Mesos Performance Isolation Working Group

Meeting agendas and notes

Official list of Apache Mesos working groups: https://goo.gl/yK66LL
Please feel free to add your topics to upcoming meeting agendas.
Note: Follow-up actions are prefixed TODO: -- when resolved, please update to DONE:.

May 19, 2016

Agenda

-

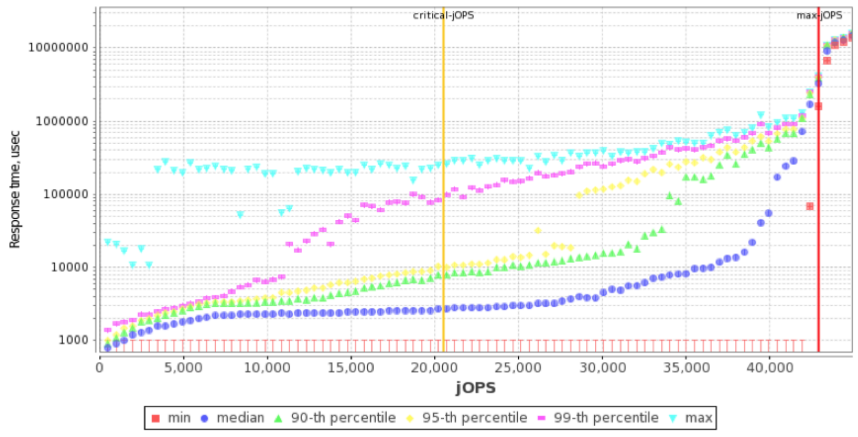
May 12, 2016

Agenda

- For Twitter
    - Be able to use recoverable resources with confidence
    - Benchmarks for burn in. Co-located SpecJBB (normal resource) and stress and others (simulations, scientific benchmarks).
        - ParSec
    - SpecJBB reports latencies
    - Maybe a prototype would be a good place to start
    - Hypothesis: Exclusive CPU sets could mitigate the interference
        - Label or task tier, specify exclusiveness
        - The isolator creates or expands exclusive cpuset
        - Topology clustering
    - (1) Create experiment
        - SpecJBB + ParSec (be)
        - See if exclusive cpusets fixes the problem
    - (2) Create QoS isolator module
        - Selection of cores: matches hyper threaded pairs. Keep on same socket.
    - Lack of CPU affinity lead to high variance while comparing with public cloud environments.

max-jOPS = jOPS passed before the First Failure					
Pass/Fail	Pass	Pass	Fail	Fail	Fail
jOPS	42417	42910	~~43403~~	~~43897~~	~~44390~~

critical-jOPS = Geomean (jOPS @ 10000; 50000; 100000; 200000; 500000; SLAs)						
Response time percentile is 99-th						
SLA (us)	10000	50000	100000	200000	500000	Geomean
jOPS	11097	15043	21455	27867	36744	20553



- ○
- ○ Example SpecJBB output
  - ■ Observation; to stay below SLO non-pinned Mesos cluster performed worse.
  - ■ ParSec as revocable task
  - ■ HP task went from 10 to 6 cores effectively when BE ran next to. Measured in CPU time.
- ○ Executor labels will encode exclusivity
  - ~~■ "Cpu_option"~~
  - ■ "Cpu_hint": "exclusive"
  - ■ Ignore smaller than 1 core (schedule normally)
  - ■ Maybe only accept even number
  - ■ We just emit a warning if people don't get it right.
- ○ Longer term, we want app classes.
- ○ QoS Isolator (delegates)

- Niklas: Talk about next steps and planning
  - ○ Ar((Niklas)): Experiment
  - ○ Ar(Niklas): QoS isolator proposal
  - ○ QoS isolator prototyping
  - ○ Ar(Chris): Do work in Github
    - ■ Collaborators: nik@qni.dk, ct-clmsn, ndwns, MrigankAtGitHub
    - ■ https://github.com/ct-clmsn/mesos branch is called "bind"

April 18, 2016

Agenda

- Current state of core isolation
- CAT support in Mesos

Mar 4, 2016

Ian: Document is here. I've only pulled across some of my own document because I think there's still a lot of questions here…

https://plus.google.com/hangouts/_/twitter.com/performance

Action items:

- TODO: [Niklas] Examine prior art in QoS, e.g., the issues and work being done in Kubernetes.
- TODO: [Ian/Kevin] Document use-cases for per-resource class.
    - We suggest container class ← justify this assertion.
- TODO: [Ian] Follow up with Jie to locate his testing and results on SCHED_IDLE. Determine if it was our configuration that was incorrect? Reproduce results from investigating revocable CPU.
- TODO: [Niklas] Contact appropriate folks for kernel patches.
- TODO: [Ian] `twmemcache`, is the benchmark open source?
- TODO: [Ian] Document (quantify) the JVM interactions with CFS bandwidth quota.

