Alibaba Group

Kubernetes & YARN: a hybrid container cloud

Jian He & Bushuang Gao

Alibaba Group

About us

- Jian He
  - Staff Engineer @Alibaba cluster management team
  - Staff Engineer @Hortonworks
  - Hadoop Committer & Project Management Committee member
- Bushuang Gao
  - Senior Engineer @Alibaba

Alibaba Group

Agenda

- What/Why co-location
- Co-location @ Alibaba
- Kubernetes vs YARN
- Kubernetes & YARN: a hybrid architecture
- Resource Isolation
- Future

Alibaba Group

What/Why co-location

Alibaba Group

Data center Avg utilization

- Gartner has long talked about the "80% rule": that 80 percent of IT budgets get spent simply "keeping the lights on”
- The average data center cpu utilization is about 10%

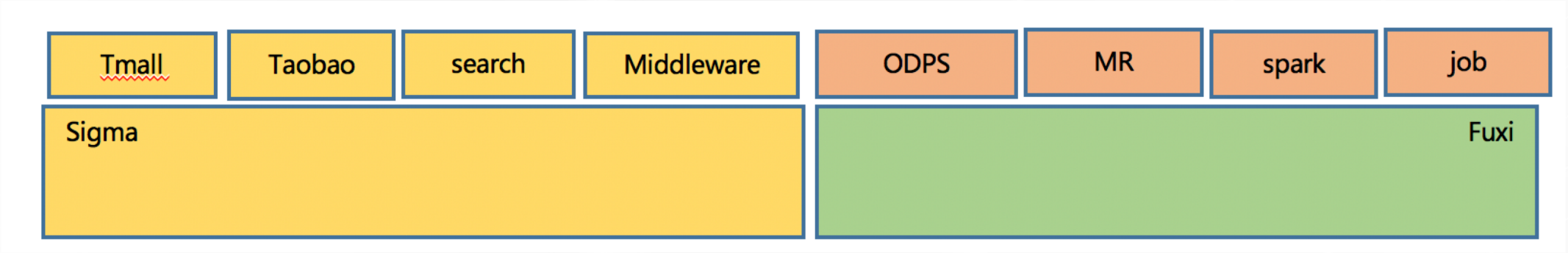
Alibaba Group

Workloads in Alibaba data center

- Services
  - Commercial platform, taobao, Tmall, retail apps, search,
  - Payment platform: Alipay
  - Middleware: queue service, cache server
- Jobs
  - Big data batch jobs: MR, spark
  - Realtime jobs: flink

Alibaba Group

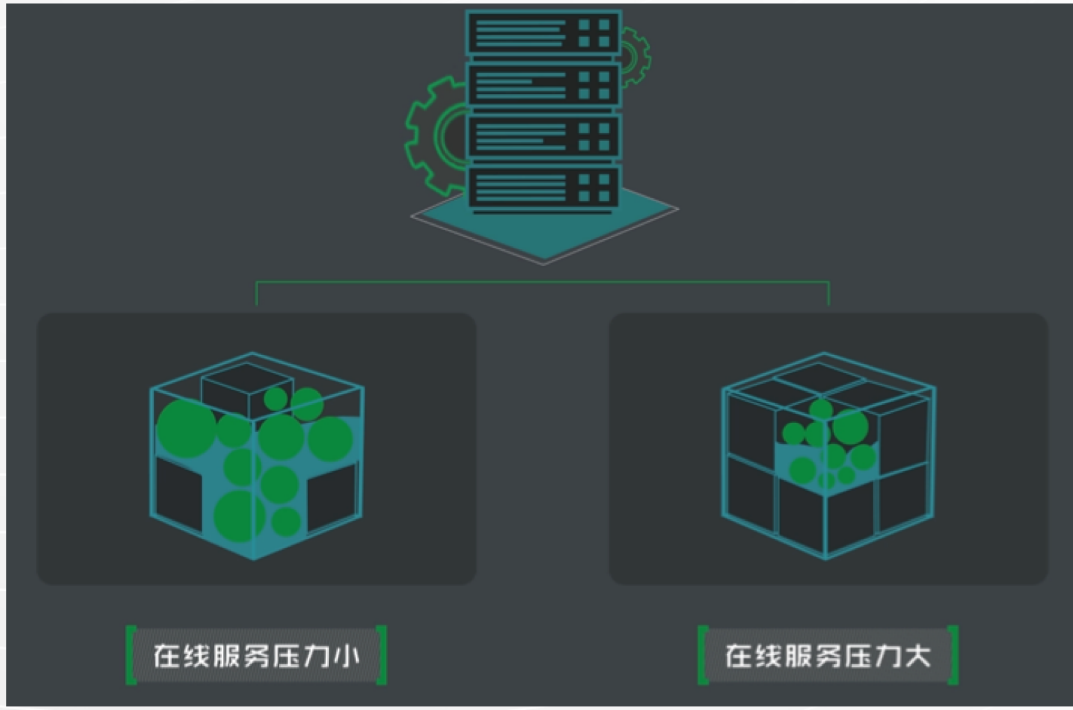
Before: separated Cluster



Services vs Jobs

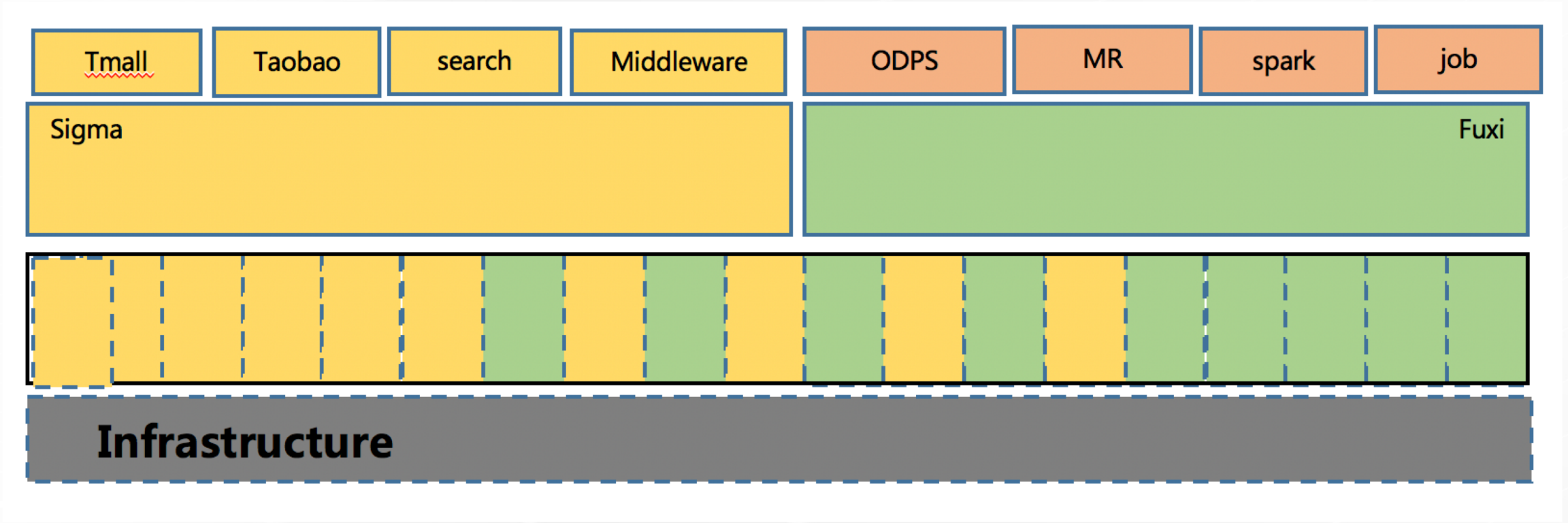
	Online service	Batch jobs
Category	Online shopping web apps, payment service	MR, spark, flink
Latency	Sensitive	Insensitive
Priority	high	low
Traffic pattern	Peak at day time	Peak at night time
Fault tolerance	should not fail	Fail and retry

Complementary !



