LINUXCON containercon CHINA 中国 CLOUDOPEN

THINK OPEN

开放性思维

A Novel Flow Network Graph Based Scheduling Approach in Kubernetes

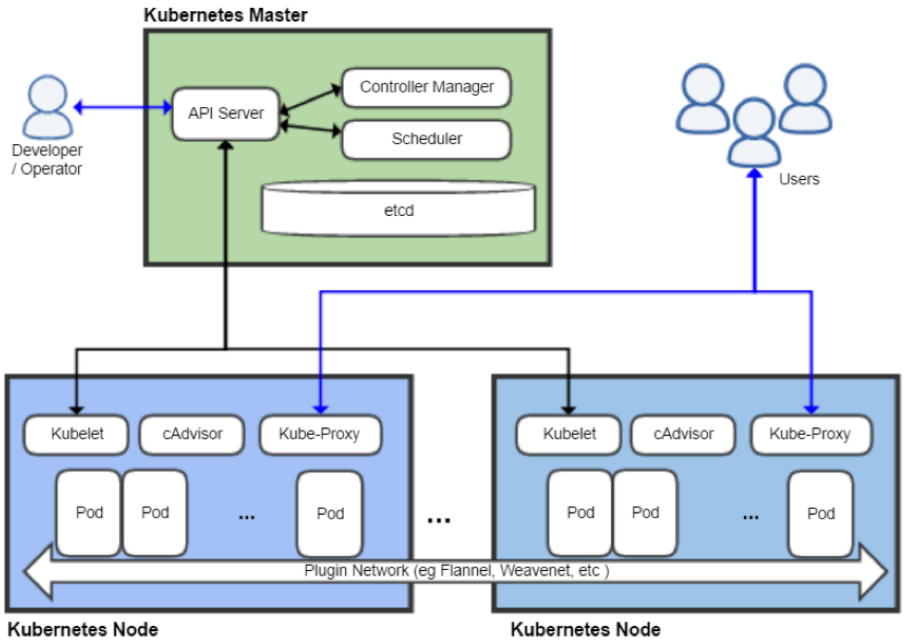
@kevin-wangzefeng
wangzefeng@huawei.com

LF ASIA, LLC

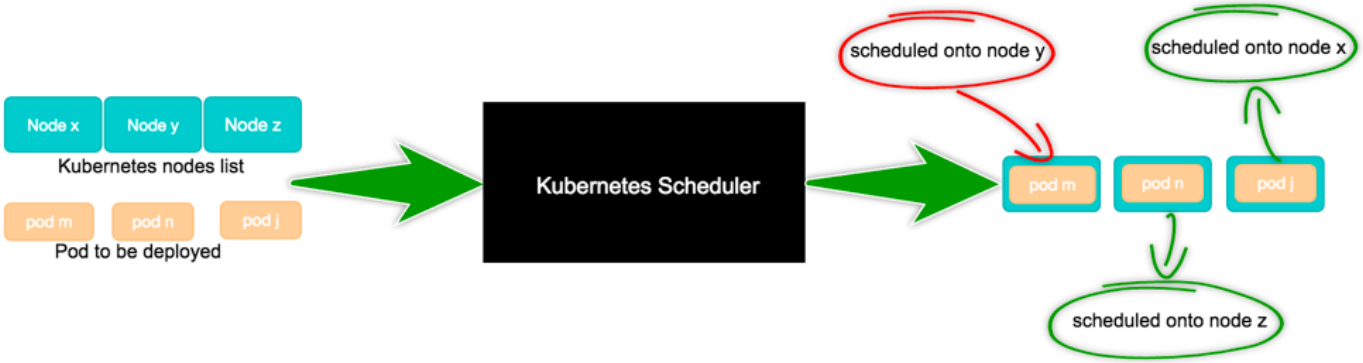
Agenda

- Scheduling in K8S
- The Default Scheduler
- Firmament & Poseidon
- Future plans

