Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis

Charles Reiss[*], Alexey Tumanov[†], Gregory R. Ganger[†], Randy H. Katz[*], Michael A. Kozuch[‡]

[*] UC Berkeley [†] CMU [‡] Intel Labs

http://www.istc-cc.cmu.edu/

Our goal

Lessons for scheduler designers

Challenges resulting from cluster consolidation

Cluster consolidation

Run all workloads on one cluster

Increased efficiency

- Fill in “gaps” in interactive workload
- Delay batch if interactive demand spikes

Increased flexibility

- Share data between batch and interactive

The Google trace

Released Nov 2011

"make visible many of the scheduling complexities that affect Google's workload"

Challenges motivating second system [Omega]:

- Scale
- Flexibility
- Complexity

The Google trace

Mixed types of workload

Would be separate clusters elsewhere

What **cluster scheduler** sees

Over ten thousand machines, **one month**

No comparable public trace in scale and variety

Background: Common workloads

High-performance/throughput computing:

large, long-lived jobs; often gang-scheduled; CPU and/or memory intensive

DAG of Tasks (e.g. MapReduce):

jobs of similar small, independent tasks

Interactive services (e.g. web serving):

indefinite-length 'jobs'; variable demand; pre-placed servers

Assumptions this trace breaks

Units of work are interchangeable
(in space or time; to a scheduler)

Scheduler acts infrequently (or simply)

Tasks will indicate what resources they require

Machines are interchangeable

Terminology and sizes

tasks (25M): 'run a program somewhere once'

- more like MapReduce worker than MR task
- Linux containers (shared kernel; isolation)
- may fail and be retried (still *same* task)

jobs (650k): collections of related tasks

- no formal coscheduling requirement

machines (12.5k): real machines

Assumptions this trace breaks

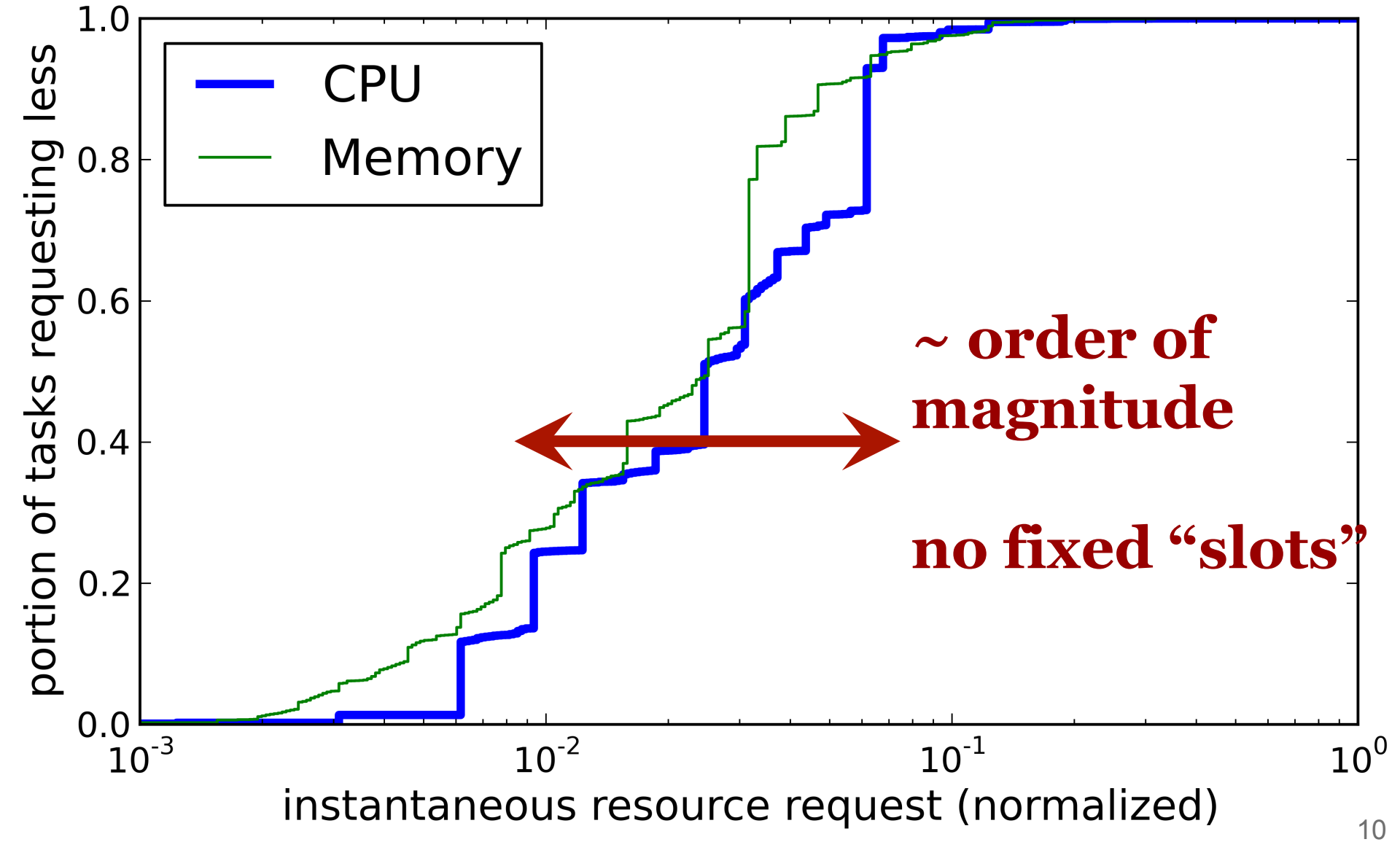
Units of work are interchangeable
(in space or time; to a scheduler)

Scheduler acts infrequently (or acts simply)

