Elastic Memory: Bring Elasticity Back to In-Memory Big Data Analytics

May 18, 2015

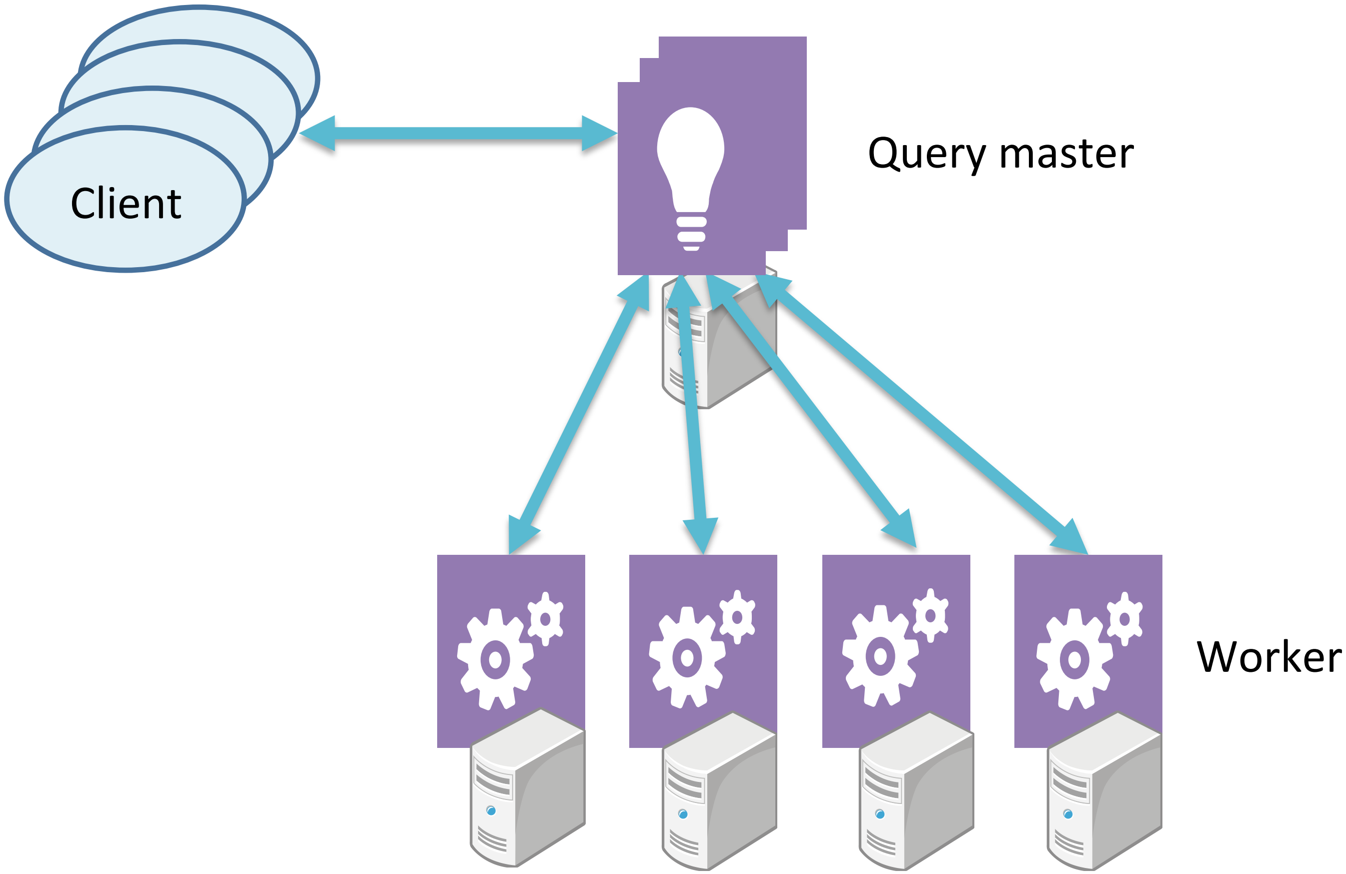
Byung-Gon Chun
Seoul National University
(cmslab.snu.ac.kr)

Joint work with Joo Seong Jeong, Woo-Yeon Lee, Yunseong Lee, Youngseok Yang, Brian Cho

Elastic Big Data Analytics Jobs

- MapReduce/DAG jobs execute on a runtime that supports elastic scale-out execution
- Distinct MapReduce/DAG jobs run together on a shared cluster, thus improving utilization
- New types of in-memory data analytics do not fit well to this model
  - The interactive query system does not share resources even when the system is idle