Scheduling in Kubernetes

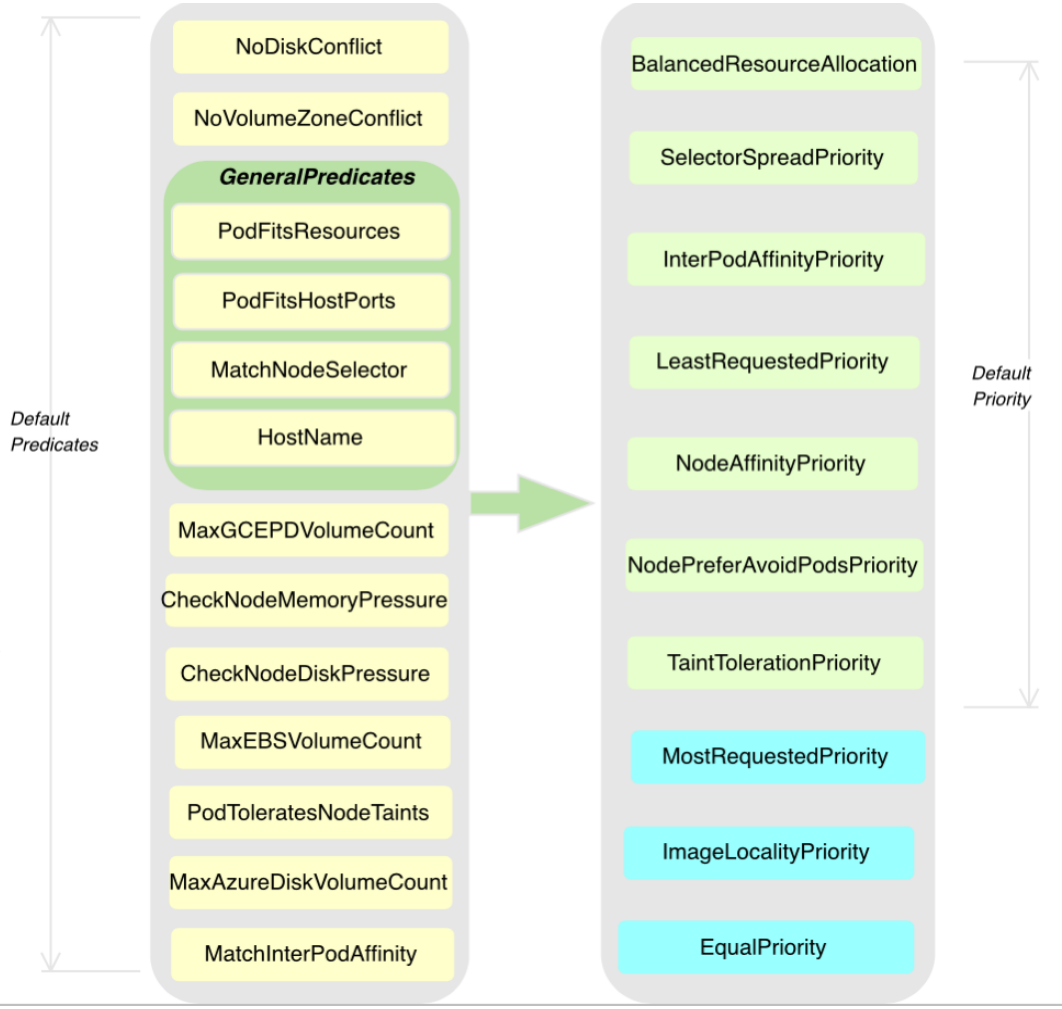



```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: my-pod
  name: my-pod-76559f5d5b-19b9p
  namespace: default
......
spec:
  dnsPolicy: ClusterFirst
  nodeName: node1
  restartPolicy: Always
  schedulerName: default-scheduler
  containers:
......
```