Alibaba Group

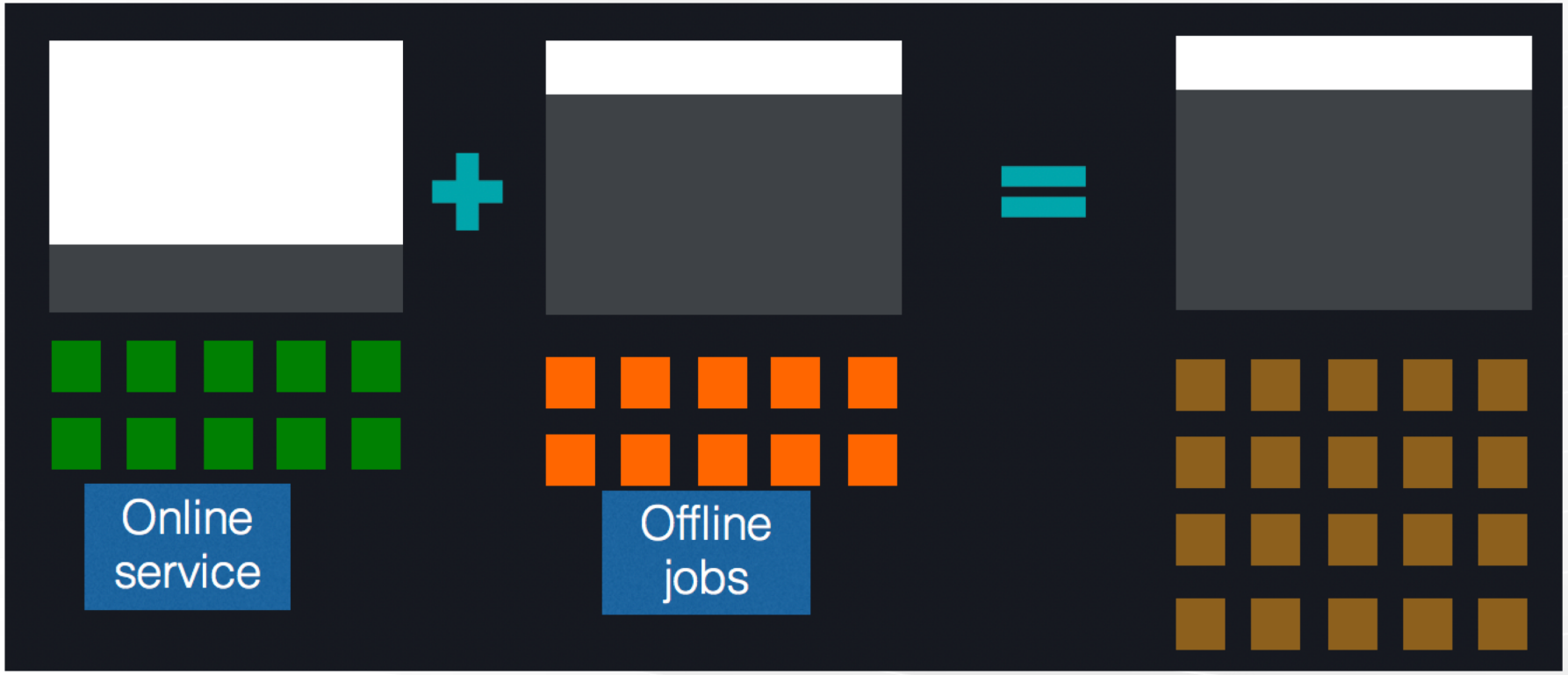
Now: co-locate clusters



Alibaba Group

What is co-location

- A hybrid deployment that supports both online services and offline jobs using the same hardware.



Alibaba Group

Key concepts in co-location

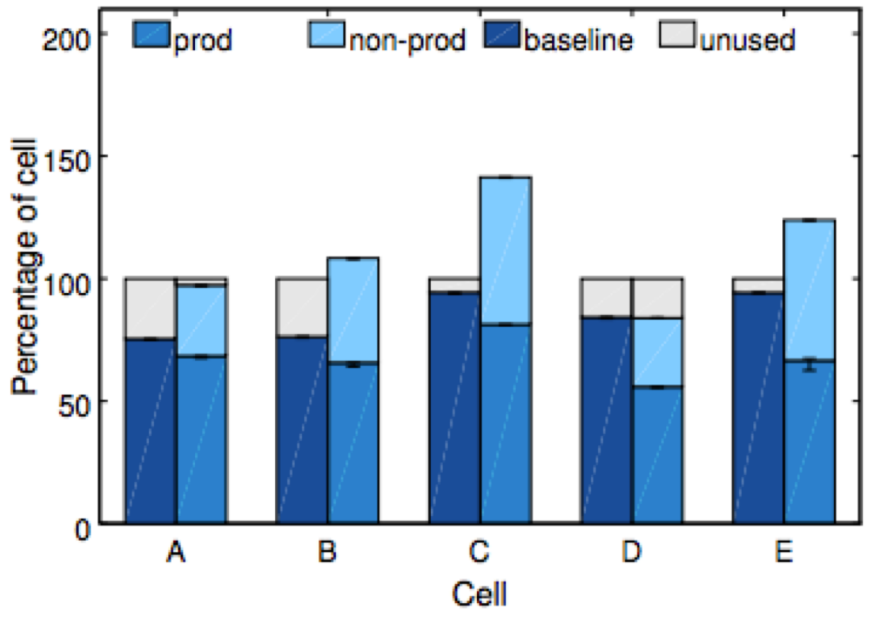
- **Resource sharing:** Make services and jobs share a single cluster and have them run on the same hardware
- **Resource Isolation**: Rely on scheduling efficiency and resource isolation to guarantee the SLA.

Alibaba Group

Why co-location

- Two separated cluster cause huge resource wastage. Increase resource utilization.
- 1 hour peak traffic on double 11 requires a lot of extra resources. However, those resources will be idle after the peak time.
- Lend those resources to computation jobs

Google Borg



(a) *The left column for each cell shows the original size and the combined workload; the right one shows the segregated case.*