New Types of In-Memory Data Analytics: Interactive Query

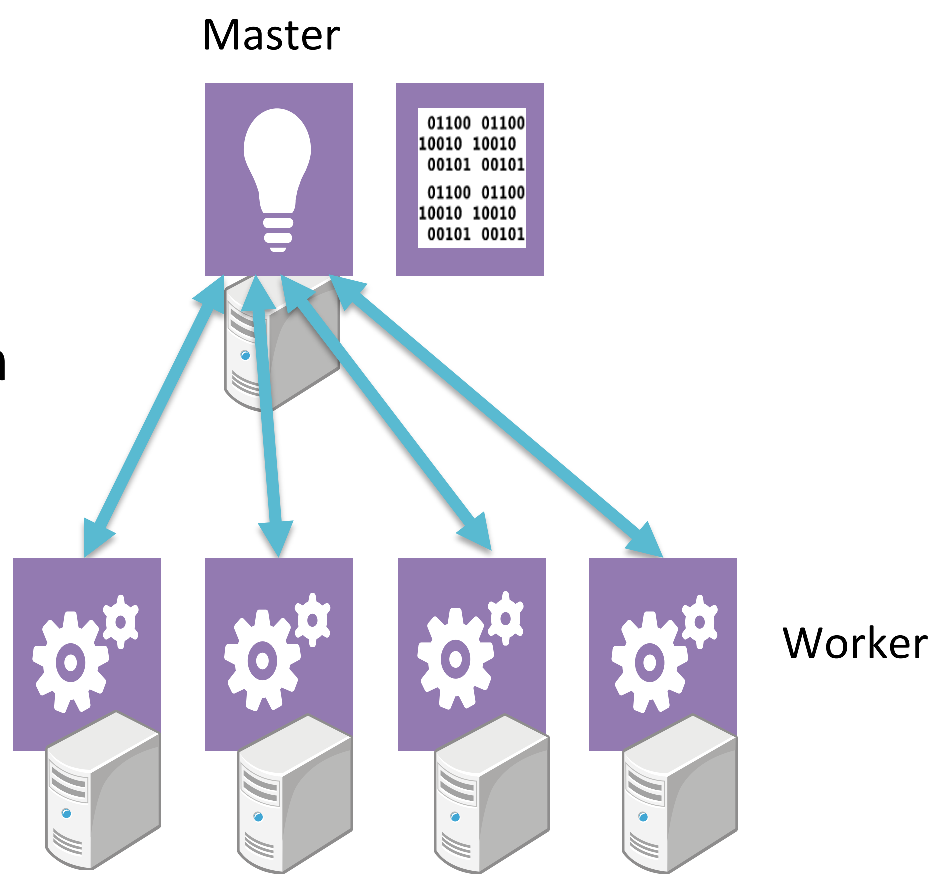


The Case for Elasticity: Interactive Query

- Scale-out
  - The workers may spill data to disks when they do not have enough memory resources => expand memory resources to perform in-memory processing
- Scale-in
  - The workers hold on to their resources even while they remain idle during periods without client queries => shrink resources to mitigate reduced cluster utilization