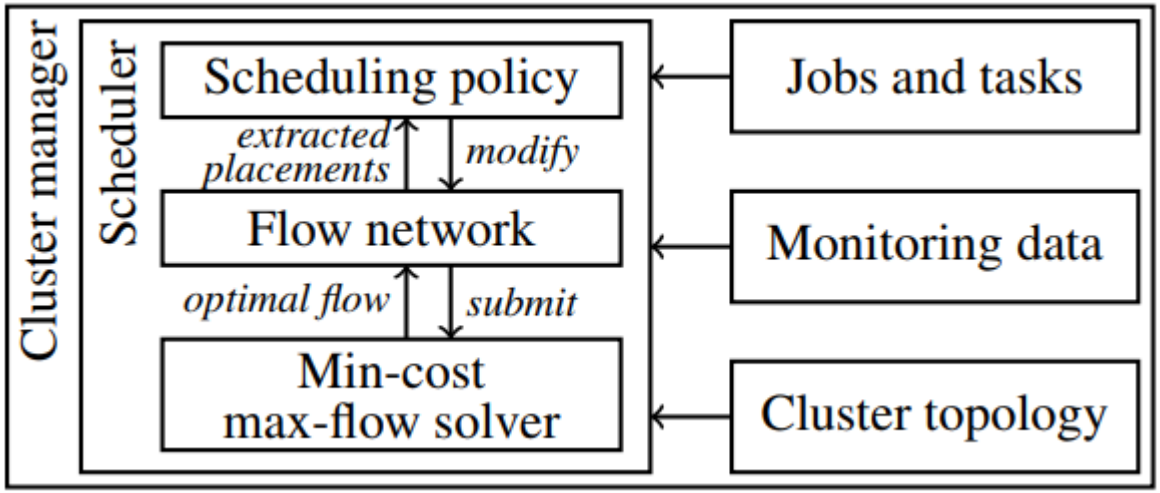
The default scheduler

- Queue based
  - one pod one time
  - best fit (scheduling time)
- Resource allocation model
  - Request based, not real-time usage
  - Low utilization (due to uncertain user resource estimation)
- Policies implemented as two sets of algorithms:
  - Predicates
  - Priorities



What is Firmament

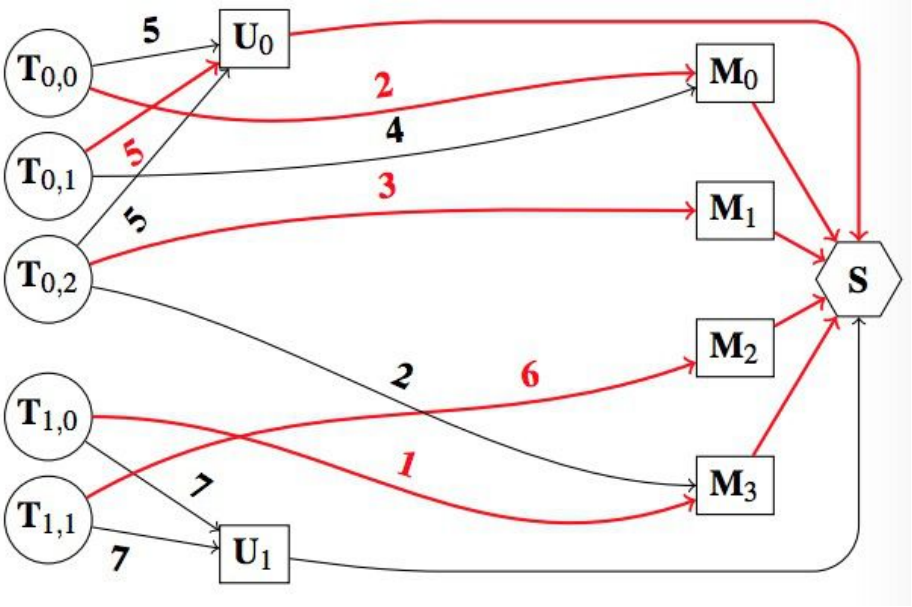
- Flow based scheduler
  - Models workloads and cluster as a flow network (DAG)
  - Policies considered at DAG build / update
  - Run Min-Cost Max-Flow (MCMF) solver to find an optimal flow
  - Scheduling results extracted from the optimal flow



diff Firmament Kube-scheduler

- Similar to default scheduler
  - “Global optimal solution”
  - Pluggable scheduling policies
- That makes differences
  - Flexible resource modeling, easy to extend to support topology (zones, racks, NUMA, etc.)
  - Built-in support with rescheduling, priority and preemption
  - And a set of other cost models:
    - network-ware, Quincy, load-spreading etc.
  - Low decision latency at scale
    - sub-second decisions at 10k+ machines
  - batching approach
  - By default use resource utilization instead of reservation

