# QuickSilver: A Distributed Policy Engine for Lustre

Chris Brumgard

brumgardcd@ornl.gov

LUG 2022

## Problem

- At ORNL, filesystems are becoming increasing complex to accommodate the needs for faster storage as well as larger storage.
  - Tiers
- Beyond just tiering, admins want an easy and reliable way to implement different policies for different users/groups.
  - Purging
  - Telemetry and querying

## Problem

- Users need both fast storage and large capacity storage.
  - Solution: We'll give them two storage stores: SSDs and hard drives.
- Oh wait!!! Users are terrible at managing multiple data stores.
  - Solution: Okay fine, well just put the different pools under Lustre and let Lustre be the unified namespace.
- Oh wait!!! Users will have to set their stripe data to use the correct pools.
  - Solution: We'll just set the default pools to be NVMe.
- Oh wait!!! Users will have to remember to migrate and purge their data using lfs_migrate and such tools.
  - Solution: ???

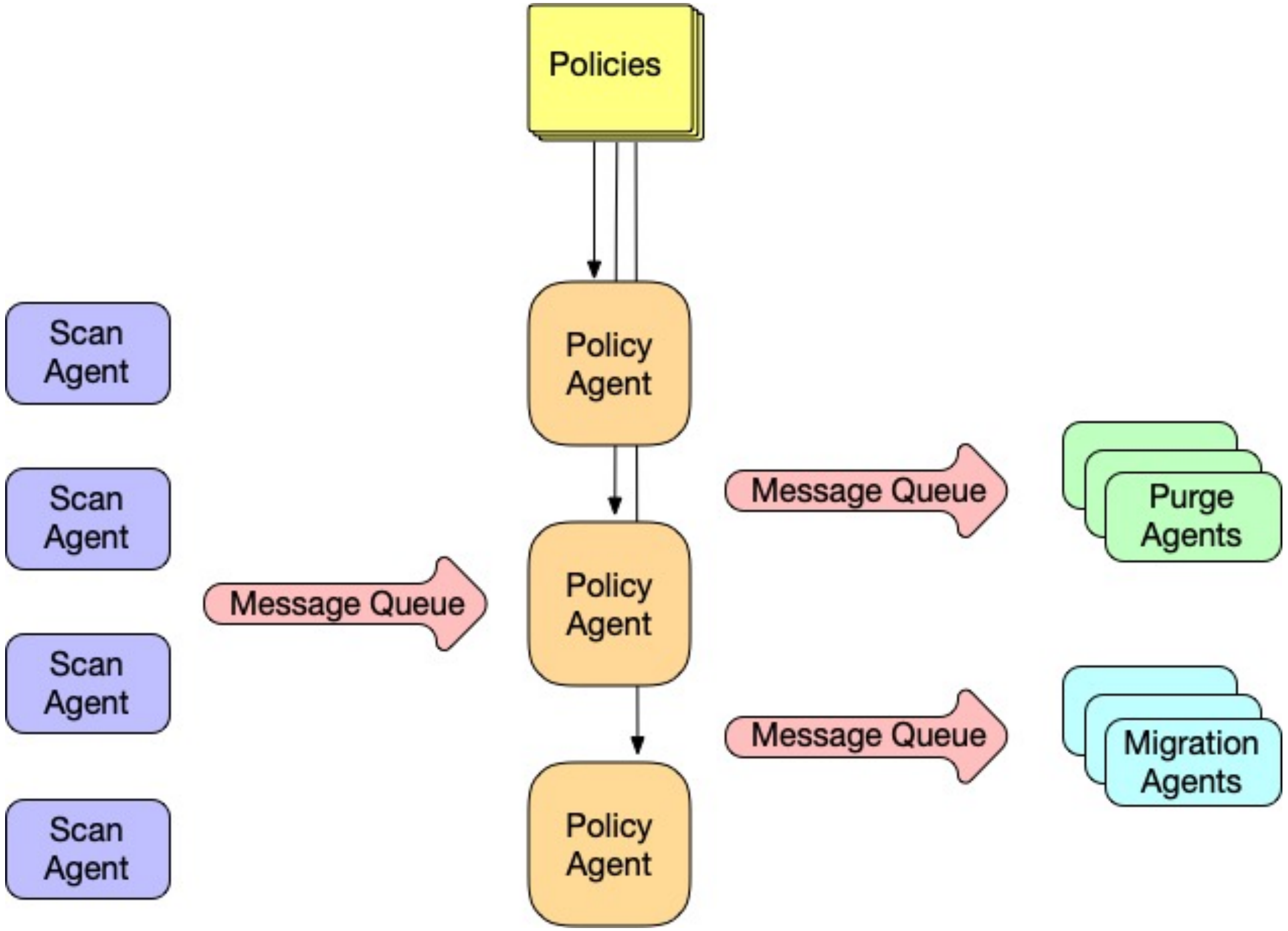
# Enter QuickSilver

- A Distributed Policy Engine for Lustre.
- Purposes:
  - Tiering writeback
  - Purging
  - Data collection and telemetry
- What do we mean by distributed?
  - Actor model
  - Message passing
  - State-lite

## QuickSilver

- Actor model
  - Private state (no shared memory) and light-weight
  - Each actor type does only one particular task.
  - Communicate via messages and process one message at a time.
  - Asynchronous to each other (respond to messages they receive)
- Fault Tolerance & Scalability
  - Passing much of the burden to the messaging system and Lustre.
  - Supervisors to launch and monitor actors.
  - Numerous instances of each actor type (scalable).
  - Raft Protocol for leader election within certain actor types.
  - Designed so that tasks can be lost without affecting the overall system.