New Types of In-Memory Data Analytics: Machine Learning

Iterative computation



The Case for Elasticity: Machine Learning

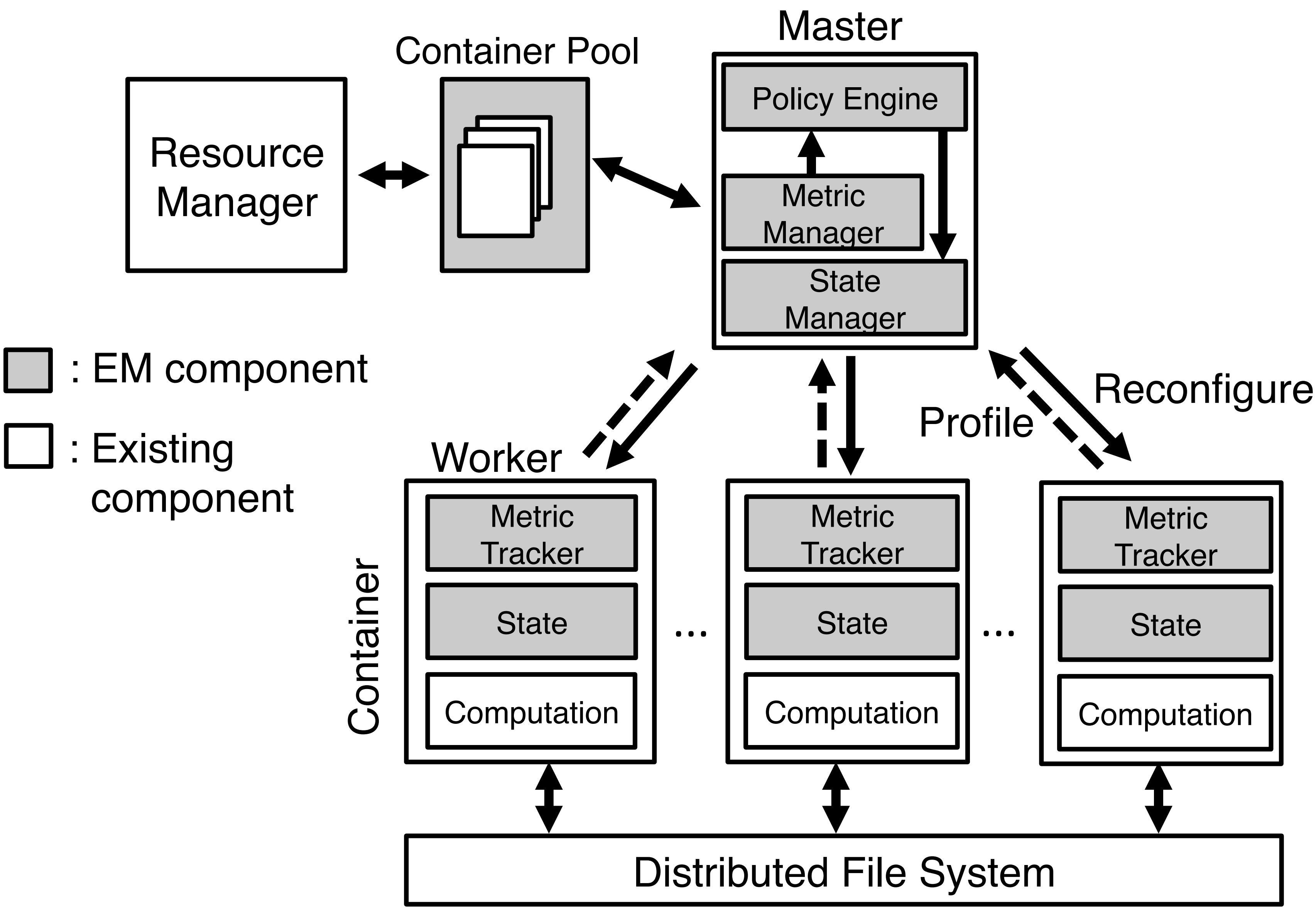
- Scale-in
  - The job is communication heavy => shrink the number of machines to reduce communication overheads

- Scale-out
  - The job is computation heavy => allocate more memory in other machines to exploit computation parallelism