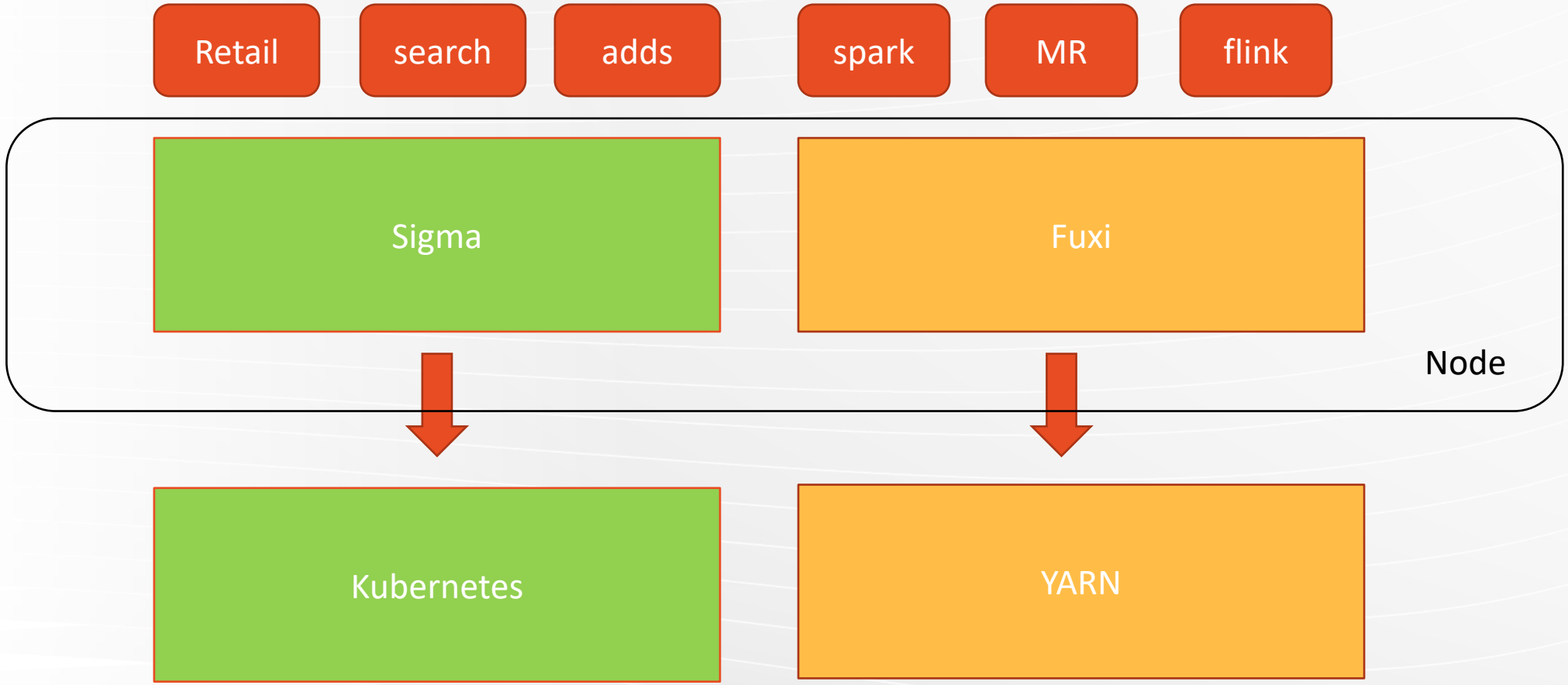
Borg paper mentions 20% - 30% more machines if
If segregating prod and non-prod workloads