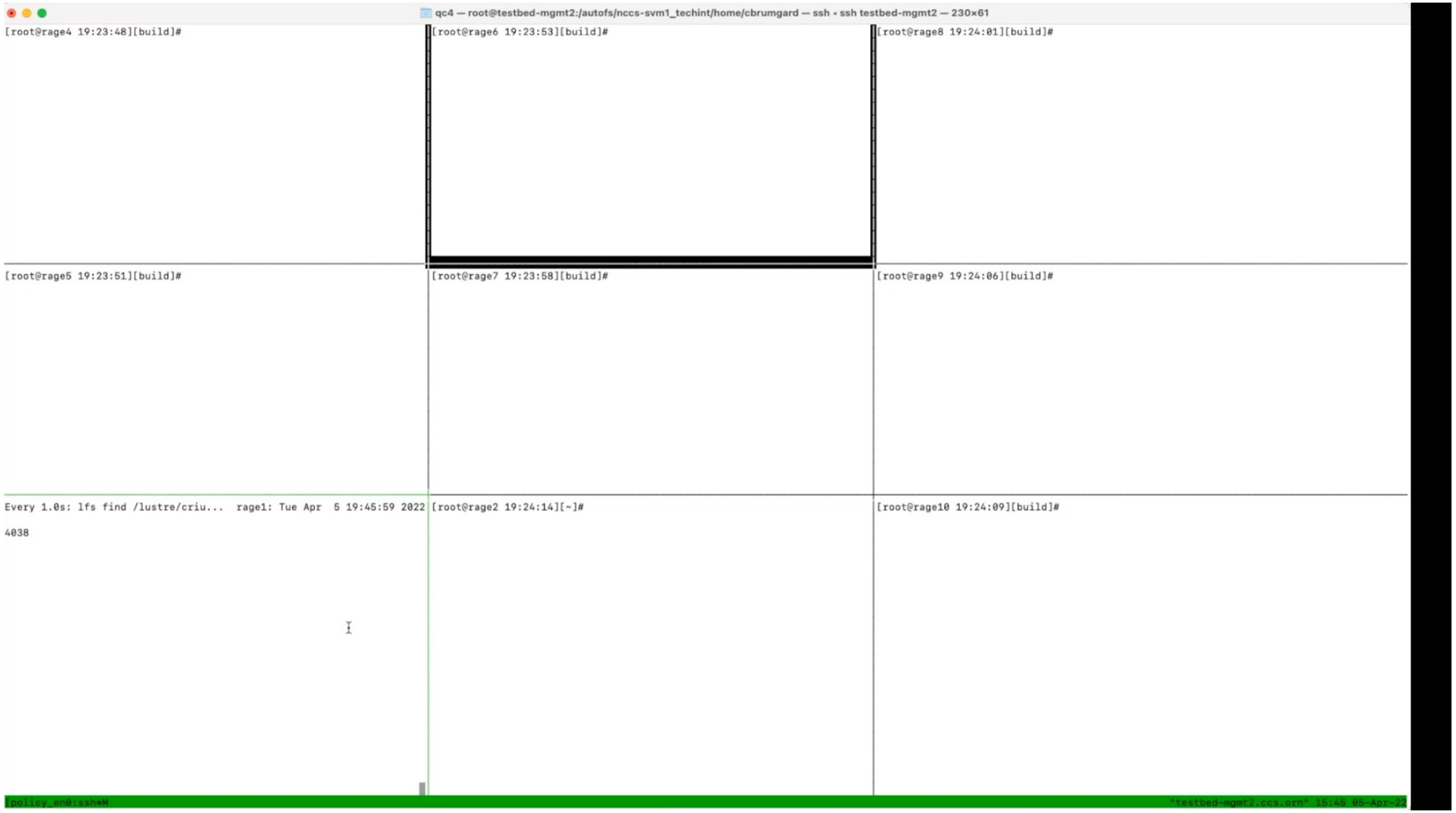
## Quick Silver diagram



## How is QuickSilver different?

- No database, the file system is our database
- No replicated state (except for some key data items like leader election)
- Best effort
  - If tasks fail, that’s okay. We’ll get them next time.
  - Actors aren’t tracking the progress of other actors and waiting on results.
- Highly scalable
  - Need more performance, add more agents of the corresponding type.

# Demo



qc4 — root@testbed-mgmt2:/autofs/nccs-svm1_techint/home/cbrumgard — ssh • ssh testbed-mgmt2 — 230×61

```
[root@rage4 19:23:48][build]#
```

```
[root@rage6 19:23:53][build]#
```

```
[root@rage8 19:24:01][build]#
```

```
[root@rage5 19:23:51][build]#
```

```
[root@rage7 19:23:58][build]#
```

```
[root@rage9 19:24:06][build]#
```

```
Every 1.0s: lfs find /lustre/criu...  rage1: Tue Apr  5 19:45:59 2022

4038
```

```
[root@rage2 19:24:14][~]#
```

```
[root@rage10 19:24:09][build]#
```

```
[policy_en0:ssh*M                                    "testbed-mgmt2.ccs.orn" 15:45 05-Apr-22
```

Open slide master to edit

## Future work

- Still in active development.
  - Reducing the scan work.
  - Productionizing.
  - Deploying to systems this Summer.
- More tiering capabilities.
  - Moving data back to the performance tier
- More complex policies.
  - Decomposing complex actions into simpler rules.
- Non-Lustre agents.
  - HPSS
  - Edge

## Acknowledgments

- Thanks to my fellow developers Anjus George, Ketan Maheshwari, Rick Mohr, James Simmons.
- This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.