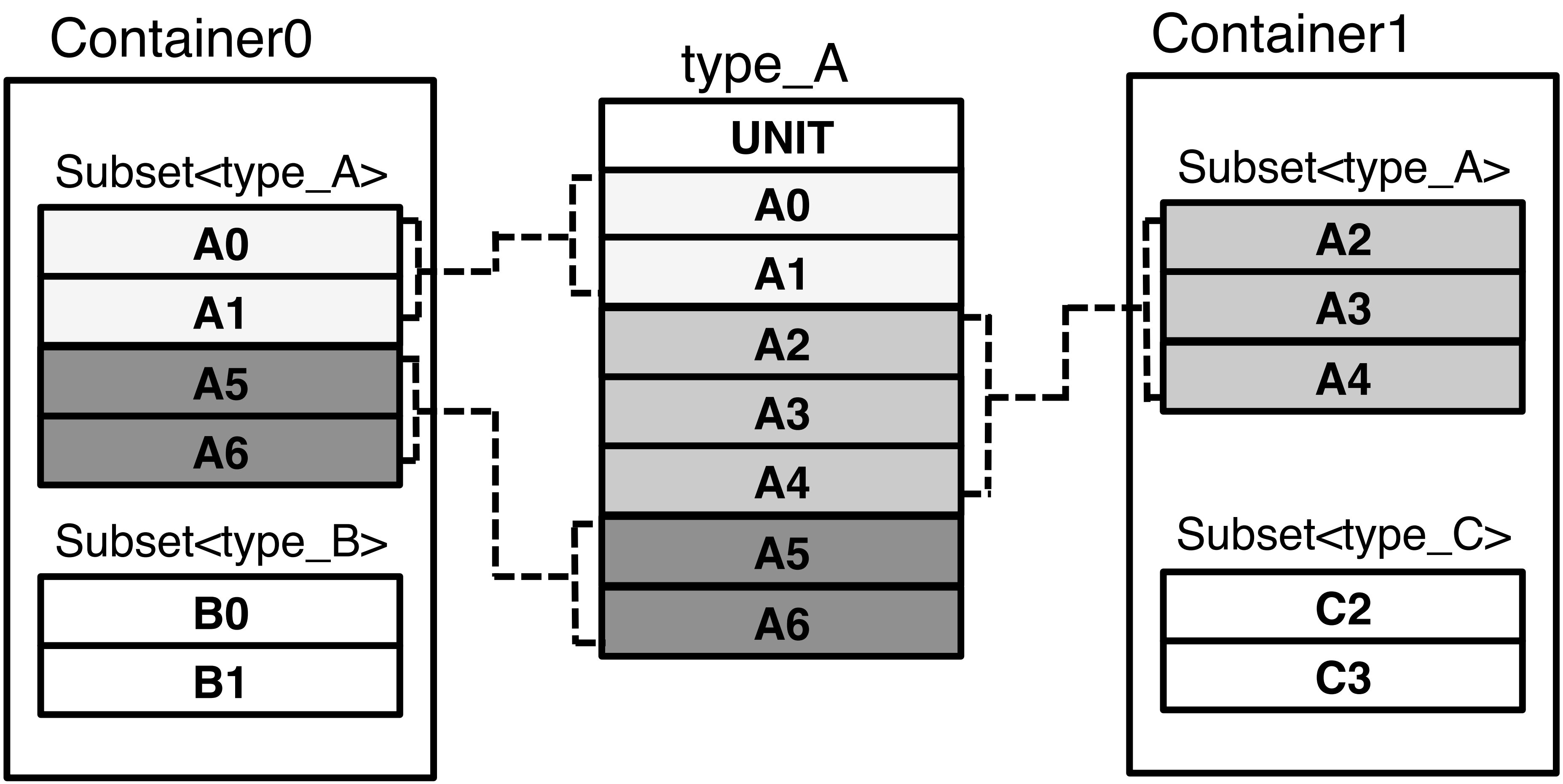
Elastic Memory (EM)

- Abstraction that provides “elastic memory” by dynamically expanding and shrinking memory resources and moving memory state
  - Mechanisms for reconfiguring memory resources and state
  - Policies for automating reconfiguration