Alibaba Group

Co-location @Alibaba

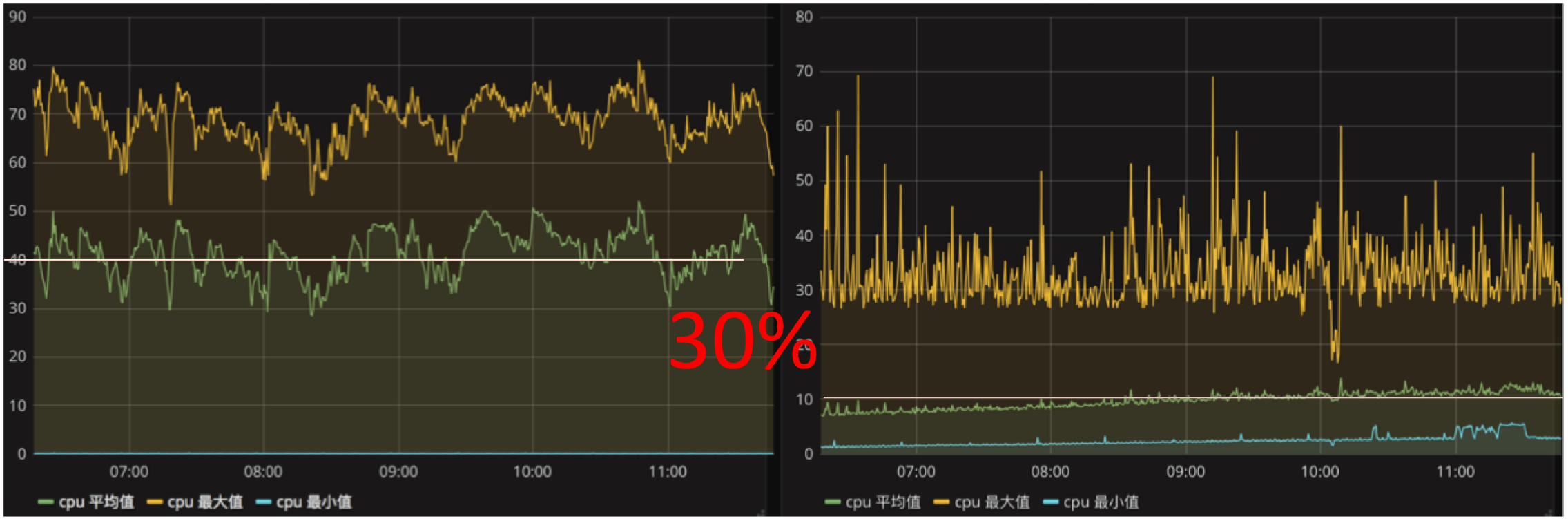
Alibaba Group

Co-location @Alibaba

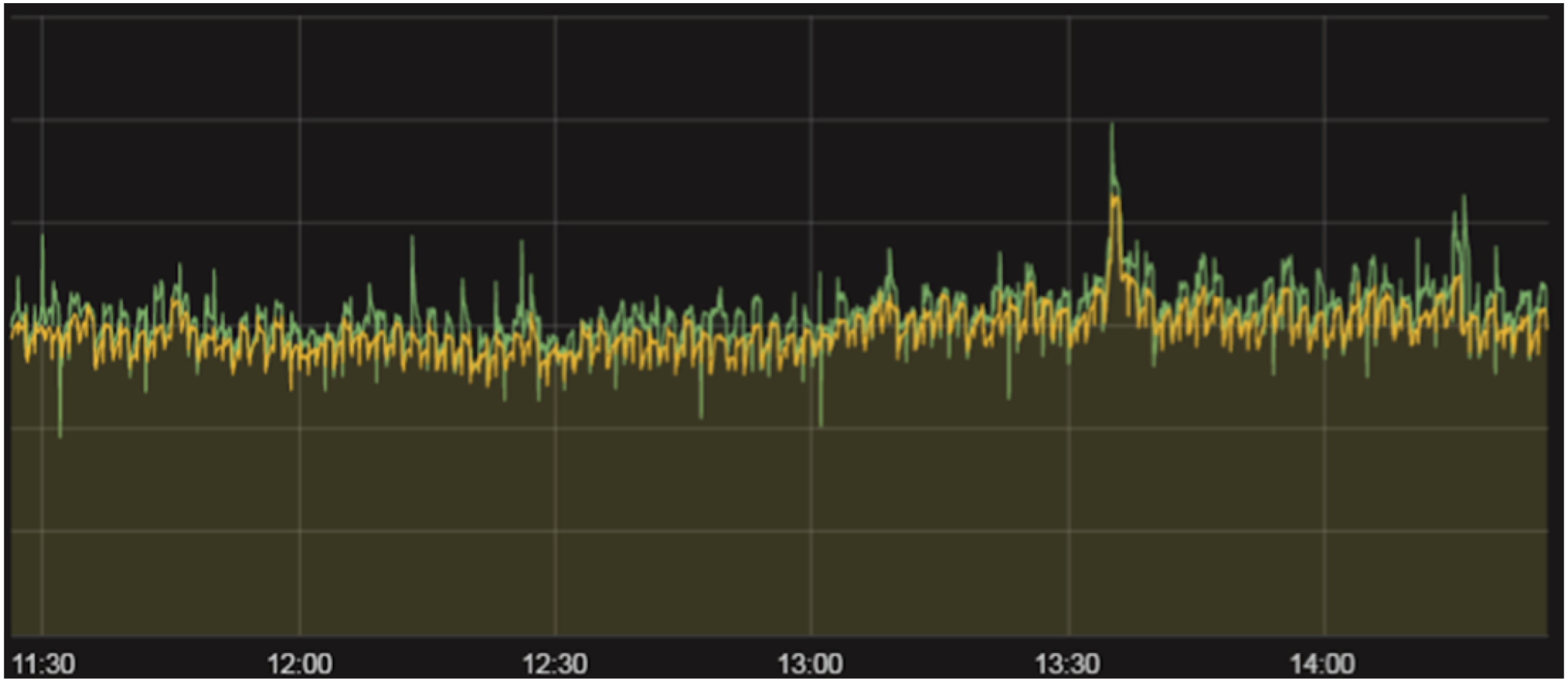


Co-located vs separated cluster CPU utilization

Co-located：40%　　Seperated：10%

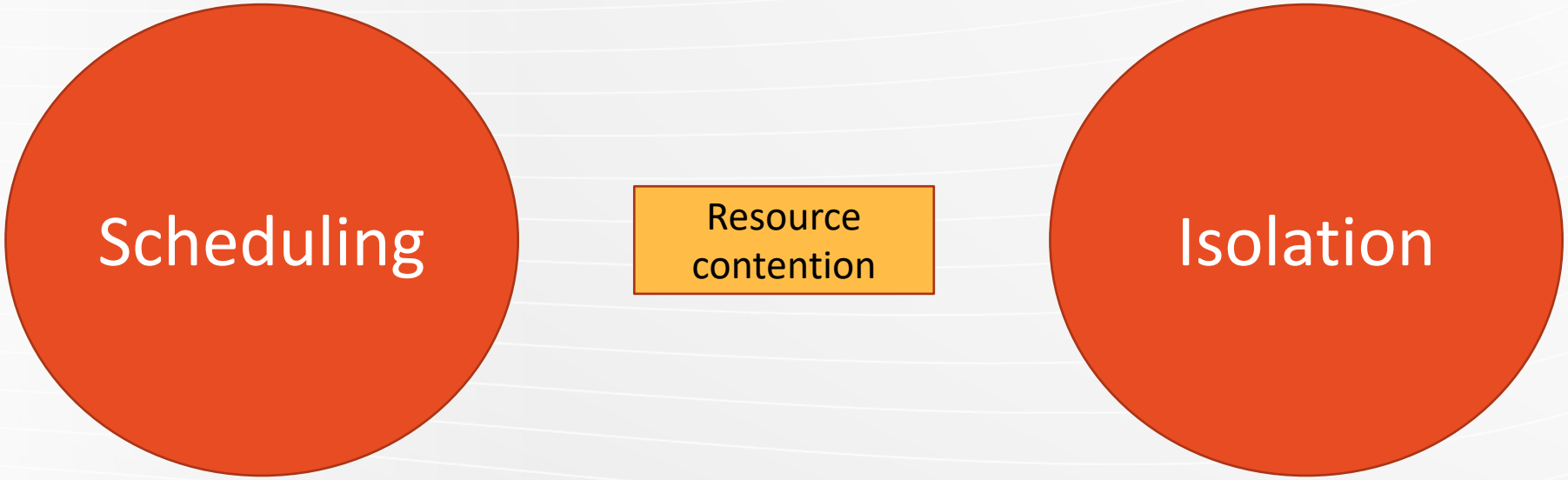


Impact on online service within 5%



Alibaba Group

Two core concepts in co-location



Efficient placement of service container and tasks

When placed together, don’t affect each other

Alibaba Group

Time sliced scheduling

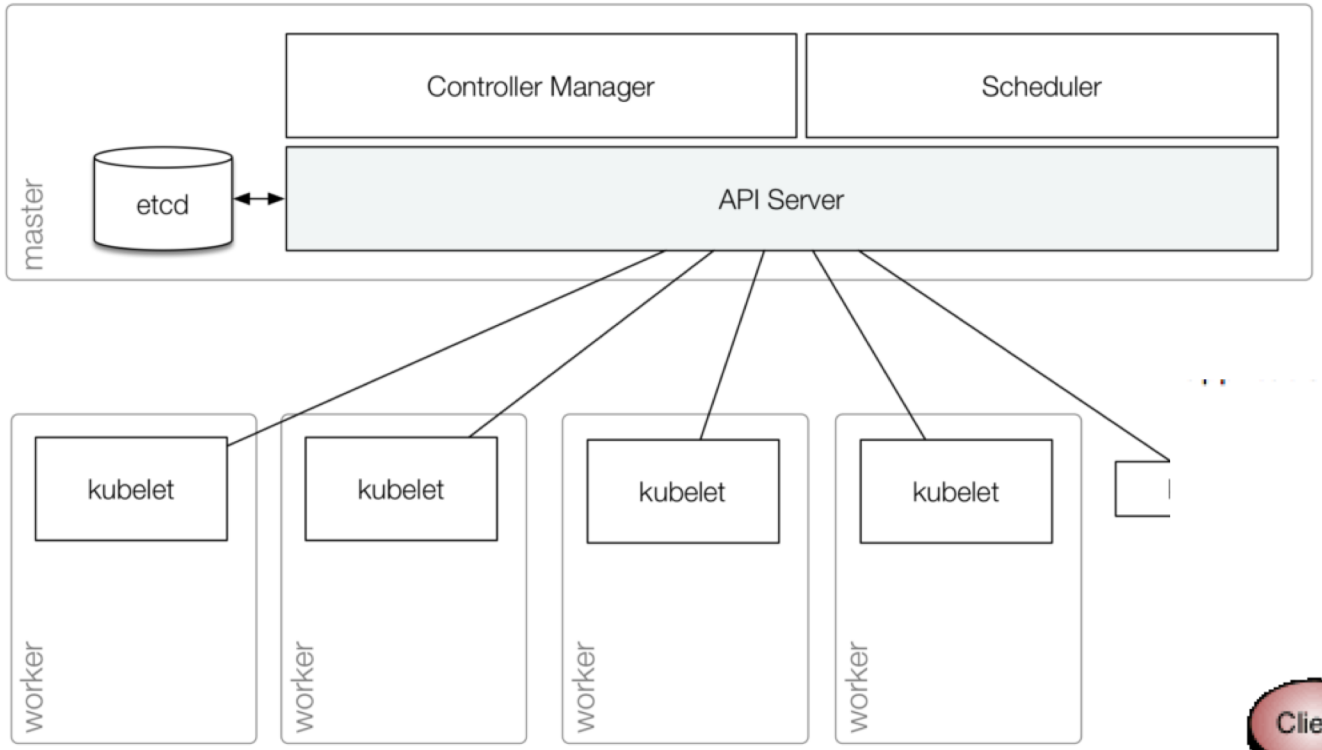
- Online workload low 1:00am – 6:00am
- Offline jobs scale up while online workload remains idle
- Offline jobs scale down while online workload comes back

