A Novel Flow Network Graph Based Scheduling Approach in Kubernetes

@kevin-wangzefeng
wangzefeng@huawei.com
Agenda

- Scheduling in K8S
- The Default Scheduler
- Firmament & Poseidon
- Future plans
Scheduling in Kubernetes

apiVersion: v1
kind: Pod
metadata:
  labels:
    run: my-pod
  name: my-pod-76559f5d5b-l9b9p
  namespace: default

spec:
  dnsPolicy: ClusterFirst
  nodeName: node1
  restartPolicy: Always
  schedulerName: default-scheduler
  containers:
The default scheduler

- **Queue based**
  - one pod one time
  - best fit (scheduling time)

- **Resource allocation model**
  - Request based, not real-time usage
  - Low utilization (due to uncertain user resource estimation)

- **Policies implemented as two sets of algorithms:**
  - Predicates
  - Priorities
What is Firmament

• Flow based scheduler
  – Models workloads and cluster as a flow network (DAG)
  – Policies considered at DAG build / update
  – Run Min-Cost Max-Flow (MCMF) solver to find an optimal flow
  – Scheduling results extracted from the optimal flow
• Similar to default scheduler
  – "Global optimal solution"
  – Pluggable scheduling policies

• That makes differences
  – Flexible resource modeling, easy to extend to support topology (zones, racks, NUMA, etc.)
  – Built-in support with rescheduling, priority and preemption
  – And a set of other cost models:
    • network-ware, Quincy, load-spreading etc.
  – Low decision latency at scale
    • sub-second decisions at 10k+ machines
  – batching approach
  – By default use resource utilization instead of reservation
Flow network example in Firmament

- Flow network
  - 4 machine cluster, 2 jobs (3 tasks and 2 tasks).
- Arc labels show non-zero costs
  - (values depends on policies.)
- All arcs have unit capacity
  - except those between unscheduled aggregators and the sink.
- The red arcs carry flow and form the min-cost solution.
  - All tasks except T0,1 are scheduled on machines.
And Poseidon?

To fill the gaps between K8S and Firmament

- **Different concepts**
  - K8S: workloads, pods
  - Firmament: jobs, tasks

- **Different language**
  - K8S: Golang
  - Firmament: C++

- **Resource Requests v.s. Real-time utilization**
  - K8S: allocate by requests and “unclaimed”
  - Firmament: utilization statistics
Poseidon Design

Kubernetes API server

Node events → Node watcher

Pod events → Pod watcher

Pod bindings → Utilization stats

Scheduler loop

Pod watcher

Pod keyed queue

Node watcher

Node keyed queue

Node worker pool

Pod worker pool

Pod worker pool

Pod worker pool

Utilization stats gRPC service

Firmament gRPC service

Node events

Task events

Schedule tasks

Scheduling decisions

Utilization stats
Flow Network aligned to Kubernetes Concept

Node Affinity
Status and Progress

- Incubating under K8S scheduler SIG
  - [https://github.com/kubernetes-sigs/poseidon](https://github.com/kubernetes-sigs/poseidon)
  - Currently Alpha (v0.3)
  - Support CPU/Memory Cost model
  - Node Affinity/Anti-Affinity
  - Pod Affinity/Anti-Affinity
  - Automation for E2E tests, PR process etc.
  - and more…
30X algorithmic throughput

Algorithmic Throughput

<table>
<thead>
<tr>
<th>No</th>
<th>Nodes</th>
<th>Pods</th>
<th>Poseidon</th>
<th>Default Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>3800</td>
<td>26027</td>
<td>761</td>
</tr>
<tr>
<td>2</td>
<td>400</td>
<td>7600</td>
<td>15200</td>
<td>361</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>11400</td>
<td>12351</td>
<td>265</td>
</tr>
</tbody>
</table>
Future plans

• Under development
  – Max allowed pods for nodes.
  – Taints & Tolerations.
  – Another round of benchmarking for scalabilities, performances.

• Longer future:
  – Transitioning to Metrics server API (Heapster is going to be deprecated).
  – High Availability / Failover for in-memory Firmament/Poseidon processes.
  – Priority Pre-emption support.
  – Gang Scheduling.
  – Resource Utilization benchmark.
  – Better cooperating with the default scheduler. (enhancements on multi-scheduler framework)
  – Checkout https://github.com/kubernetes-sigs/poseidon/issues for more…
Join us!

- **Scheduling SIG**
  - [https://groups.google.com/forum/#!forum/kubernetes-sig-scheduling](https://groups.google.com/forum/#!forum/kubernetes-sig-scheduling)

- **Poseidon Project**
  - [https://github.com/kubernetes-sigs/poseidon](https://github.com/kubernetes-sigs/poseidon)

- **Follow Huawei Container team on WeChat**

![QR Code for WeChat](image-url)
Thank you!
LINUXCON
containercon
CLOUD OPEN

CHINA 中国

THINK OPEN
开放性思维