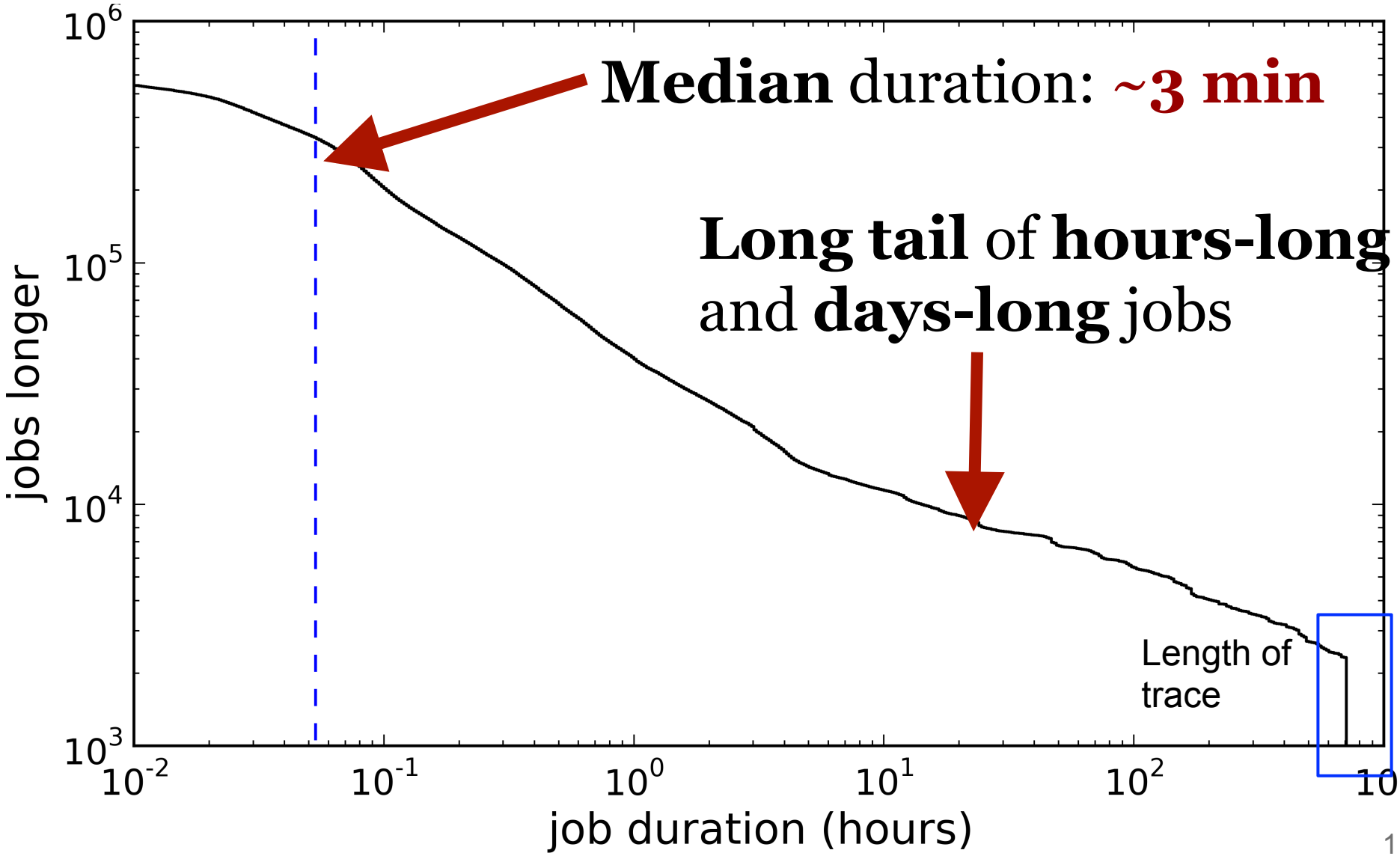
Tasks will indicate what resources they require