Alibaba Group

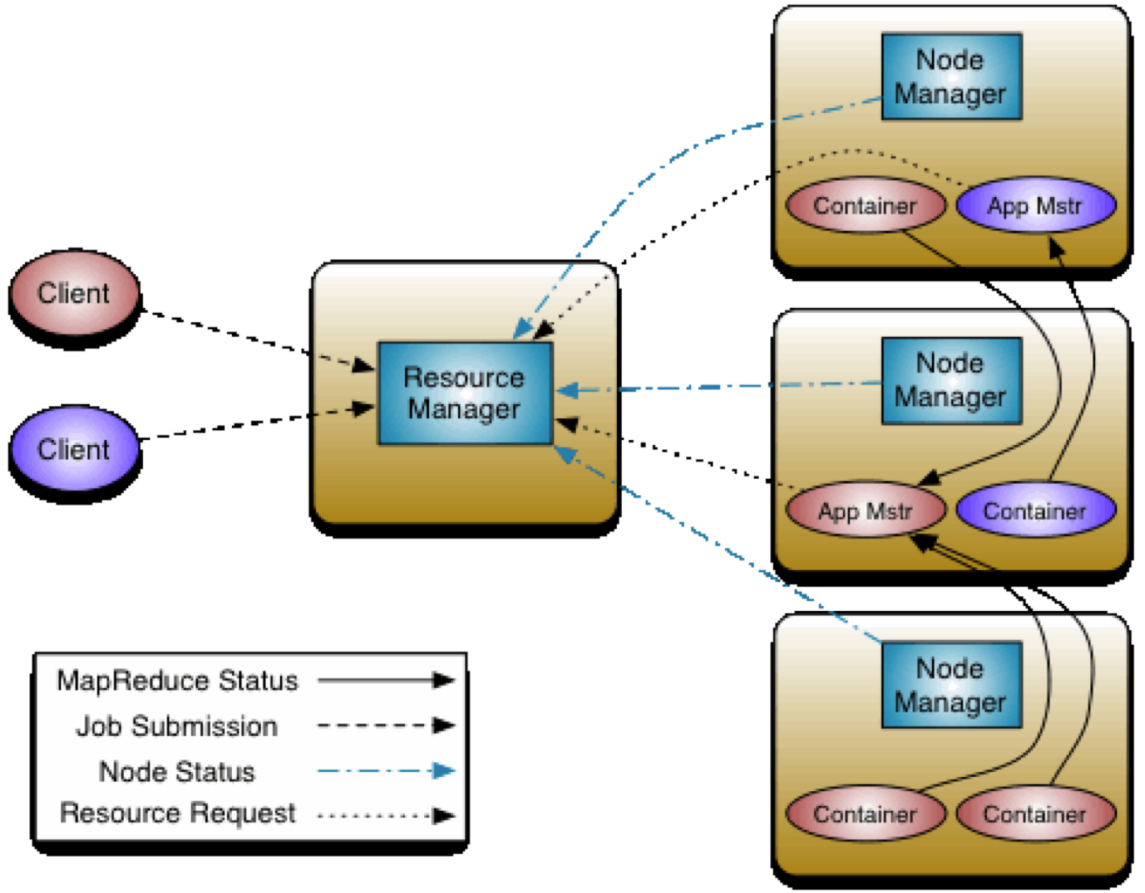
Normal day <-> Sales day

Alibaba Group

Kubernetes vs YARN



Alibaba Group



Alibaba Group

Design Principal

Kubernetes	Focus on long running service. Driving current state towards desired state with control loops
YARN	Focus on scheduling jobs

Alibaba Group

Scheduling Unit

Kubernetes	**Container centric – bottom up.** **Container is the primitive. Other primitives such as replicaset, deployment are built around containers.**
YARN	Application centric: top down. Scheduling sequence: Queue -> user -> application -> container request

Alibaba Group

Communication

kubernetes	**Based on api-server watch mechanism** **Everything stored in etcd**
YARN	Based on RPC Only application-level metadata persisted. Container data is not persisted. Recover from in-memory state from peers

