Cluster API scalability - Introducing the In-memory provider

Lennart Jern

28.11.2023

Ensure scale of 1000 clusters – CAPI&Metal3

- Insanely expensive if done on real hardware
- Can we do it some other way?
- Bare Metal Operator has a "test mode" (fake Ironic)
- Workload cluster API?
- Iterative approach – solve the biggest issue and move forward

Faking workload clusters

Step 1

- Run API server + etcd with certificates expected by CAPI
- Just a data store -> Add Nodes and Pods as needed

Step 2

- 1:1 mapping too heavy
- Shared etcd; shared API?

Step 3

- Distribute and run remotely (e.g. 100 API servers per VM)

Collaboration and help from maintainers

Issues:

https://github.com/kubernetes-sigs/cluster-api/issues/8052

https://github.com/kubernetes-sigs/cluster-api/issues/8602

Extremely helpful to get insights from maintainers

Hints and ideas about how to build the test environment

Bug fixes and performance improvements as soon as we discovered them

Results

Enabled scaling to 1000 clusters in reasonable time

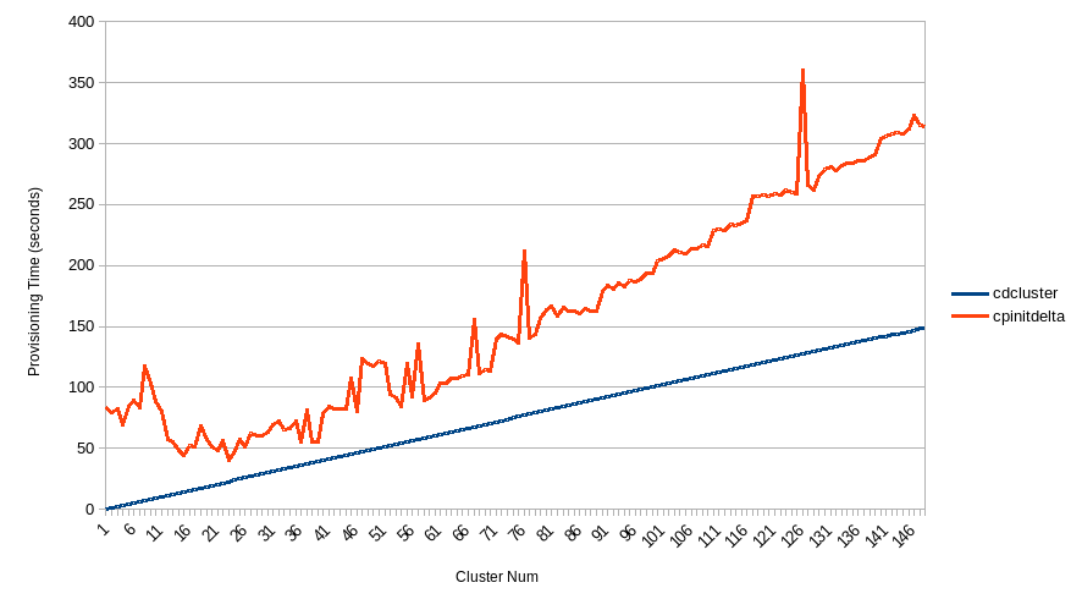
- Initially stopped at ~300 due to concurrency and test env issues
- Progressed to ~600, blocked by rate limits

KubeadmControlPlane controller performance

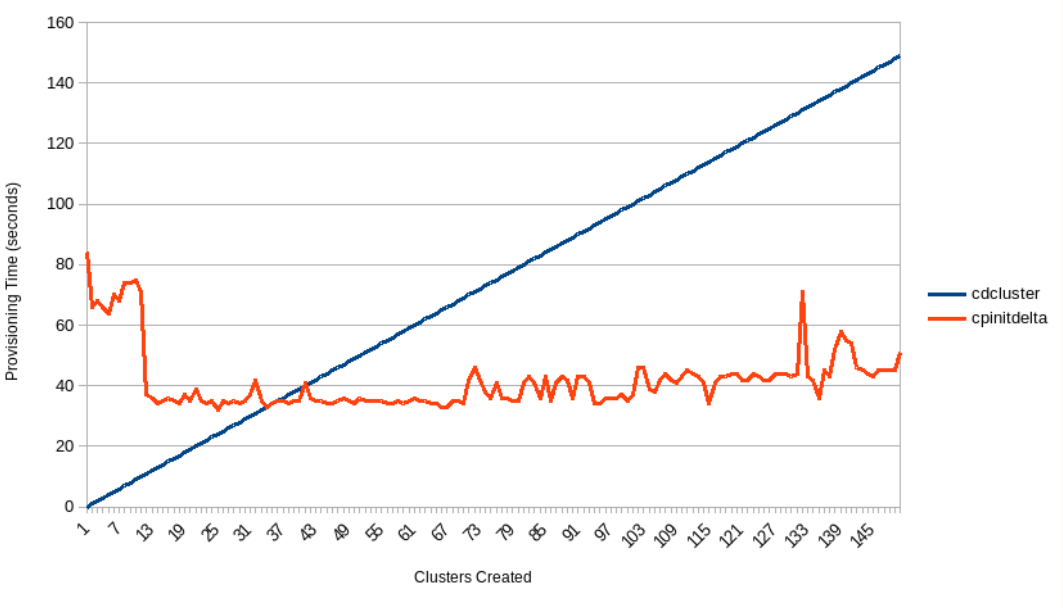
- 80 % reduction in CPU usage
- Cluster creation time reductions