Machines are interchangeable

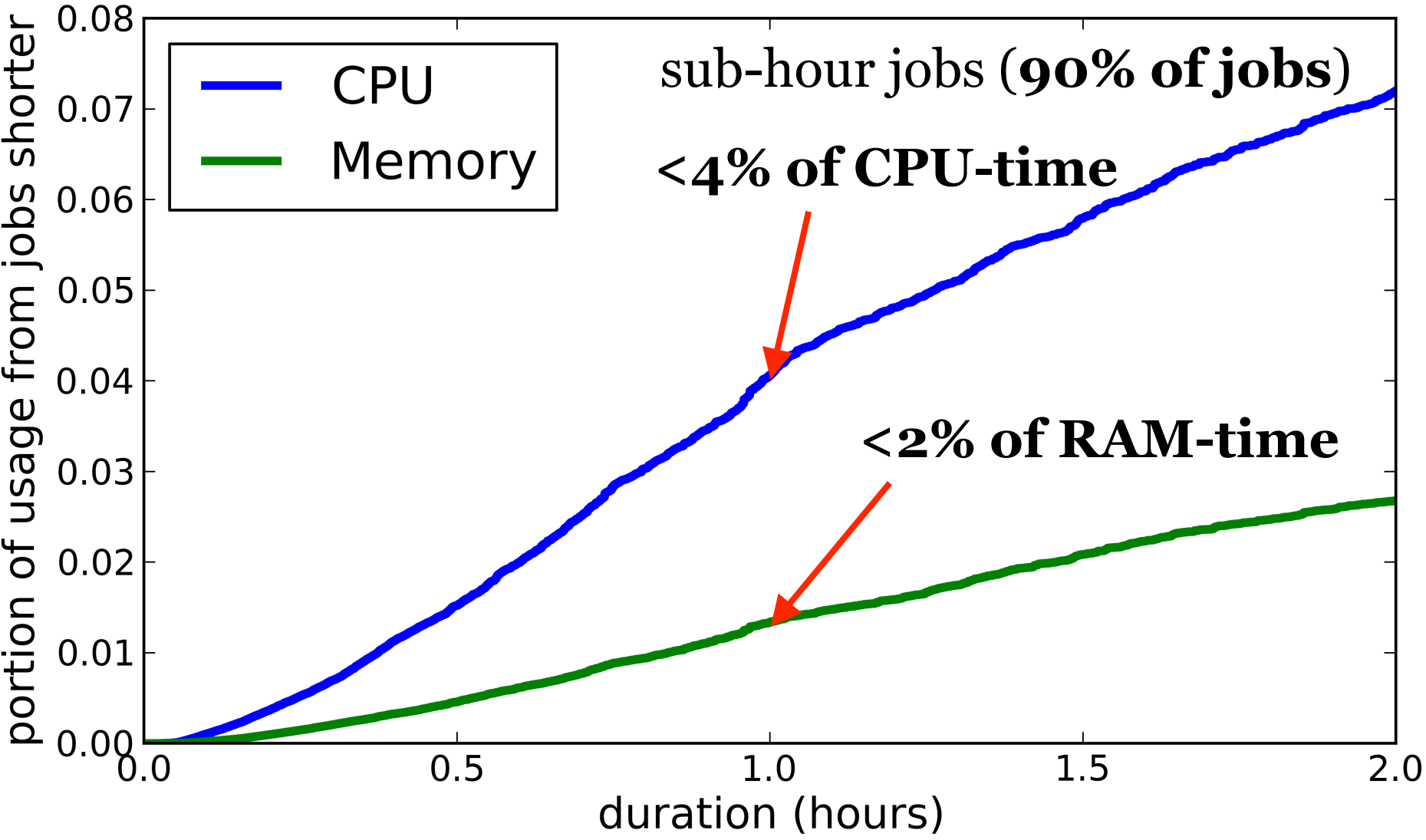
Mixed workload: Task sizes



Mixed workload: Job durations



Mixed workload: Long-running tasks are most usage



Assumptions this trace breaks

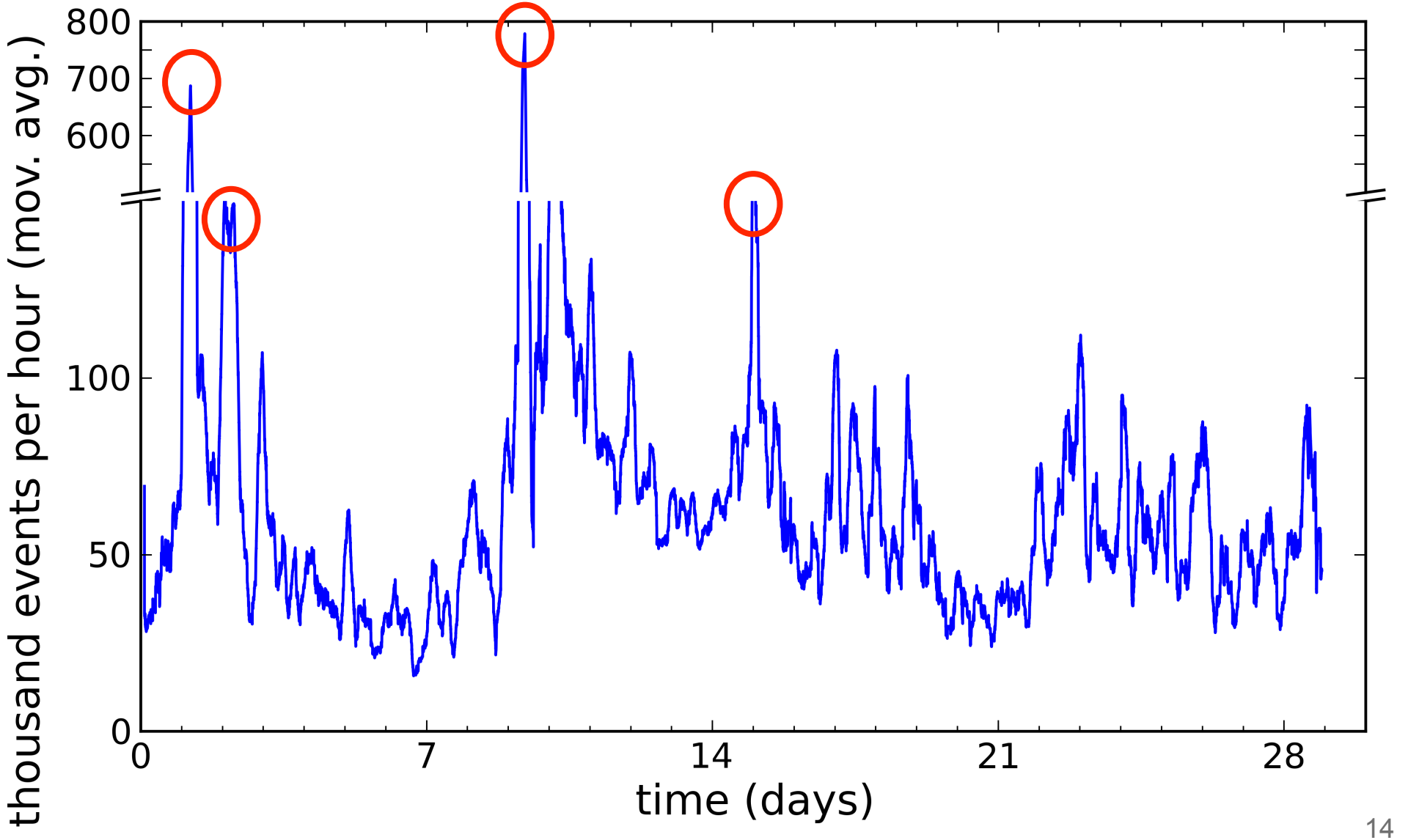
~~Units of work are interchangeable~~
~~(in space or time; to a scheduler)~~

Scheduler acts infrequently (or acts simply)

