Kairos: Preemptive Data Center Scheduling Without Runtime Estimates

Pamela Delgado, Diego Didona, Florin Dinu and Willy Zwaenepoel

October 11, 2018
Kairos

Data center scheduling
without task runtime estimates
Kairos key idea

- New preemption approach

✓ No head-of-line blocking
✓ Good scheduling performance
Data center scheduling challenge

- Heavy-tailed workloads
Problem: head-of-line blocking

- Short waiting for long
- High likelihood
Historical use of runtime estimates

- **per-task estimations**
  - Apollo’14
  - Yaq’16
  - Tetrisched’16
  - Eagle’16
  - Firmament’16

- **dual classification**
  - Hawk’15
  - Mercury*’15

- **no estimations**
  - Yarn’13
  - Sparrow’13

---

**Motivation**

Do not avoid head-of-line!

Depend on runtime estimates
Hard to obtain reliable estimates

• Mis-estimations happen
  • unseen jobs, skewed input, failures/spikes

• Consequences:
  • poor scheduling decisions*, violate SLOs^
  • complex designs to compensate

*Job-aware scheduling in Eagle: Divide and Stick to Your Probes (SoCC’16)
^Tetrisched: global rescheduling with adaptive plan-ahead in dynamic heterogeneous clusters (Eurosys’16)
Can we dispense with task runtime estimates altogether?
Can we dispense with task runtime estimates altogether?

- Avoid head-of-line blocking
- No task runtime estimates

Kairos
Kairos insight

Use preemption!!
Preemption in Kairos

Costly resuming elsewhere:
Do preemption locally!
Kairos architecture

Centralized component

Distributed component

Local preemption
Kairos architecture

Kairos centralized scheduler

Load balancing

Node

Kairos node scheduler

Node x

Kairos node scheduler

Node y

Kairos node scheduler

Local preemption

Local preemption

Local preemption
Kairos architecture

Kairos: Preemptive Data Center Scheduling Without Runtime Estimates | SoCC’18
Least-Attained Service (LAS)

• Preemptive policy
• Give resources to task that received least service

✓ New task runs immediately
✓ Runs as long as it is the one with least received service
LAS rationale

• Good for heavy-tailed workloads*

• Benefits:
  1. Shorter tasks have priority (no head-of-line blocking)
  2. Shorter tasks –very likely– execute until completion

*Performance modeling and design of computer systems: queueing theory in action M. Harchol-Balter 2013
Kairos distributed scheduling

- Node schedulers
- LAS at the nodes

How to dispatch tasks among nodes?
Kairos architecture

Kairos centralized scheduler

Load balancing

Local preemption

Node j

Node x

Node y
Kairos centralized scheduling

1\textsuperscript{st} Load balancing

2\textsuperscript{nd} Maximize LAS effectiveness
Load balancing rationale

1. Lowest # tasks: no idle nodes
   - Bound max # tasks

1. Avoid!
Load balancing rationale

2. LAS-aware policy break ties:
   • Heavy-tailed for each node
   • Maximize LAS effectiveness
   • Node with lowest AS variance*

*Minimizing total flow time and total completion time with immediate dispatching. Avrahami et.al. 2003
Multi-layered round robin routing for parallel servers Down et.al. 2006
Kairos recap

1. Distributed:
   ✓ LAS node level

2. Centralized:
   ✓ LAS-aware load balancing technique
Evaluation

- Yarn and Docker containers
- 120 cores in 30 nodes
- heavy-tailed workload (100 jobs)
- Metrics: Job runtime and slowdown
- Compare to: Big-C [ATC’17], FIFO
- Simulation: Google trace, compare to Eagle [SoCC’16]
What is the slowdown?

\[
\text{job slowdown} = \frac{\text{observed job runtime}}{\text{uncontended job runtime}}
\]

Best job slowdown = 1
Kairos vs Big-C and FIFO

Job slowdown

CDF

0 2 4 6 8 10 12 14 16 18 20
0 20 40 60 80 100 120

Kairos vs Big-C and FIFO

Kairos: Preemptive Data Center Scheduling Without Runtime Estimates | SoCC’18
Kairos vs Big-C and FIFO

Job slowdown

Slowdown in Kairos <1.8X
Kairos vs Big-C and FIFO

Job running times

- Kairos
- Big-C
- FIFO

CDF

Job runtime [s]

0 500 1000 1500 2000 2500 3000 3500 4000

2X 1.6X 2.3X

better

Kairos: Preemptive Data Center Scheduling Without Runtime Estimates | SoCC’18
Kairos vs Big-C and FIFO

Job running times

Kairos better across the board
Kairos vs Eagle

- Short jobs runtime
- Google trace

![Bar chart comparing Kairos and Eagle with different cluster load levels (lower, higher) and runtime percentiles (50th, 90th, 99th). The chart indicates that Kairos/Eagle performs better in terms of short job runtime.]
Kairos vs Eagle

- Short jobs runtime
- Google trace

Kairos works well at large scale
Why are we better?

Against FIFO
✓ FIFO does not avoid head-of-line

Against Big-C
✓ We do preemption better

Against Eagle
✓ Preemption
More in the paper

- Evaluation with a uniform workload
- Sensitivity to parameters
- Comparison with other load balancing techniques
- How we do preemption

- Soon open sourced
Kairos

✓ First preemptive scheduler without runtime estimates
✓ Smart preemption: good job runtime and slowdown
✓ LAS at node level
✓ LAS-aware load balancing