Flow network example in Firmament

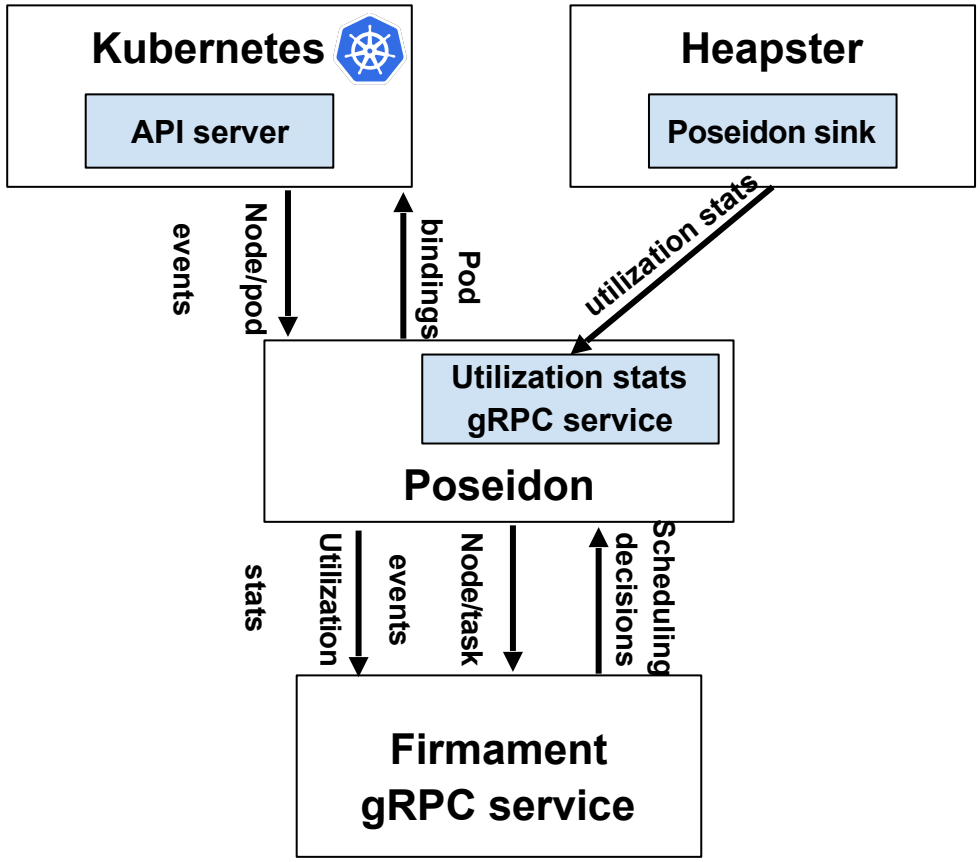


- Flow network
  - 4 machine cluster, 2 jobs (3 tasks and 2 tasks).
- Arc labels show non-zero costs
  - (values depends on policies.)
- All arcs have unit capacity
  - except those between unscheduled aggregators and the sink.
- The red arcs carry flow and form the min-cost solution.
  - All tasks except T0,1 are scheduled on machines.

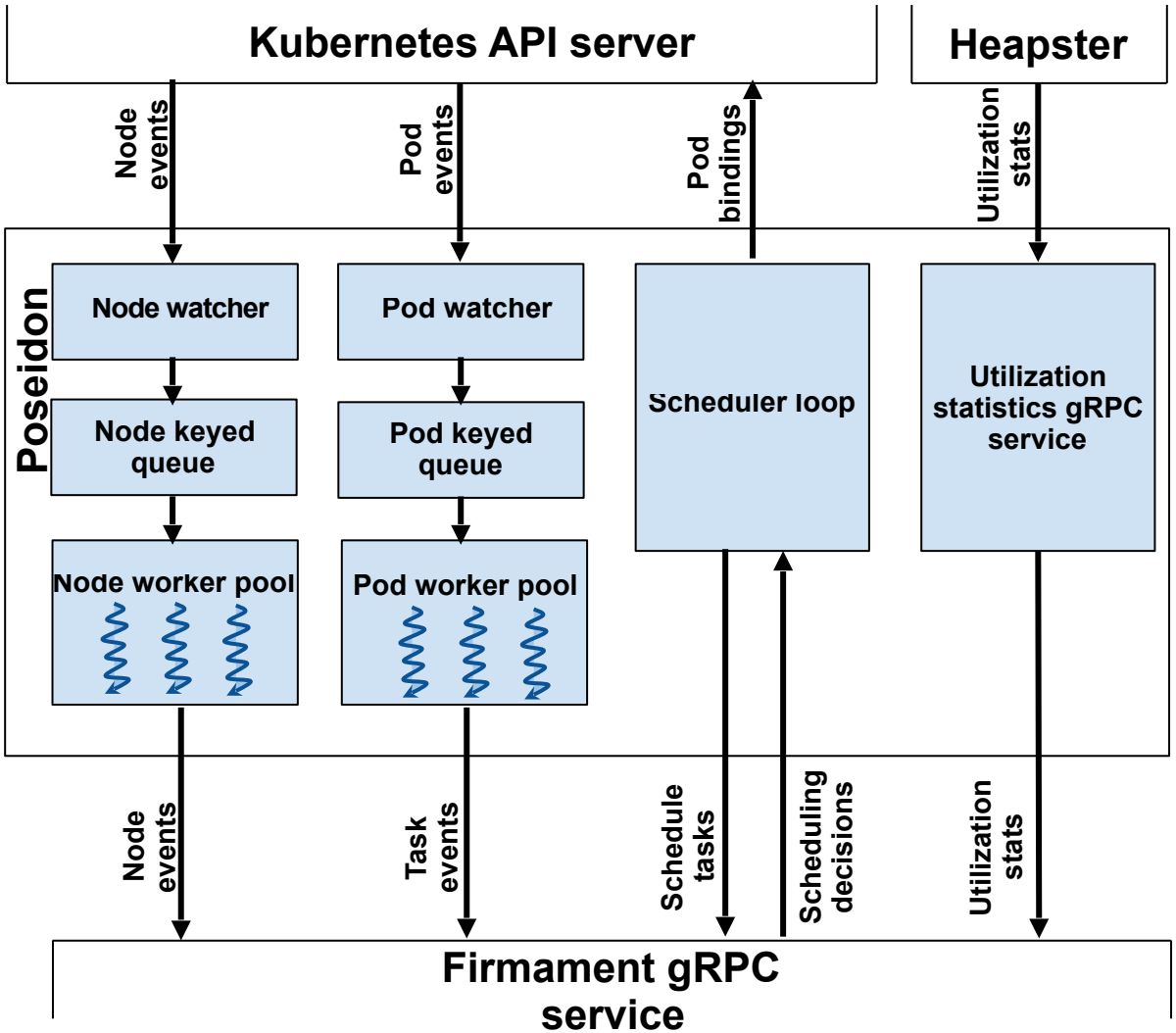
And Poseidon?