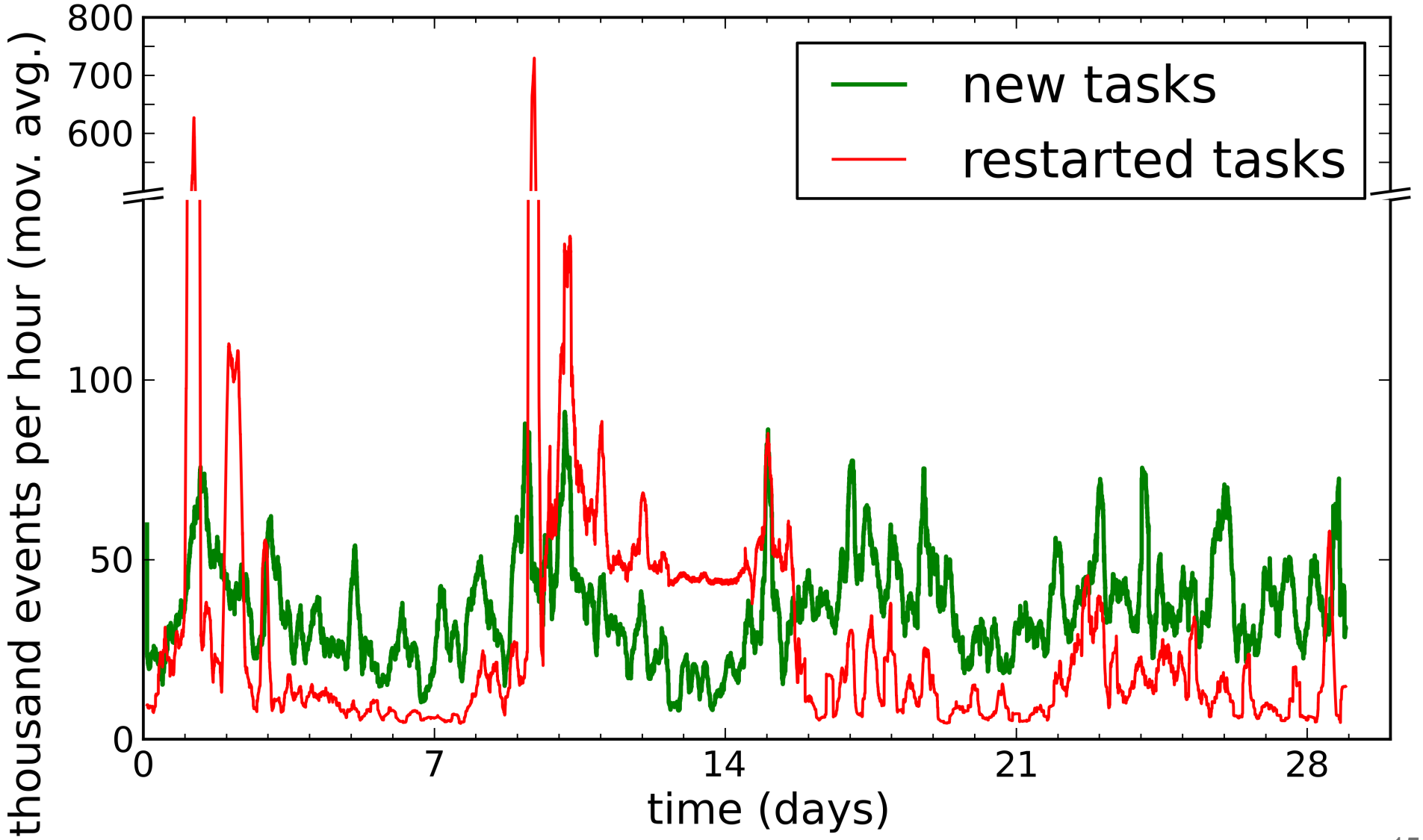
Tasks will indicate what resources they require

Machines are interchangeable

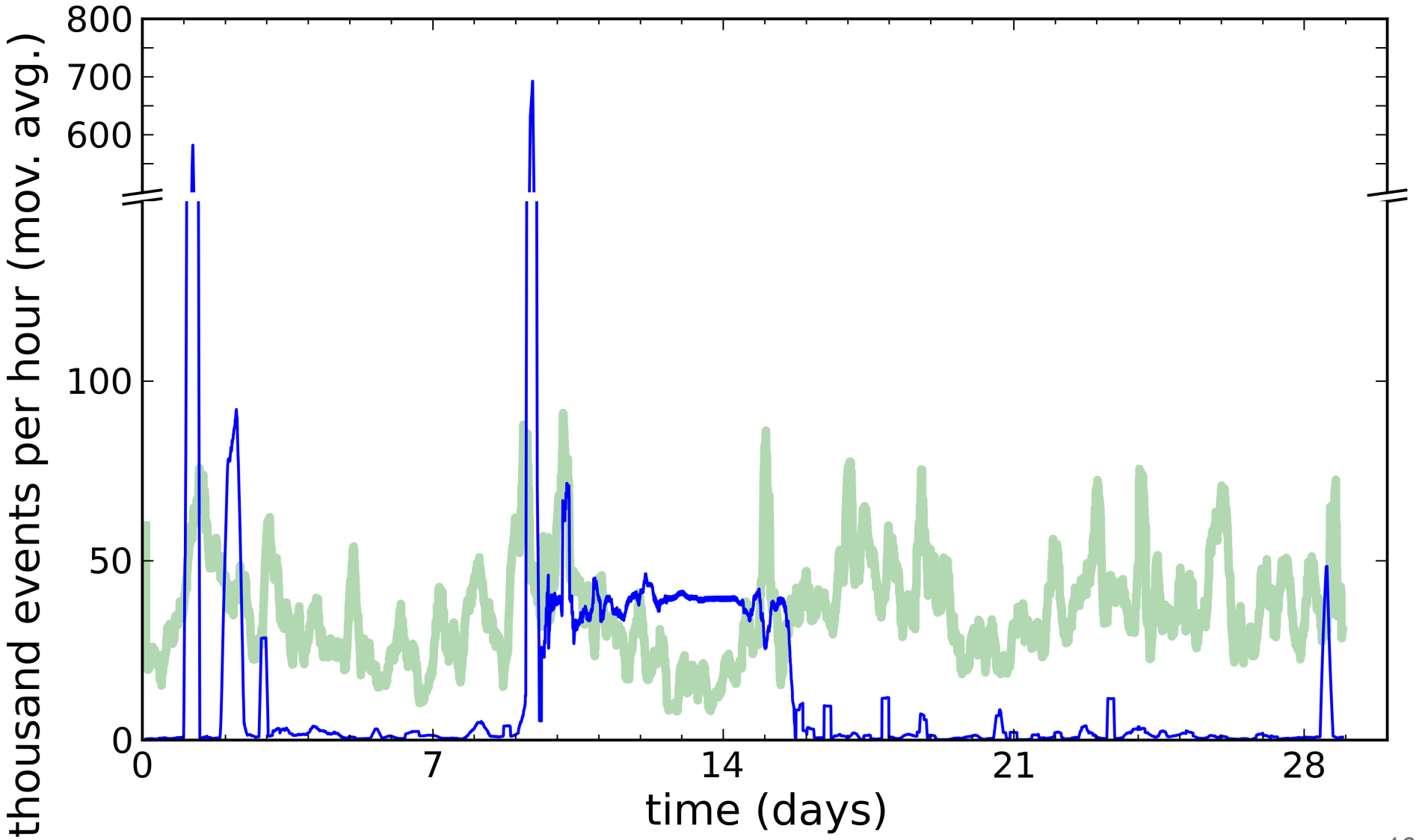
Fast-moving workload: 100k+ of decisions per hour