Results

Before

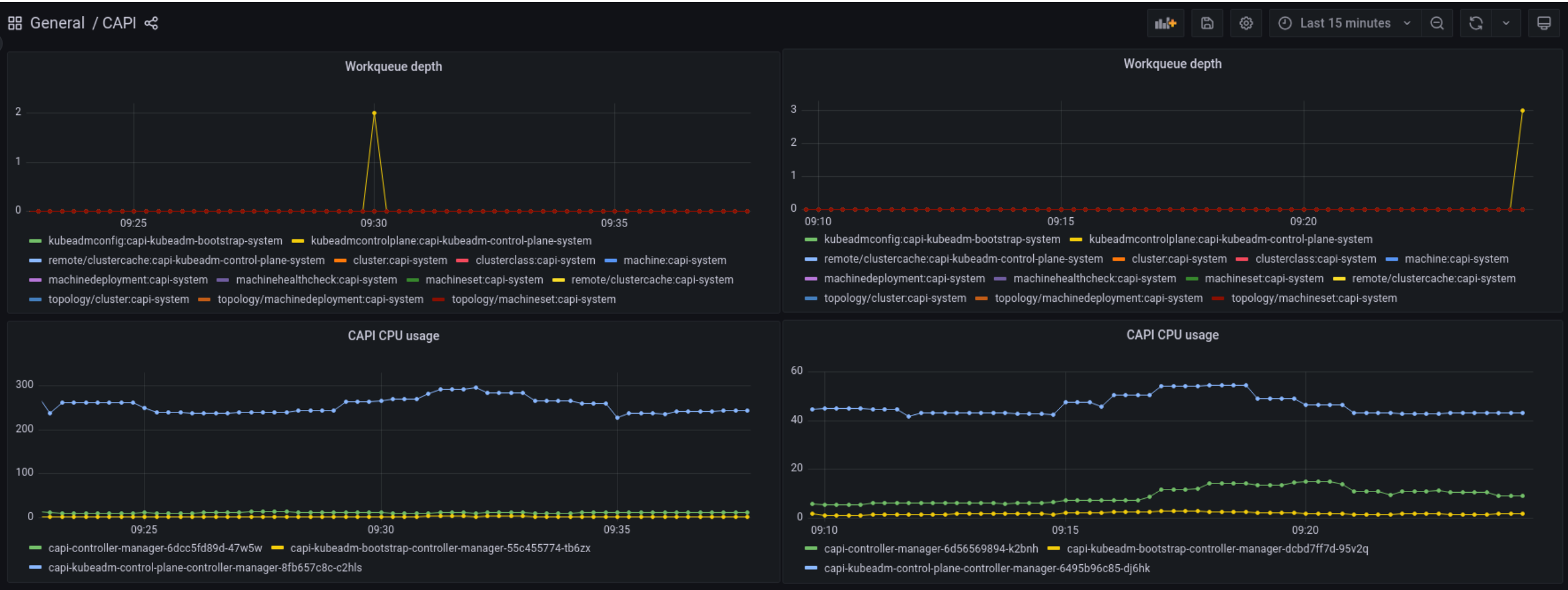


After



Before

After

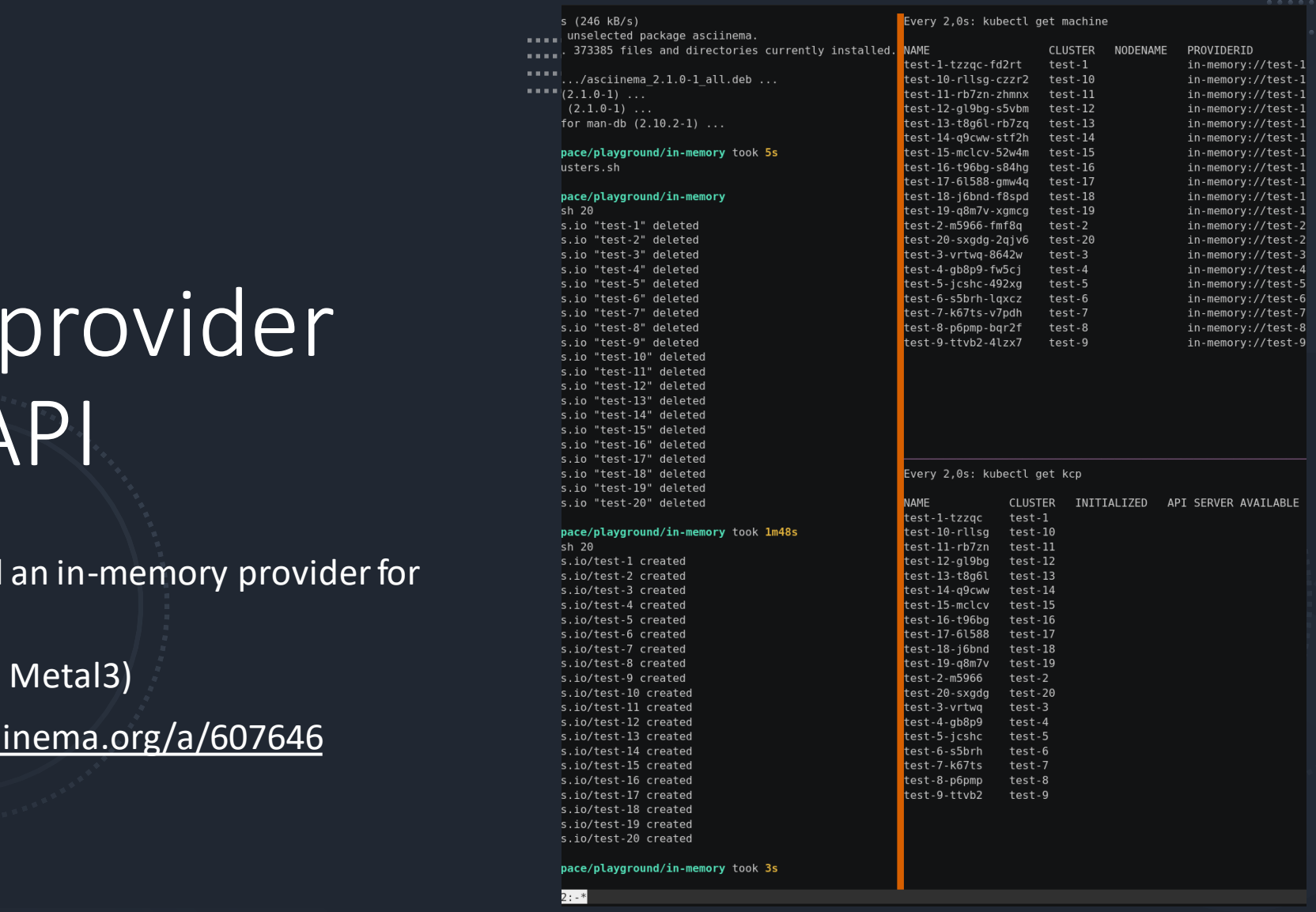


Introducing the In-memory provider

Scale testing CAPI like a Pro

In-memory provider for Cluster API

- Cluster API has implemented an in-memory provider for scaling tests
- Tests CAPI (not providers like Metal3)
- Demo recording: https://asciinema.org/a/607646



Scale testing CAPI like a Pro

- An "infrastructure" provider
- Controller just fakes the minimum necessary
- Each control plane gets an endpoint with unique port
- Each Machine results in a Node object in the API
- A fully operational workload cluster from CAPI perspective

```yaml
---
apiVersion: infrastructure.cluster.x-k8s.io/v1alpha1
kind: InMemoryMachineTemplate
metadata:
  name: in-memory-quick-start-default-worker-machinetemplate
spec:
  template:
    spec:
      behaviour:
        vm:
          provisioning:
            startupDuration: "30s"
            startupJitter: "0.2"
        node:
          provisioning:
            startupDuration: "10s"
            startupJitter: "0.2"
        apiServer:
          provisioning:
            startupDuration: "10s"
            startupJitter: "0.2"
```

Getting started with the in-memory provider

- Recording: https://asciinema.org/a/618386
- Initialize with clusterctl like any other provider
- Check cluster template and cluster class in CAPI releases
- Note that timings can be configured in the InMemoryMachineTemplate

The work goes on

- CAPI efforts to automate scale tests in CI + metrics and observability
- Ensure scalability of full Metal3 stack (Ironic and Bare Metal Operator)
- More details on CAPI work in the KubeCon deep dive 🎉