Alibaba Group

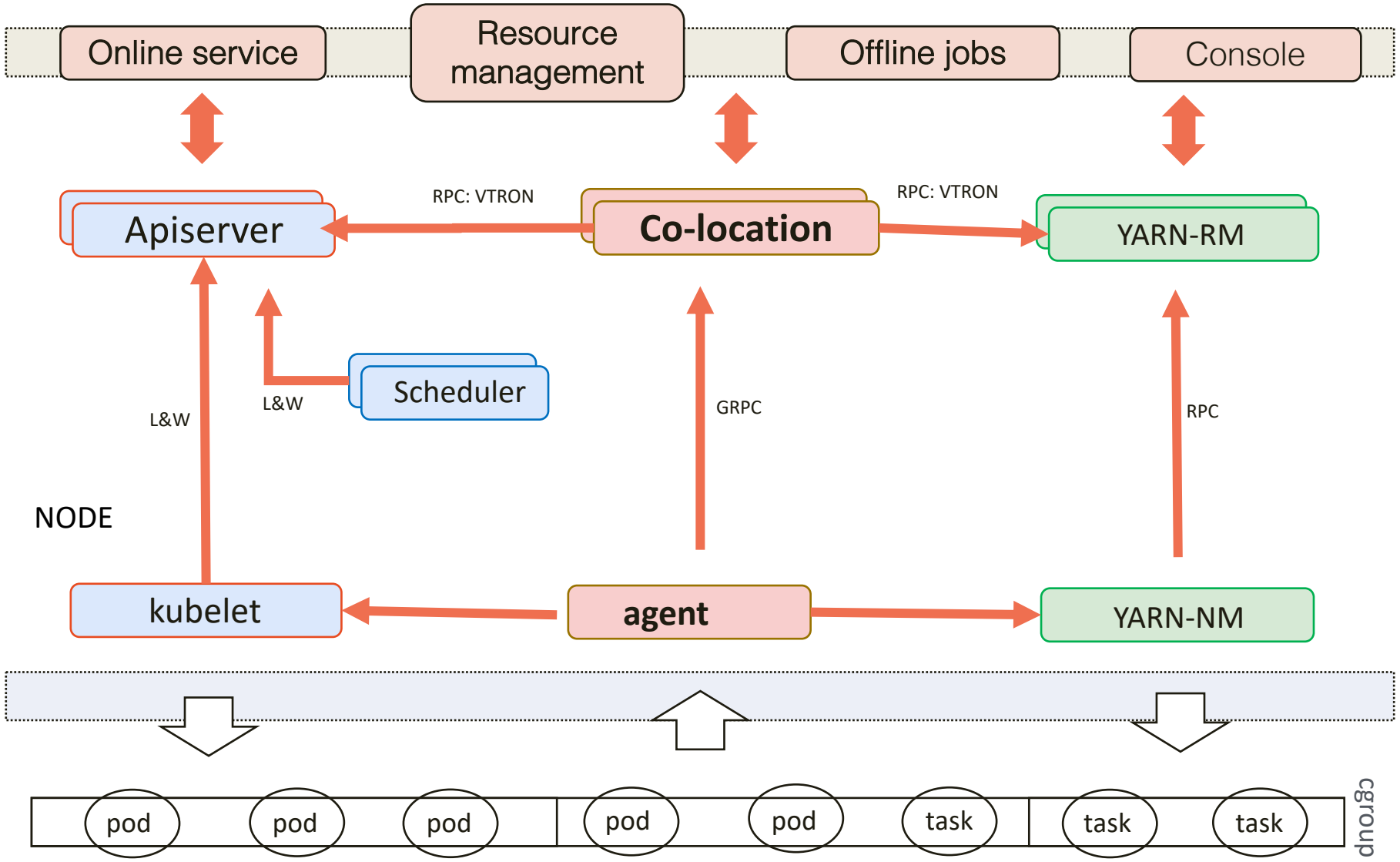
Container Runtime

kubernetes	CRI compatible. Docker etc.
YARN	Docker + TAR ball

Alibaba Group

Kubernetes & YARN: a hybrid architecture

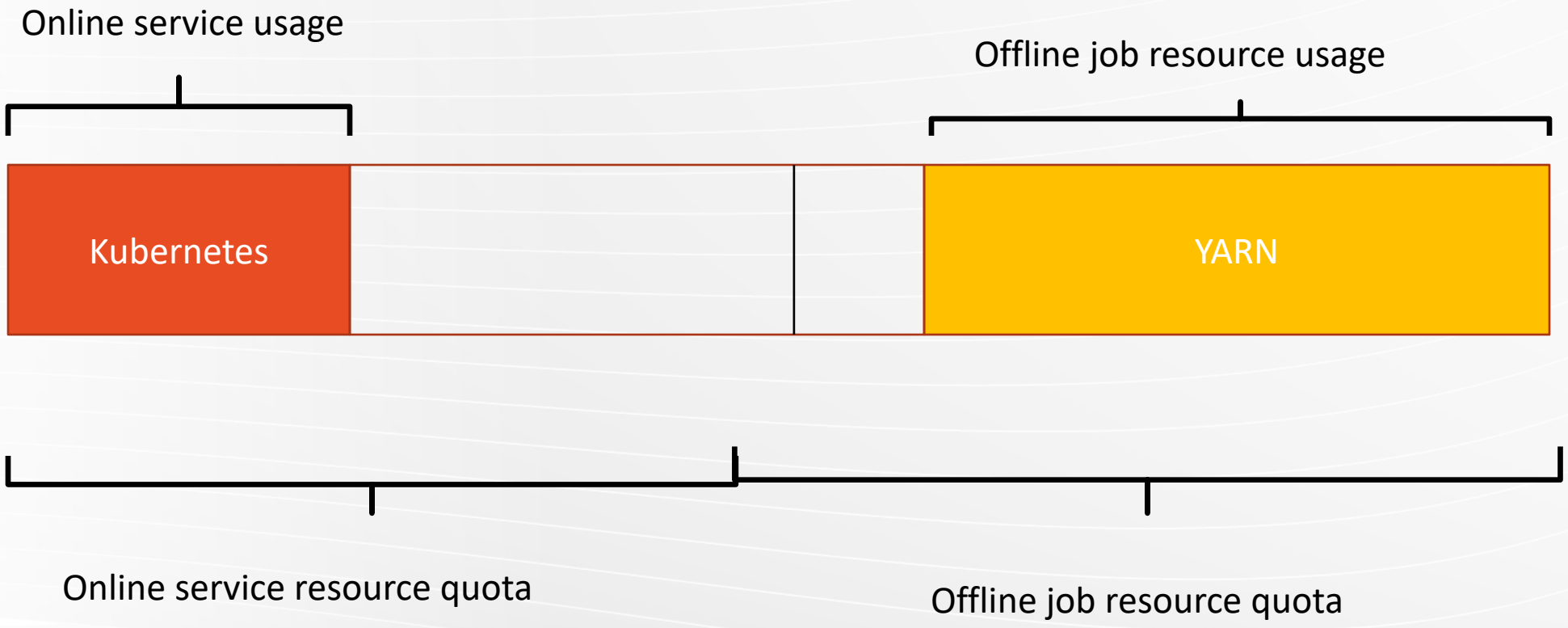
Architecture



VTRON: Virtual Total Resources Of Node

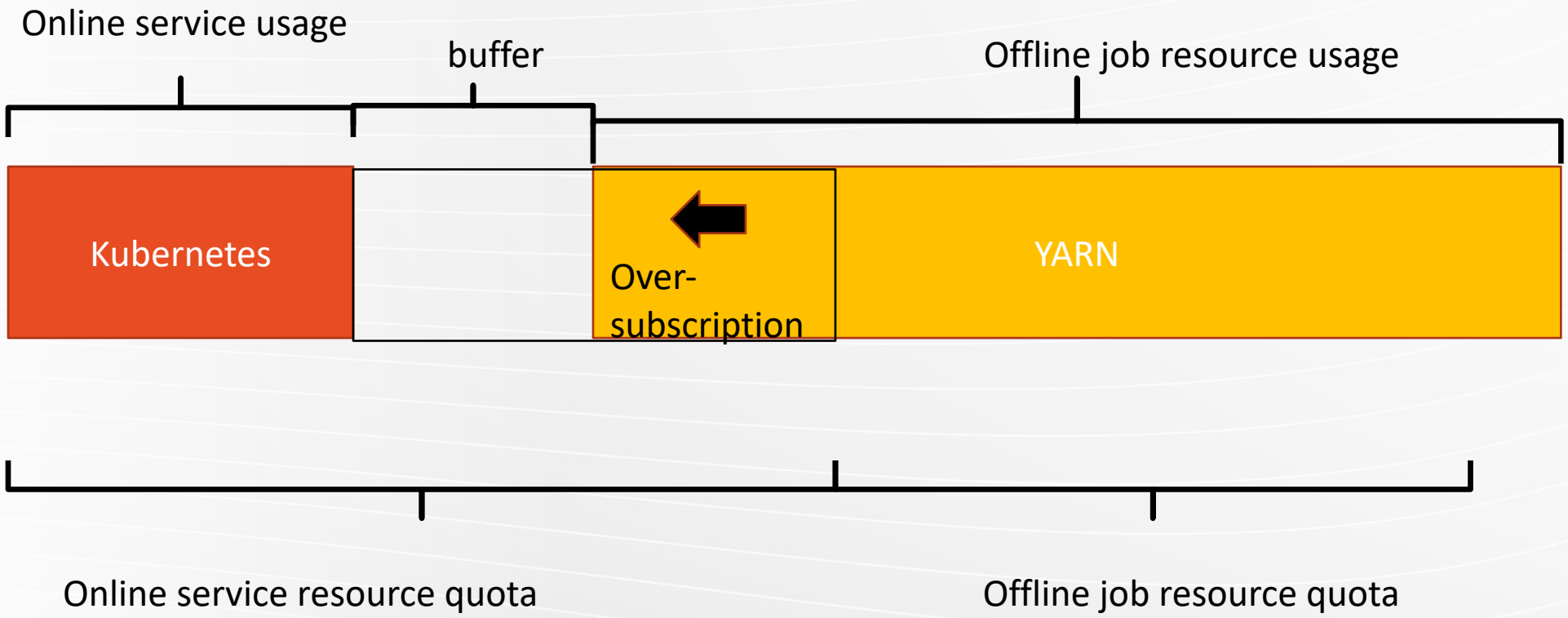
Alibaba Group

resource sharing



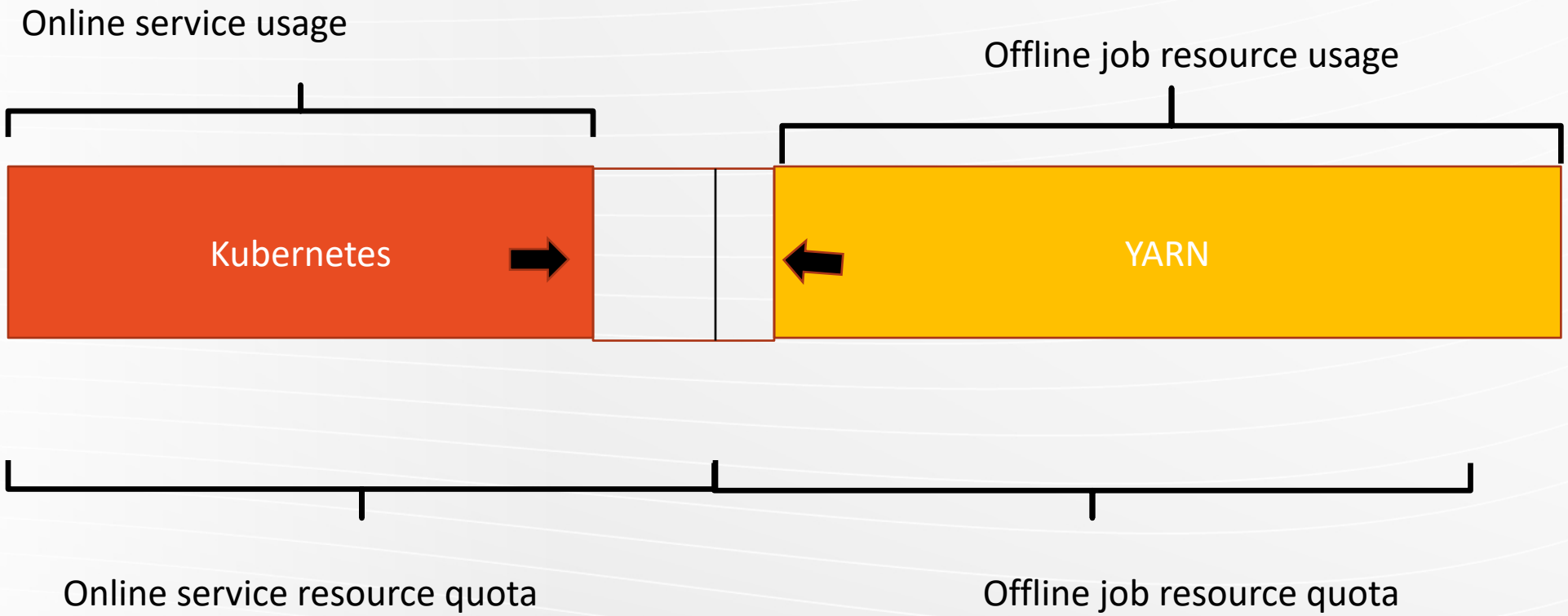
Alibaba Group

resource sharing



Alibaba Group

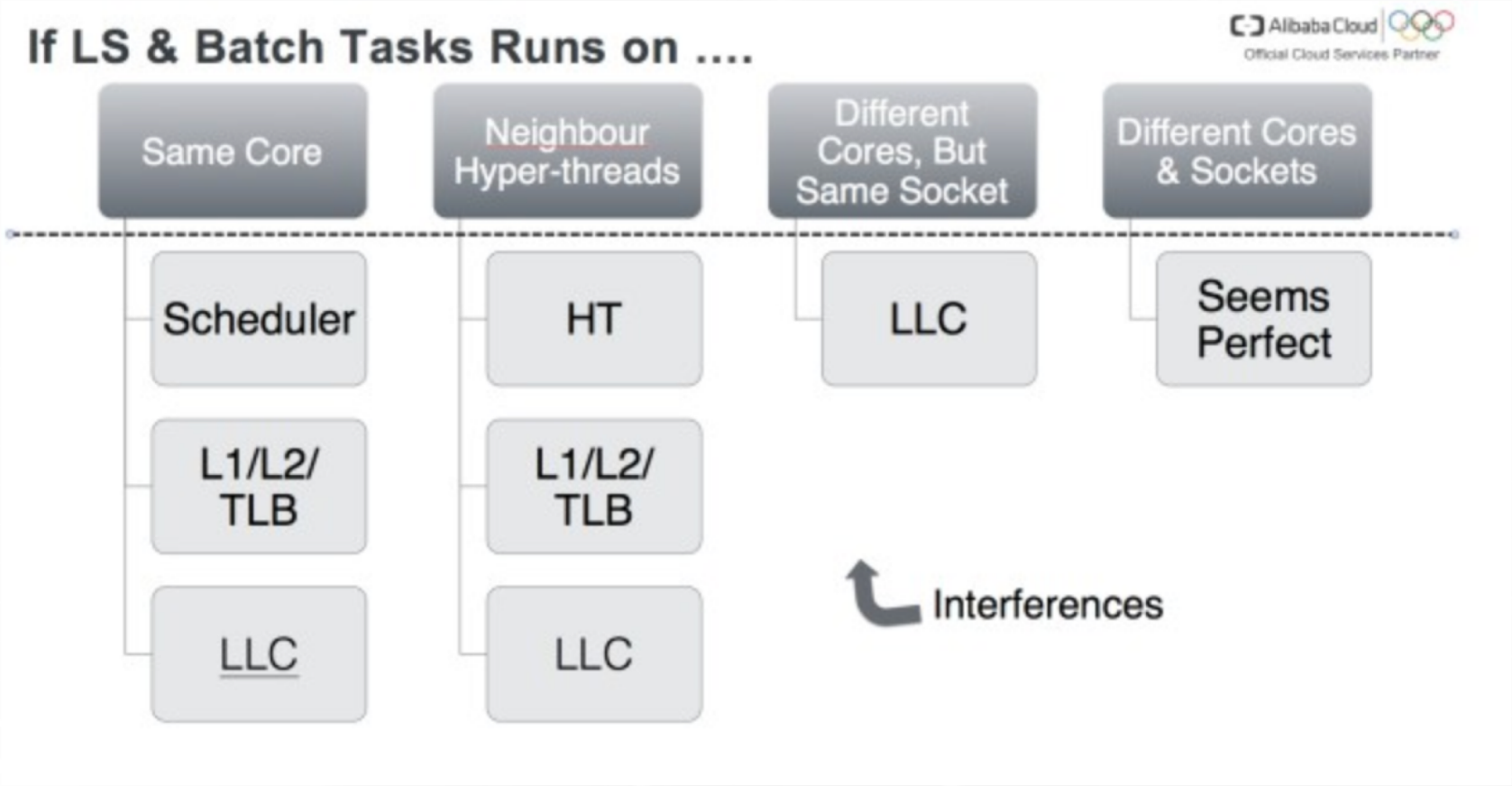
resource sharing



Alibaba Group

Isolation

Interferences



Alibaba Group

CPU Scheduler interference

- Affect two tasks on the same core
- Task preempt (Alikernal feature)
  - Online service can preempt offline job more easily
- cpu.shares
  - Online service has more shares than offline job

Alibaba Group

HT Interference

- 2 HT on two logic cores share L1-d/L1-i/L2/TLB Cache
- Eliminate the need to BIND the online service and offline jobs to exclusive cores

Core	Neighbor HT		
Idle	Idle	job run	
Idle	Service run	Job not run	
Job run	Service wakeup	Job leave	