To fill the gaps between K8S and Firmament

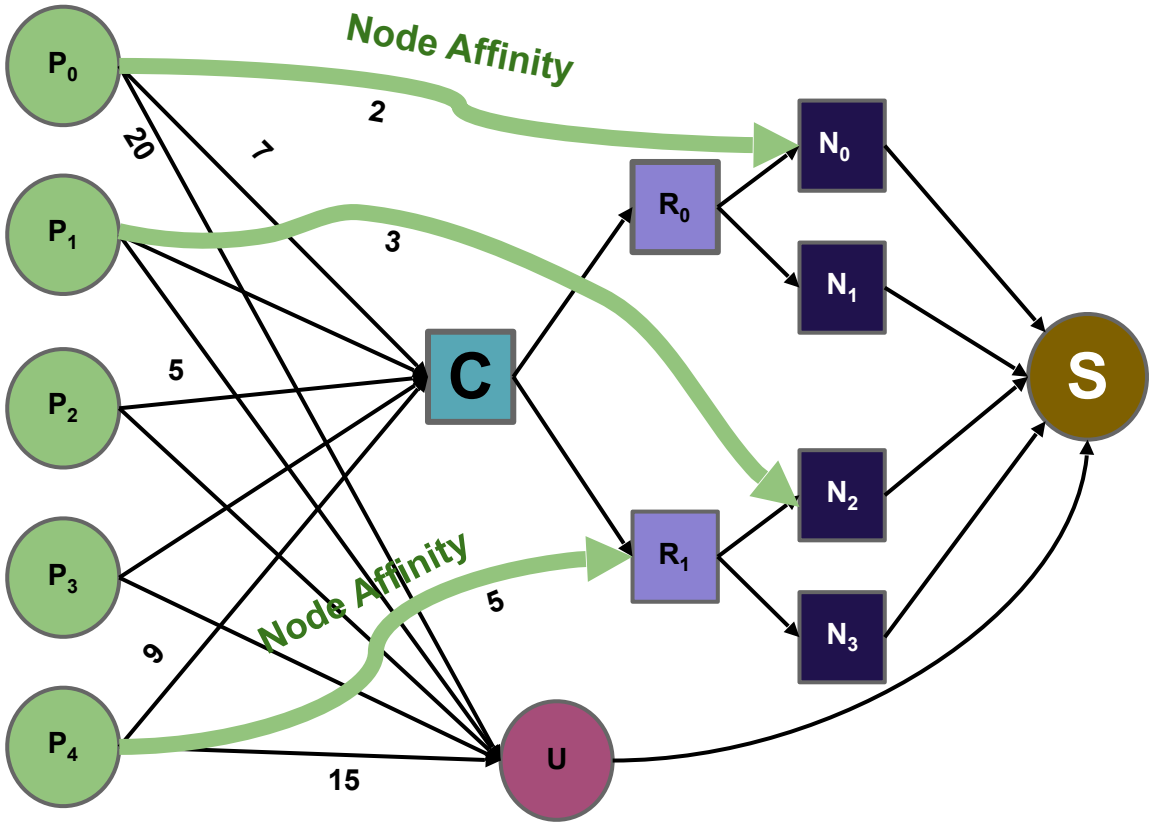
- Different concepts
  - K8S: workloads, pods
  - Firmament: jobs, tasks
- Different language
  - K8S: Golang
  - Firmament: C++
- Resource Requests v.s. Real-time utilization
  - K8S: allocate by requests and “un-claimed”
  - Firmament: utilization statistics