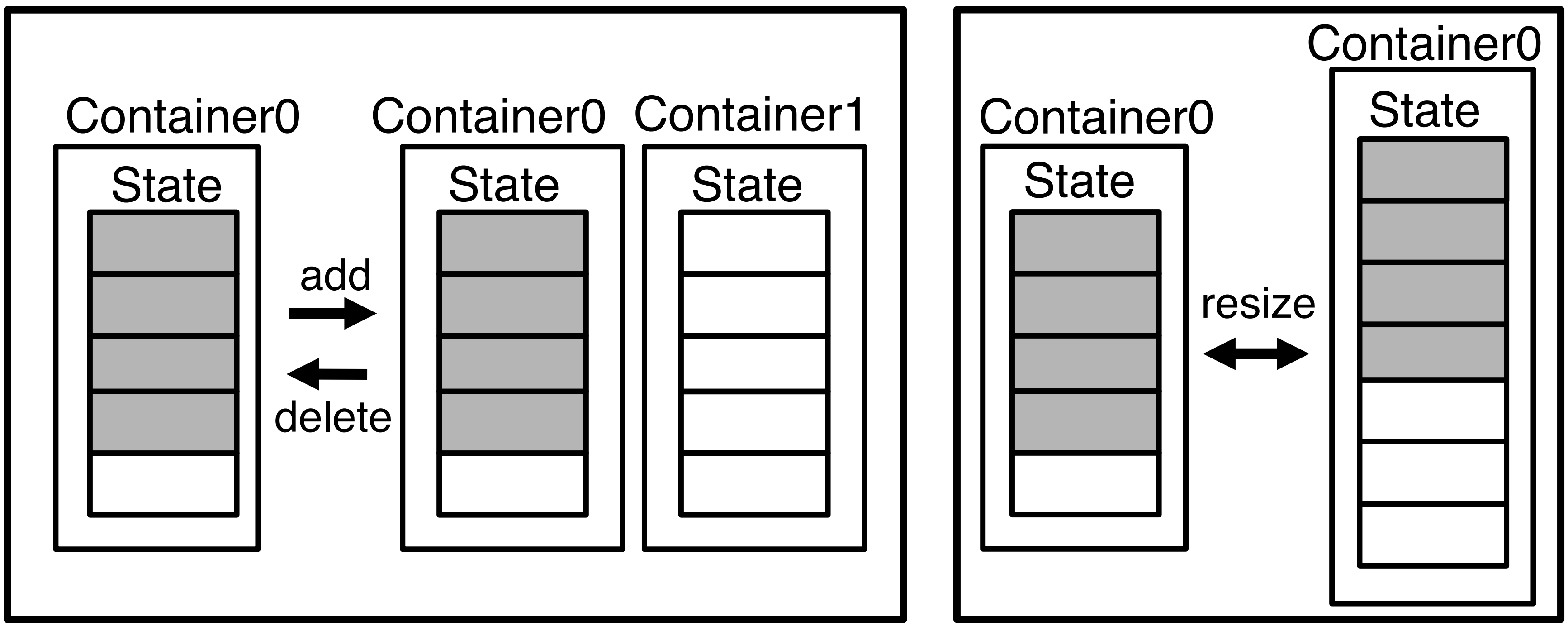
EM Architecture



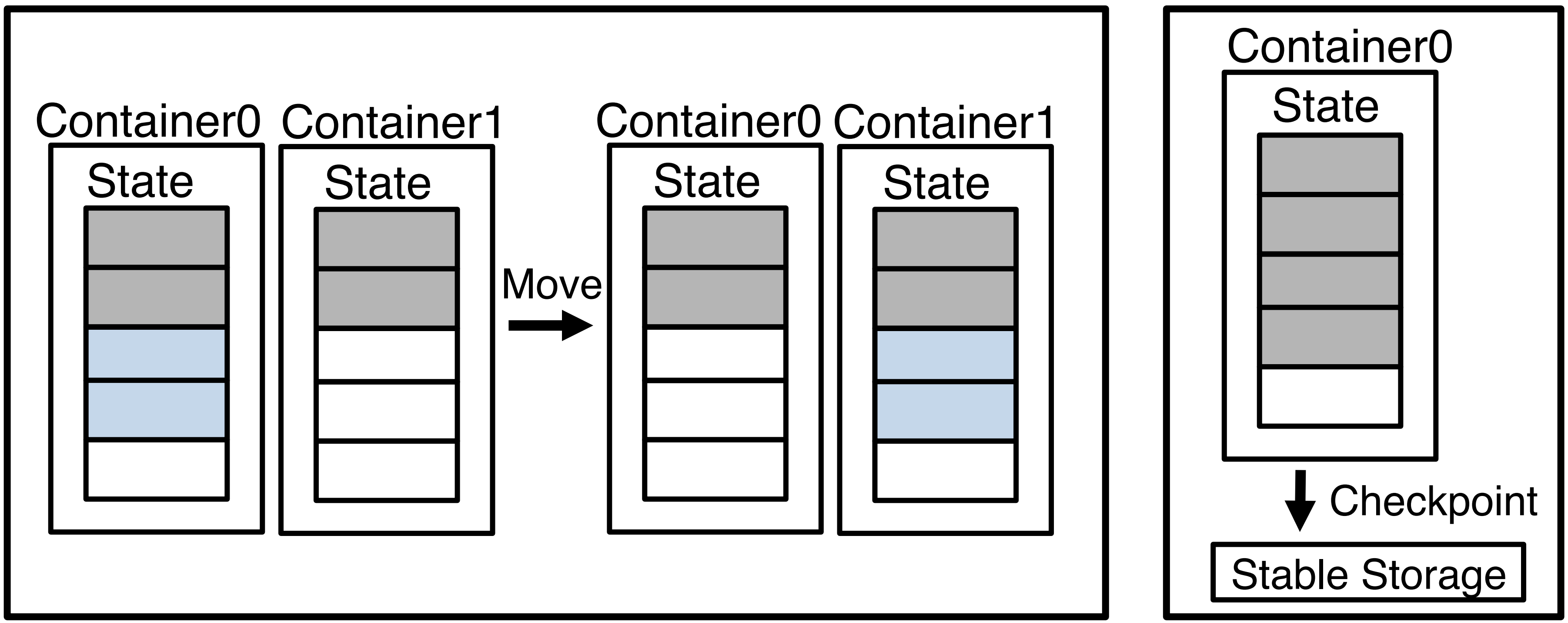
State Representation



Primitives for Reconfiguring State



Primitives for Reconfiguring State



Profiling

- Each worker's metric tracker measures local metrics and sends them to the metric manager

