provisioning scripts, build on commodity technology
  - Kolla, Kolla-Ansible, Ansible
  - All infrastructure expressed as versioned code. Infrastructure can be built from scratch repeatably (good for disaster recovery.)
- Provide installation and support documentation
  - Install guide, troubleshooting, runbooks
- We “dogfood” CHI-in-a-Box internally
  - Being consumers of our own product improves quality
- Focus on reducing coupling between sites for reliability

USERS: THE FINAL FRONTIER

- Tickets vs. support lists
  - Dedicated, trackable communication versus discoverable, noisy communications
- Covenant between users and operators
  - Everything works better when users are educated about “proper use”
- Education and outreach
  - OpenStack documentation is mixed blessing
  - Chameleon docs are updated with each release
  - Live webinars and face-to-face meetups most impactful
- Incentivizing an **ecosystem**
  - Sharing is much more powerful in the cloud!

PARTING THOUGHTS

- Physical environment: Chameleon is a rapidly evolving experimental platform
  - From “adapts to the needs of your experiment”…
  - … to “adapts to the changing research frontier”
- Clouds are hard to operate because they solve a complex problem – and experimental facilities even more so
  - More skilled personnel, more effort – and especially in testbeds more development
- Towards an ecosystem: a meeting place of users sharing resources and research
  - Testbeds are more than just experimental platforms: common/shared platform is a “common denominator” that can eliminate much complexity that goes into systematic experimentation, sharing, and reproducibility
  - Working with other operators via CHI-in-a-Box and BYOH initiatives
  - Working with users via providing sharing mechanisms and fostering community development

CHAMELEON:

WE'RE HERE TO CHANGE

www.chameleoncloud.org