Improving Condor configuration management with Wallaby

William C. Benton and Robert H. Rati
Red Hat, Inc.
Forecast

- Background
- Introducing Wallaby
- How it works
- Availability
Why?
```bash
$ wc -l condor_config.generic
  2500 condor_config.generic
$ grep = condor_config.generic \n  | grep -vc ^\#
   279
$ grep -c = condor_config.generic
   558
```
I'd like a high-availability schedd....
MASTER_HA_LIST = $(MASTER_HA_LIST), SCHEDD
HA_LOCK_URL = file:$(SPOOL)
VALID_SPOOL_FILES = $(VALID_SPOOL_FILES), SCHEDD.lock
SCHEDD_NAME = schedhost
SCHEDD.QMF_STOREFILE = $(SPOOL)/.schedd_storefile
HA_LOCK_HOLD_TIME = 300
HA_POLL_PERIOD = 60
Other problems

- Dependencies and conflicts
- Restart vs. reconfig vs. full-reconfig
- Managing multiple nodes
- Configuration versioning, backup, and rollback
Introducing Wallaby
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Wallaby as a service

- Inspect and manipulate features, nodes, groups, parameters, and subsystems
- API access from many languages (including C++, Python, and Ruby)
- Two flavors of command-line tools
- Simple client library for quick scripts
Feature relationships

\[ \text{feature}_A \rightarrow \text{FOO} = \text{val}_a \]

\[ \text{feature}_A \rightarrow \text{BAR} = \text{val}_b \]

\[ \text{feature}_A \rightarrow \text{BLAH} = \text{val}_c \]
Feature relationships

feature\textsubscript{B} \rightarrow feature\textsubscript{A} \rightarrow FOO=val\textsubscript{a}

feature\textsubscript{C} \rightarrow feature\textsubscript{A} \rightarrow BAR=val\textsubscript{b}

BLAH=val\textsubscript{c}
Feature relationships

- feature_A
- feature_B
- feature_C
- feature_D

Values:
- FOO=\text{val}_a
- BAR=\text{val}_b
- BLAH=\text{val}_c
Feature inheritance

“Windows execute node”

“Low-latency execute node”

“Low-latency Windows execute node”
Feature inheritance

“Windows execute node”

“Low-latency execute node”

“Low-latency Windows execute node”
Feature inheritance

“Windows execute node”

“Low-latency execute node”

“Low-latency Windows execute node”

$\text{FOO} = val_a$
Configuring groups

group$_A$ → EC2 Enhanced
   Scheduler
   Execute node
   Master
Configuring groups

Groups have a list of enabled features; their parameters are merged into the group config in inverse priority order.
Configuring nodes

node.local.

node.local.

Cloud submitters

Execute nodes

default
Configuring nodes

Nodes have a list of group memberships; their configs are merged to the node config in inverse order. Wallaby validates configurations at the node level.
Configuring nodes

There are two special kinds of groups: *identity groups*, which contain only one node, and the *default group*, which is applied to every node at the lowest priority.
Versioning and backup

- Config files are good candidates for traditional version control tools, but not everyone is willing to use these!

- Wallaby includes version control; you can snapshot the state of the service at any time and roll back if you need to.
Forecast

- Background
- Introducing Wallaby
- Wallaby’s design and architecture
- Availability
How it works
High-level architecture

- A pool has one *Wallaby server*
- Each node has a *config daemon*
- Clients use the *Wallaby API*
- Client-server communication occurs via the *QMF management bus*
Sidebar: QMF

- AMQP is an open standard for enterprise messaging; Qpid is a scalable implementation of AMQP
- QMF is a management framework built on Qpid messaging
- (see http://qpid.apache.org/ for more)
1. Inspect, modify, and save group config via Wallaby API
1b. (Correct any reported errors or inconsistencies.)
2. Activate configuration

Config tools

Wallaby backend (DB)

Wallaby interface (QMF)

Condor node
Config tools

3. Condor restart event sent to configd processes on affected nodes

Wallaby interface (QMF)

Wallaby backend (DB)

Condor node
4. Config daemons on affected nodes check in with Wallaby using the QMF API
4b. (All nodes check in periodically, to ensure that they have updated configurations)
5. Nodes receive updated configurations

- Wallaby interface (QMF)
- Wallaby backend (DB)
- Condor node

Config tools
6. Config daemons store new config files in `${LOCAL_CONFIG_DIR}`

7. Config daemons issue `restart` or `reconfig` commands to Condor daemons as necessary
Using the Wallaby API

- Interactive and batch command-line tools for pool management
- Example specialized clients: a node inventory tool, a store serializer, and a config file importer
- $(YOUR_IDEA HERE) via client lib
Availability
Wallaby is a service that manages semantically meaningful, versioned configurations for even the largest Condor pools.
Availability

• Wallaby and its dependencies are open-source. If you’re comfortable installing cutting-edge libraries and tools, we’d love your feedback!

• Packaged versions of Wallaby will likely be available in future Red Hat and Fedora Project offerings.
Thanks!

• For more information:
  http://getwallaby.com/
  http://www.redhat.com/mrg/

• Ask us for a demo at Condor Week

• Contact us via email:
  {willb, rrati}@redhat.com