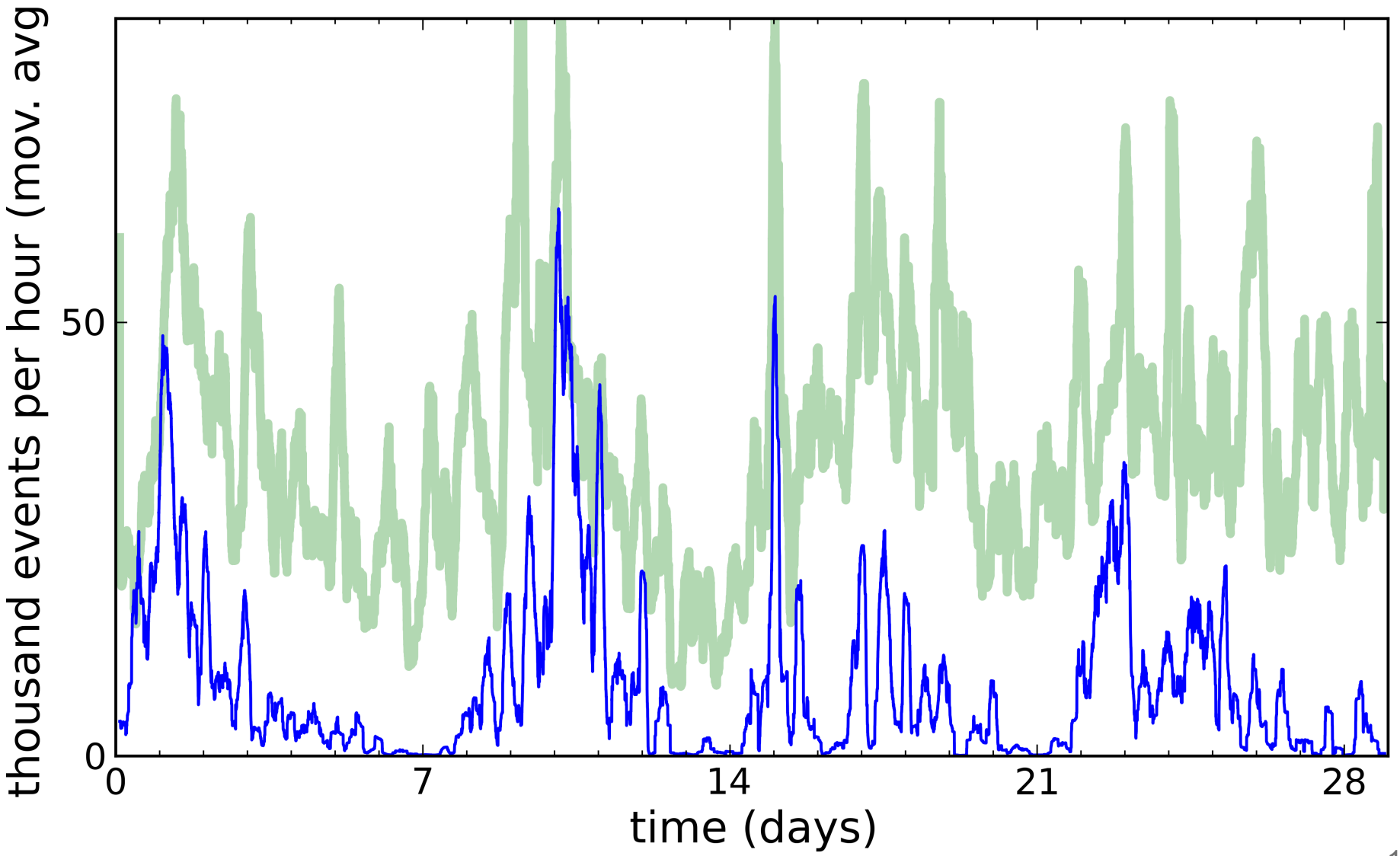
Fast-moving workload: 100k+ decisions per hour



Fast-moving workload: Crash-loops



Fast-moving workload: Evictions



Fast-moving workload: Evictions

Most evictions for **higher-priority tasks**:

- Coincide with those tasks *starting*
- 0.04 evictions/task-hour for lowest priority

A *few* for machine downtime:

- **40%** of machines down once in the month
- Upgrades, repairs, failures

Assumptions this trace breaks

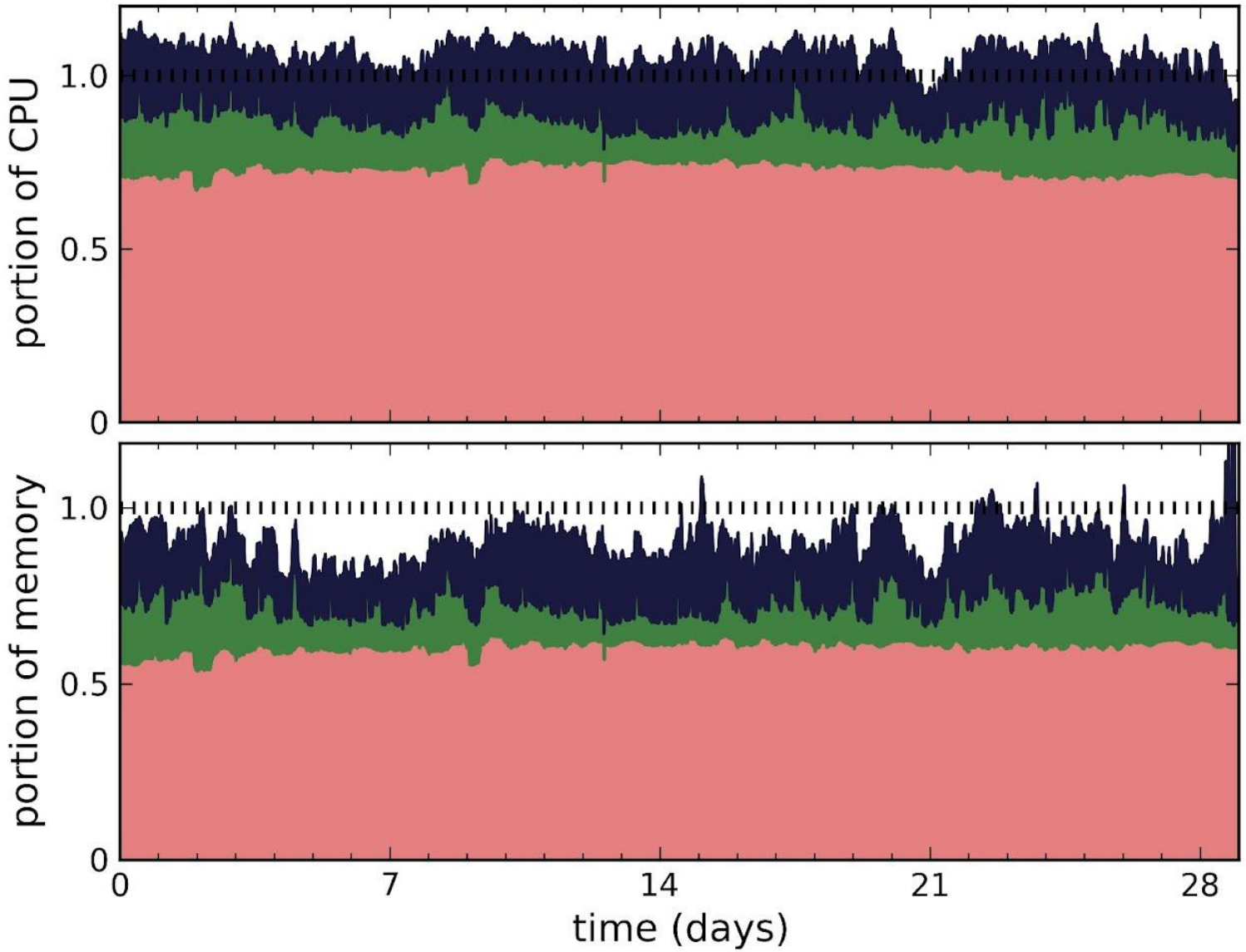
~~Units of work are interchangeable (in space or time; to a scheduler)~~