Alibaba Group

LLC(last-level cache) Interference

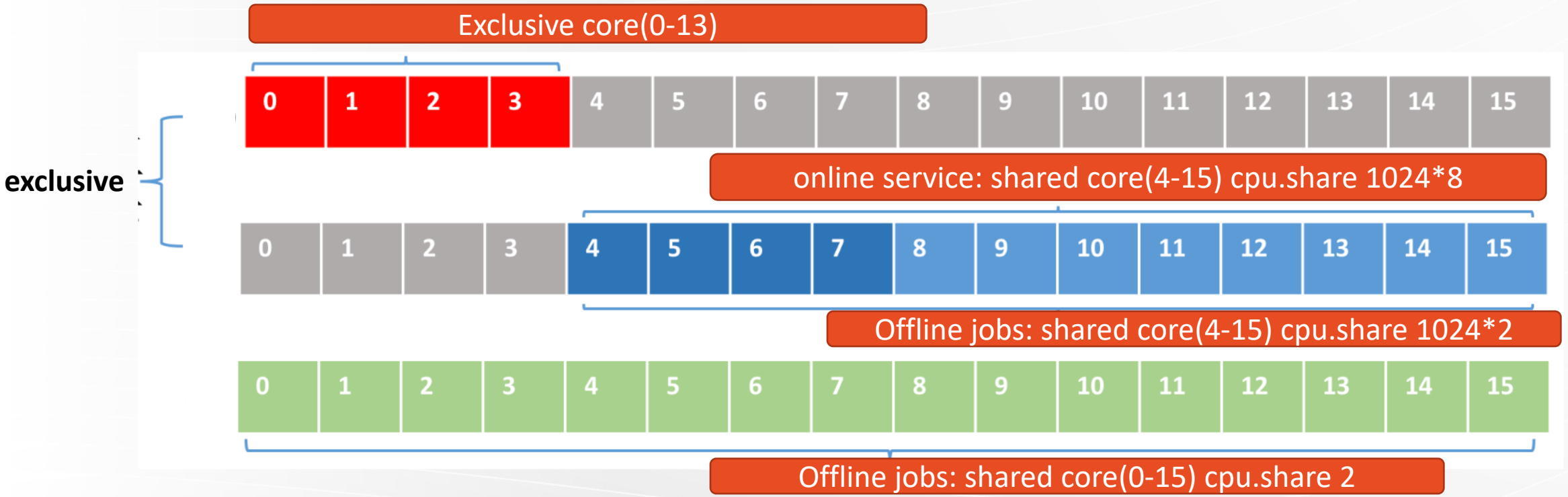
- Shared in the same CPU socket
- Intel CAT for LLC isolation
- 20 cache unit (16 for service, and 4 for job)

Alibaba Group

Memory Isolation

- Memory.priorty
  - Online service memory reclaim will be less than offline job
- memory.use_priority_oom
  - Kill lower priority offline jobs when OOM
- memcg wmark_ratio
  - Similar to kswapd, when memory hits wmark_ratio, start memory reclaimation

CPU

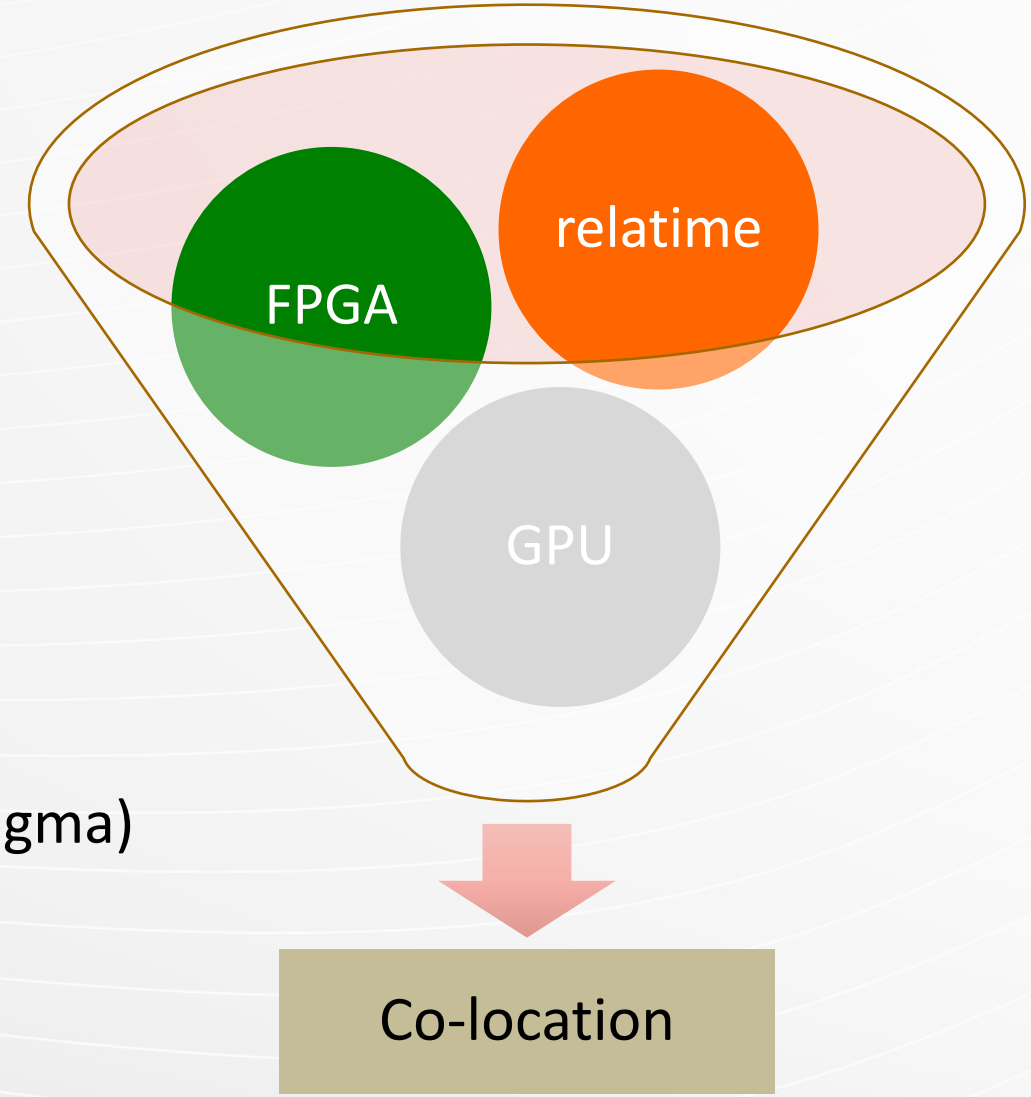

Exclusive core(0-13)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
exclusive
online service: shared core(4-15) cpu.share 1024*8
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Offline jobs: shared core(4-15) cpu.share 1024*2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Offline jobs: shared core(0-15) cpu.share 2

Alibaba Group

Future

Alibaba Group

Larger scale



- More resource dimension
- Expand Alibaba internal co-location scale (Fuxi & sigma)

Alibaba Group

Intelligent auto-rebalancing

- Learn online service and offline jobs and do prediction to auto-rebalance
- Adjust service and job cluster scale more efficiently on sales day events.

Alibaba Group

More

- Improve stabilities at large scale and high pressure.
- Consolidate the colocation technologies apply those in kubernetes & YARN co-location project