9:00am Introductions / agenda
Attendees:

- **Intel:** Connor, Niklas, Nicholas Weaver
- **Twitter:** Ian
- **Mesosphere:** Kevin, Gilbert
- … sorry, didn't catch others. Please add yourselves

Discuss Ian's updates to exclusive resources proposal

- Should application classes be at the application level or at the individual resource level? E.g., does a latency sensitive application get priority access to all the resources? Or, is it useful to specify this at the resource level?
- Here is the patch for scheduler latency statistics in Mesos: https://reviews.apache.org/r/38074

How to manage dynamic isolation mechanisms for resources? (pushed from last week's meeting)

Current state is that in general, Mesos knows about resources you intend to isolate. A smart set of isolators might want to do dynamic control based on priority, SLO/SLI, etc.

Two cases:

1. Mesos does not know about the isolation mechanism (easy)
    - Cache allocation partitioning
    - Memory bandwidth partitioning
2. Mesos is already managing about the isolation mechanism (harder)

    - CPU shares
    - Memory limits

Issues:
- How to communicate allocation updates back to the master to form accurate offers?
- Is there a need to capture the difference between those two cases?

Options:
1. https://issues.apache.org/jira/browse/MESOS-1279 (Add resize task primitive)
    - In the proposed design doc, TaskStatus gets a new resources field.
    - Allocation adjustments made by the container runtime could trigger a status update?
    - This could get pretty chatty, depending on how active the runtime management is.
2. Turn off the isolators in question you intend to manage (makes Mesos allocations a lie)
    - Some resource types are special
3. In Kubernetes, (request, limit) gives flexibility for local adjustments (burst up to limit), at least in theory
    - This would break many things.

Feb 26, 2016

First official Mesos perf isolation meeting

10:15am Introductions / agenda
Attendees:
- *Intel:* Niklas, Connor
- *Twitter:* Ian
- *Uber:* Ashwin, Zhitao, Michael
- *Apple:* James Peach, Dario
- *Mesosphere:* Kevin, Jie

10:20am Updated proposal for core affinity from Twitter (Ian)
Previously, there was a proposal for exclusive resources from Ian
Comments from Christos: maybe unwise to give access to exclusive resources because it can be abused and hurts utilization
From the Borg paper, there are tiers of service or app classes (latency sensitive, batch, exclusive cpus).
Proposing four labels: 1) latency critical (exclusive cpus) 2) latency sensitive 3) Normal 4) Best effort
DONE(CD): Make the TaskInfo available via the isolator interface so people don't have to use the slave run task hook? (Created issue https://issues.apache.org/jira/browse/MESOS-4816)

Suggestion from James: make it possible to decide which isolators get applied at task launch time. Perhaps add an extra level of indirection to configure first-class groups of isolators for different strategies (discussed previously together with Jie)
TODO: Ask Kapil Arya about the above.
TODO: Look into a way for agents to advertise capabilities (available containerizers, isolators, etc.) (Create an issue)
Ian wants to expand the idea of exclusive resources to network egress bandwidth as well.
CPU affinity could be an easy way to address the CFS problem, but at a high cost to utilization.
For most other people, choosing among two or three other tiers of service should be preferred.
TODO: Create an issue for encoding app classes (start with codifying labels, find path to first-class) *Presumably this needs to be done at the executor level as the unit of containerization?*
DONE: Ian to share document.
Does Twitter intend to use revocable resources? How does it interact with app classes?
A(Ian): best effort tasks only run on revocable resources.
Kevin: does it make sense to add another class called “provisionable”? You have them as long as you wuant them, and yo have preference for those “exact” resources (mainly thinking about CPU cores). *A(Ian): will add a discussion about this to his proposal.*

10:50am CFS configuration status from Intel (Niklas)
Christos Kozyrakis is meeting with kernel people at Intel, Niklas will get more involved in those discussions.

10:55am workload benchmarking
Working on coming up with representative workloads as a driver for landing PQoS controls to help with oversubscription.
Want to make control loops more generally applicable.
Workloads and the testing tools will be open-sourced following approval by Intel legal.
This project could be a good place to collaborate and share use cases.
Twitter indicated they would like to contribute.

Last topic, dynamic isolation controls for resources that are managed / not managed already by Mesos. Ran out of time, will pick this up next meeting.

Ian will try to send out an updated doc next Monday or Tuesday in time to review for Friday March 4th.