~~Scheduler acts infrequently (or acts simply)~~

Tasks will indicate what resources they require

Machines are interchangeable

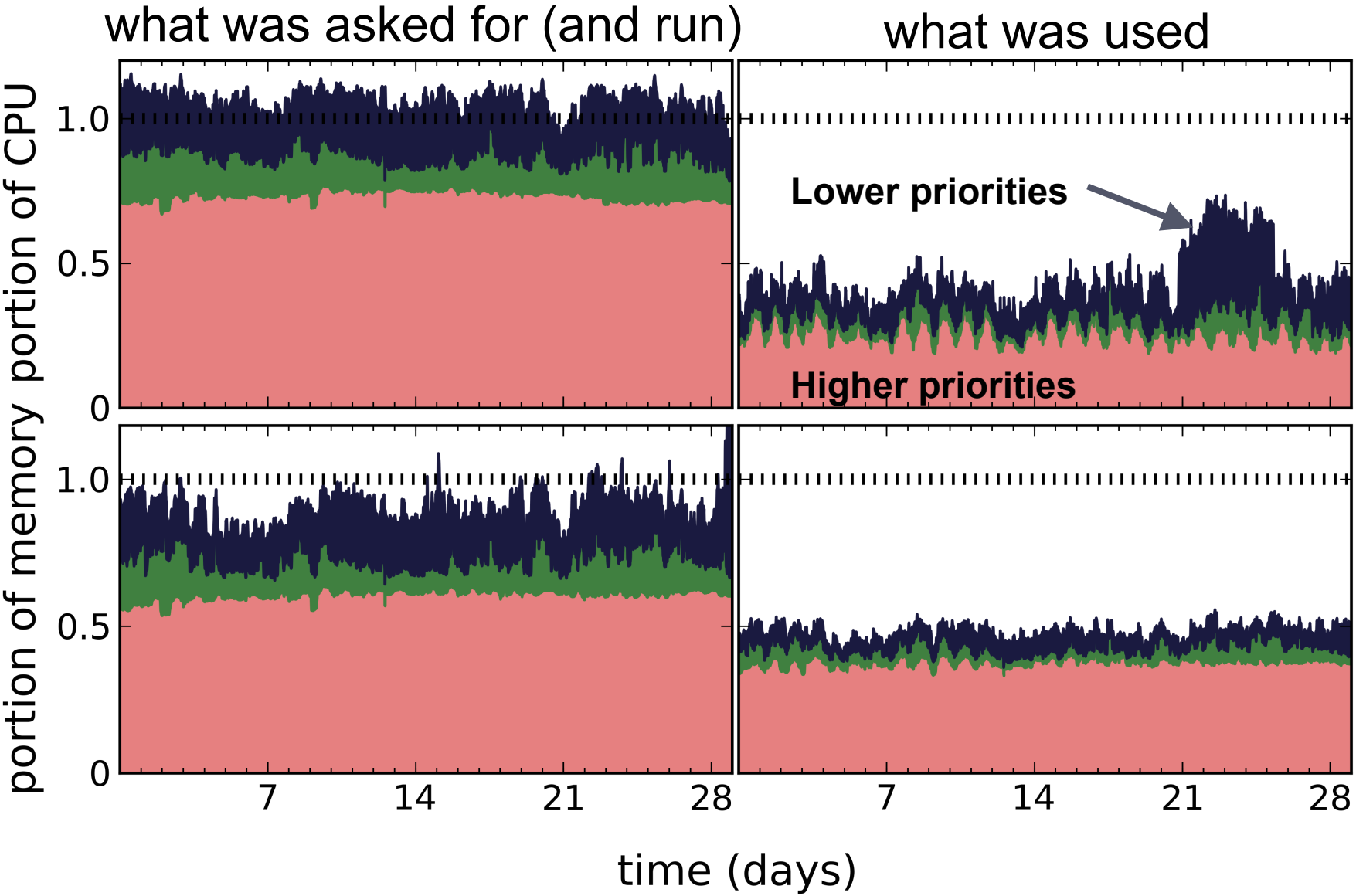
How busy is the cluster?



lower prio

"production" (higher prio)

How busy is the cluster *really*?