- The metric manager aggregates and processes the received metrics

Policies

- Policy = Rules
- Rule = Condition, Actions
- Condition = Function(metrics)
- Action
  - `Add` <ResourceSpec>
  - `Delete` <SelectFunc>
  - `Resize` <SelectFunc> <ResourceSpec>
  - `Merge` <SelectFunc> <n>
  - `Split` <SelectFunc> <n>
  - `Migrate` <SelectFunc1> <SelectFunc2>

Elastic Interactive Query with EM: Unit, Metrics

- Unit: a row of a table
- Metrics
  - Requests for data per second
  - Memory utilization
  - Idle time
  - …

Elastic Interactive Query with EM: Policy

- Rule 1 (scale out)

  Condition: *avg*(load) > 0.8

  Action: `Add`(resource-spec)

- Rule 2 (scale in)

  Condition: idle-time > 10 min

  Action: `Delete`(*top*(idle-time))

Distributed Machine Learning

- Start by loading data from disk and storing it to memory; access data in memory throughout the job execution
- Iterate
  - The workers run the algorithm independently on its partition of the data
  - The master aggregates the computation results and calculates a model.
  - This model is broadcast to the workers

Elastic Machine Learning with EM: Unit, Metrics

- Unit: an independent observation (e.g., a single number, vector, a matrix)

- Metrics
  - Task time per iteration
  - Computation time per iteration
  - Communication time per iteration
  - …

Elastic Machine Learning with EM: Policy

- Rule1 (straggler handling)

  Condition: *is_straggler*(task-iter-time)
  Action: `Migrate`(top1(task-iter-time), bottom1(task-iter-time)

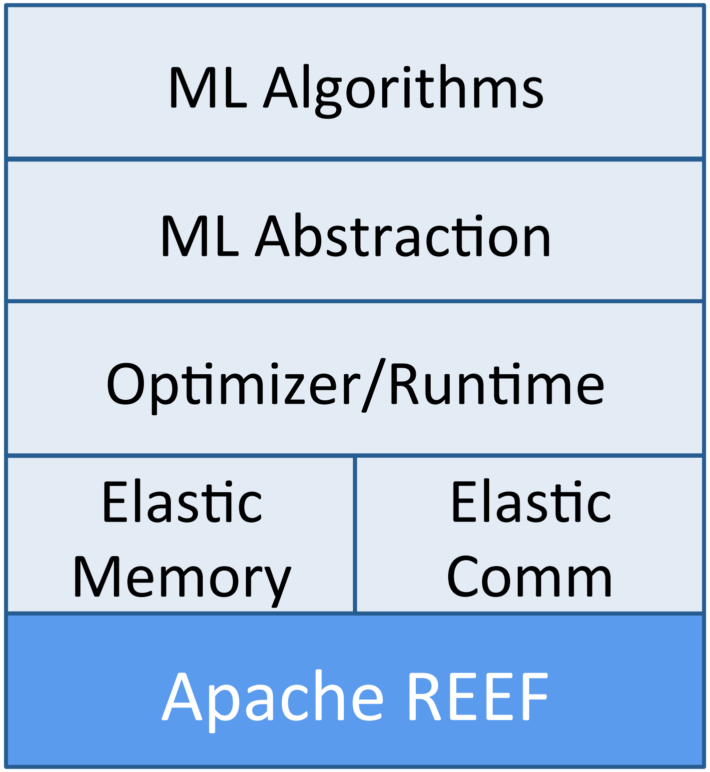
- Rule2 (scale-out)

  Condition: *avg*(task-comp-time/task-comm-time) > TH1
  Action: `Split`(top1(task-comp-time/task-comm-time), 2)

- Rule3 (scale-in)

  Condition: *avg*(task-comp-time/task-comm-time) < TH2
  Action: `Merge`(bottom2(task-comp-time/task-comm-time), 2)

Elastic Machine Learning Framework



Meta-framework
for big data systems

(http://reef.apache.incubator.org,
SIGMOD 2015)

Current Status

- Building Elastic Memory on Apache REEF
- Building a new Elastic Machine Learning Framework that runs on Elastic Memory
- Exploring SparkSQL-like engines to work with Elastic Memory

Thank you!
Q & A