Request accuracy: Maximum versus Average

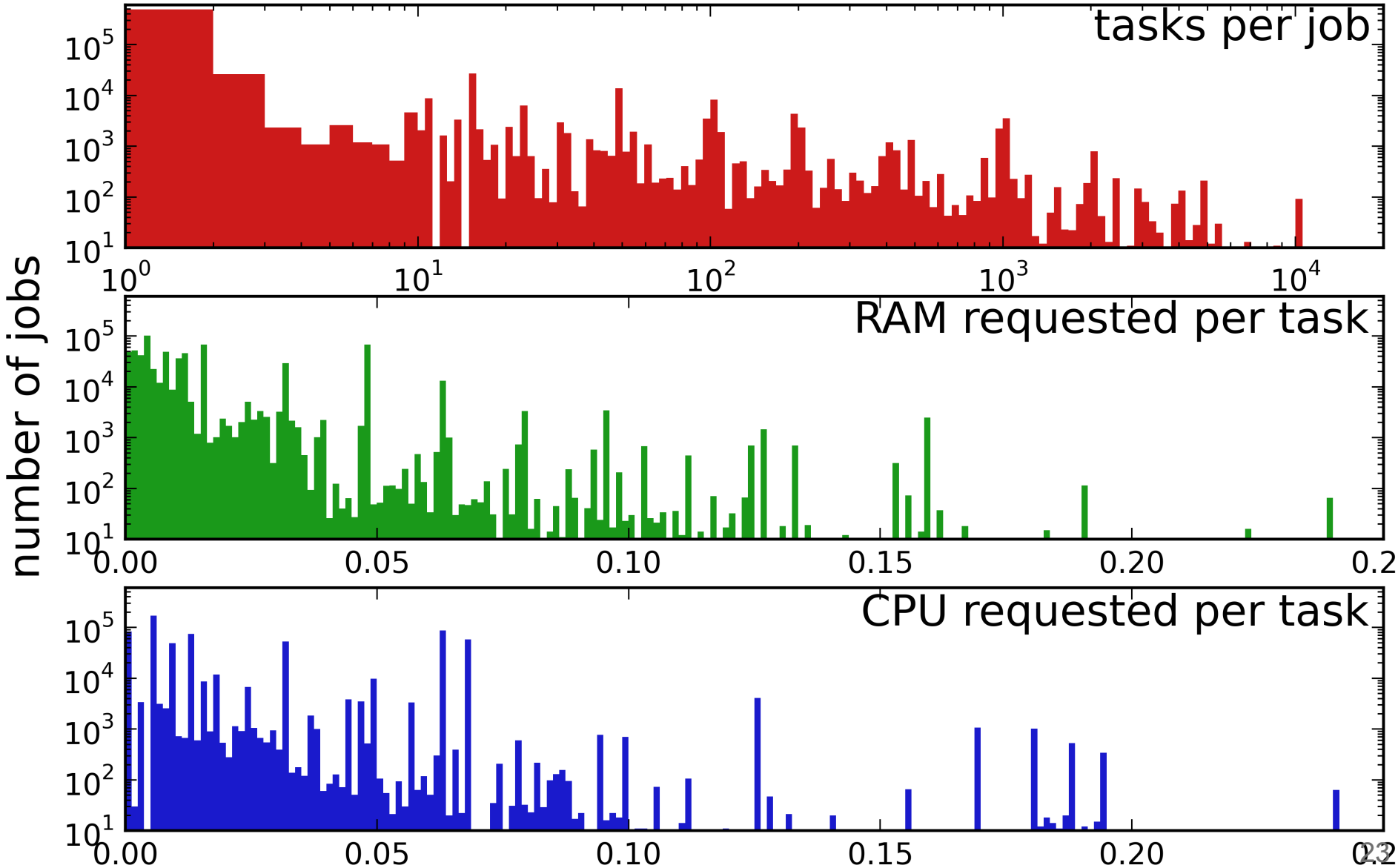
Requests estimate **worst-case** usage

~60% of request/usage difference from difference between worst/average usage:

- Average task versus worst task in job
- Average usage versus worst usage in task

But **not enough** to explain request/usage gap

Request accuracy: Requests are people



Assumptions this trace breaks

~~Units of work are interchangeable~~
~~(in space or time; to a scheduler)~~

~~Scheduler acts infrequently (or acts simply)~~

~~Tasks will indicate what resources they require~~

Machines are interchangeable

Not all machines are equal: Machine types

Factor of **4**

Count	Platform	CPU	Memory
6732	B	0.50	0.50
3863	B	0.50	0.25
1001	B	0.50	0.75
795	C	1.00	1.00
126	A	0.25	0.25
<100	B and C	(various)	(various)

Three micro-architectures

Not all machines are equal: Task constraints

Tasks can restrict acceptable machines (for reasons other than resources)

Used by ~6% of tasks

Examples:

- Some jobs require each task to be on a different machine

- Some tasks avoid 142 marked machines

Conclusion

New scheduling challenges for mixed workloads:

Complex task requests
- Order of magnitude range of resources
- Extra constraints
- Matched against variety of machines

Rapid scheduling decisions
- Short tasks (with little utilization)
- Restarted tasks

Users' requests not enough for high utilization

[Backup/Discarded Slides]

Request accuracy: Evaluating

Resource requests = **worst-case usage**

Estimate: "ideal" request ≈ high percentile of usage within each job

Imperfect:

- Opportunistic usage
- Assumes outliers are spurious
- Doesn't account for peaks
- Extra capacity needed for failover, etc.

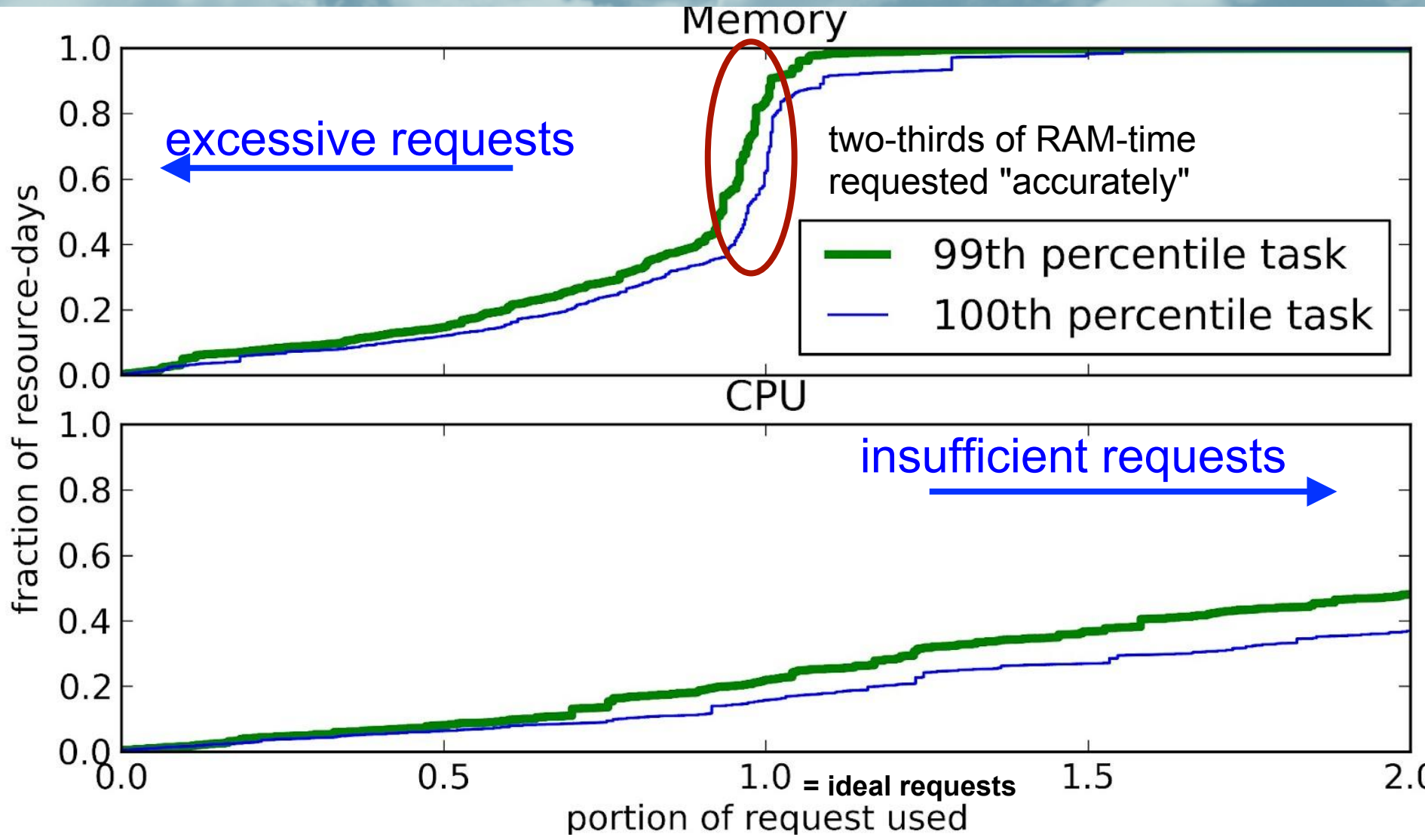
Conclusion

Heterogenous: Machines, type of work varies
- Some scheduling strategies won't work
- Space on a machine varies

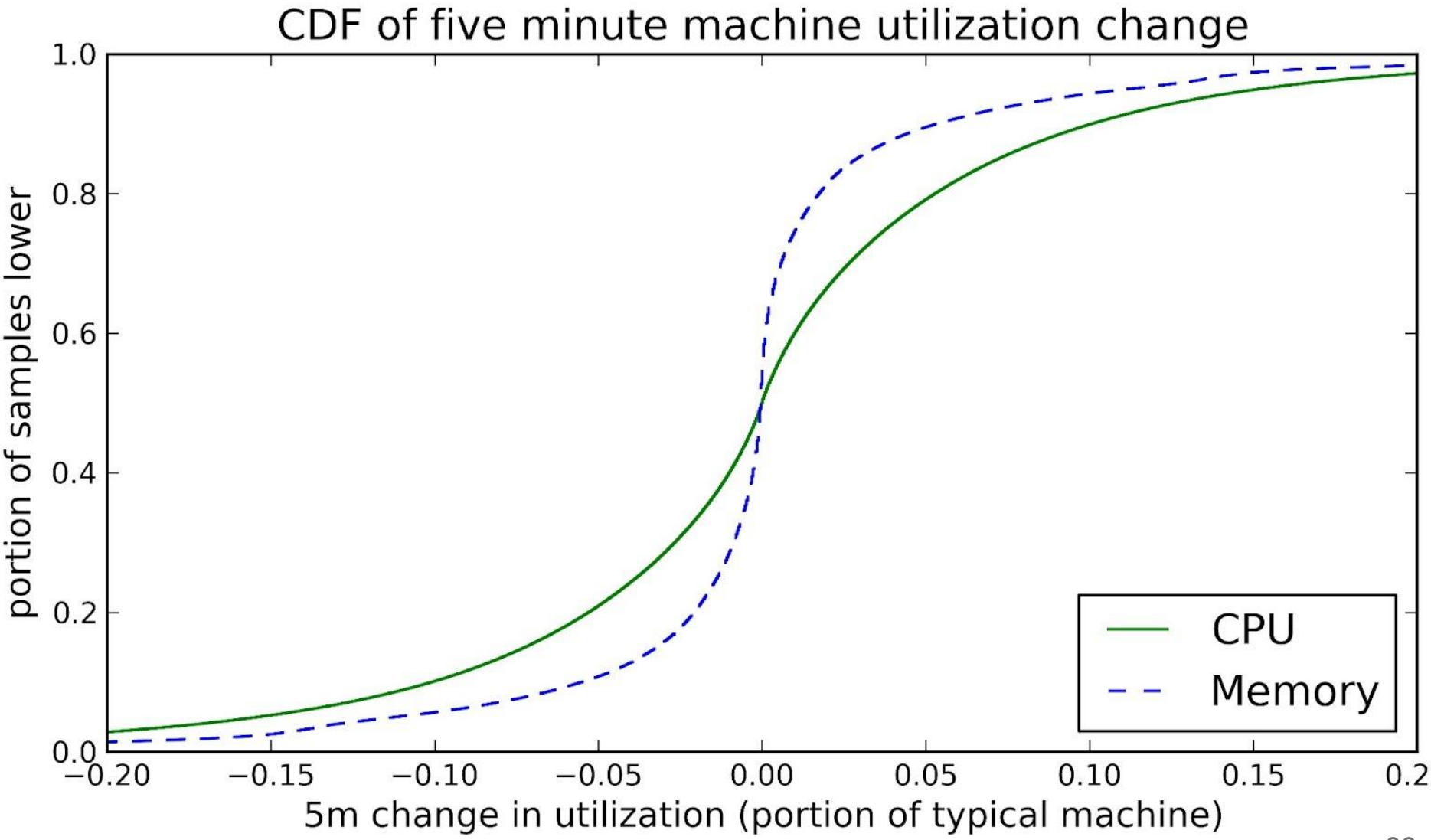
Dynamic: Work comes fast
- Not just initial submissions
- Only a small amount matters for utilization

Resource requests are suboptimal
- Users do not make good requests
- Resource requirements may vary

Request accuracy: Not very



Predictable usage: Usage stability



Mixed workload: Daily patterns