Poseidon Design



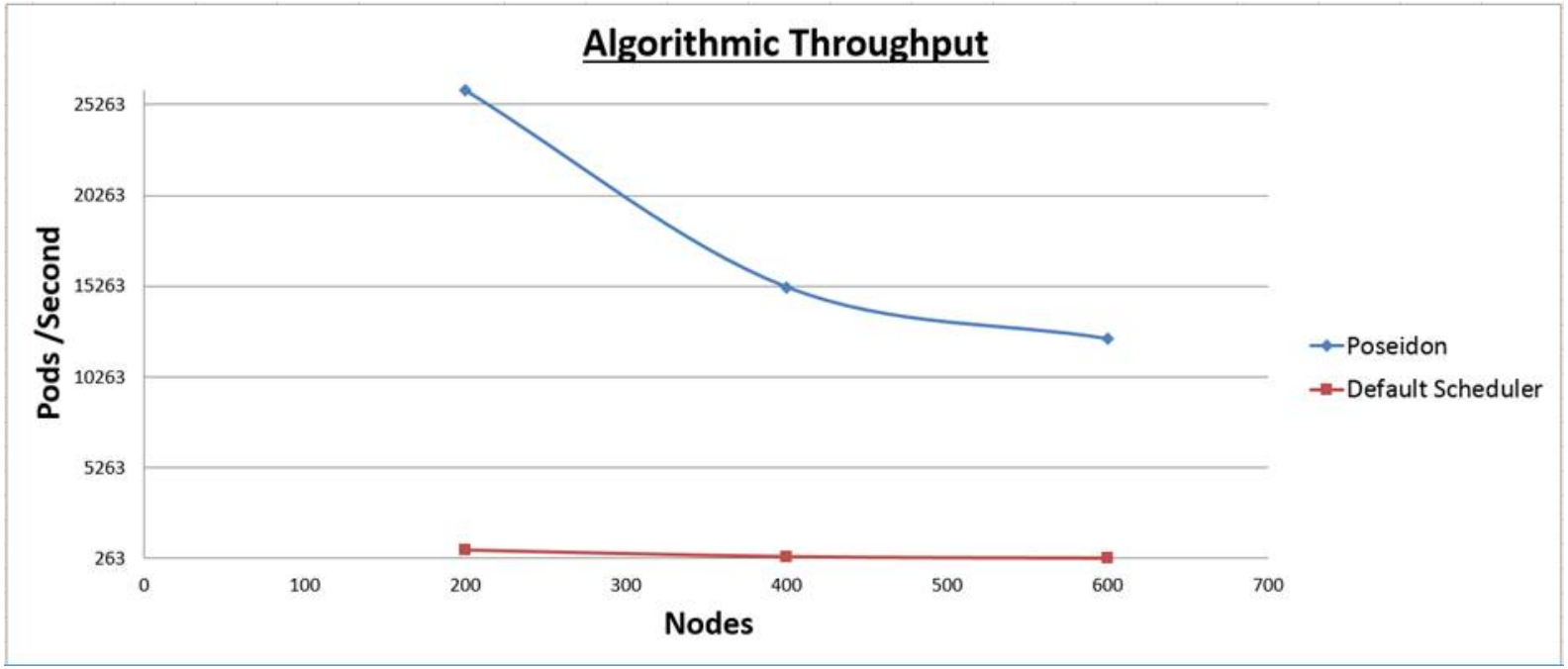
Flow Network aligned to Kubernetes Concept



Status and Progress

- Incubating under K8S scheduler SIG
  - https://github.com/kubernetes-sigs/poseidon
  - Currently Alpha (v0.3)
  - Support CPU/Memory Cost model
  - Node Affinity/Anti-Affinity
  - Pod Affinity/Anti-Affinity
  - Automation for E2E tests, PR process etc.
  - and more…

30X algorithmic throughput



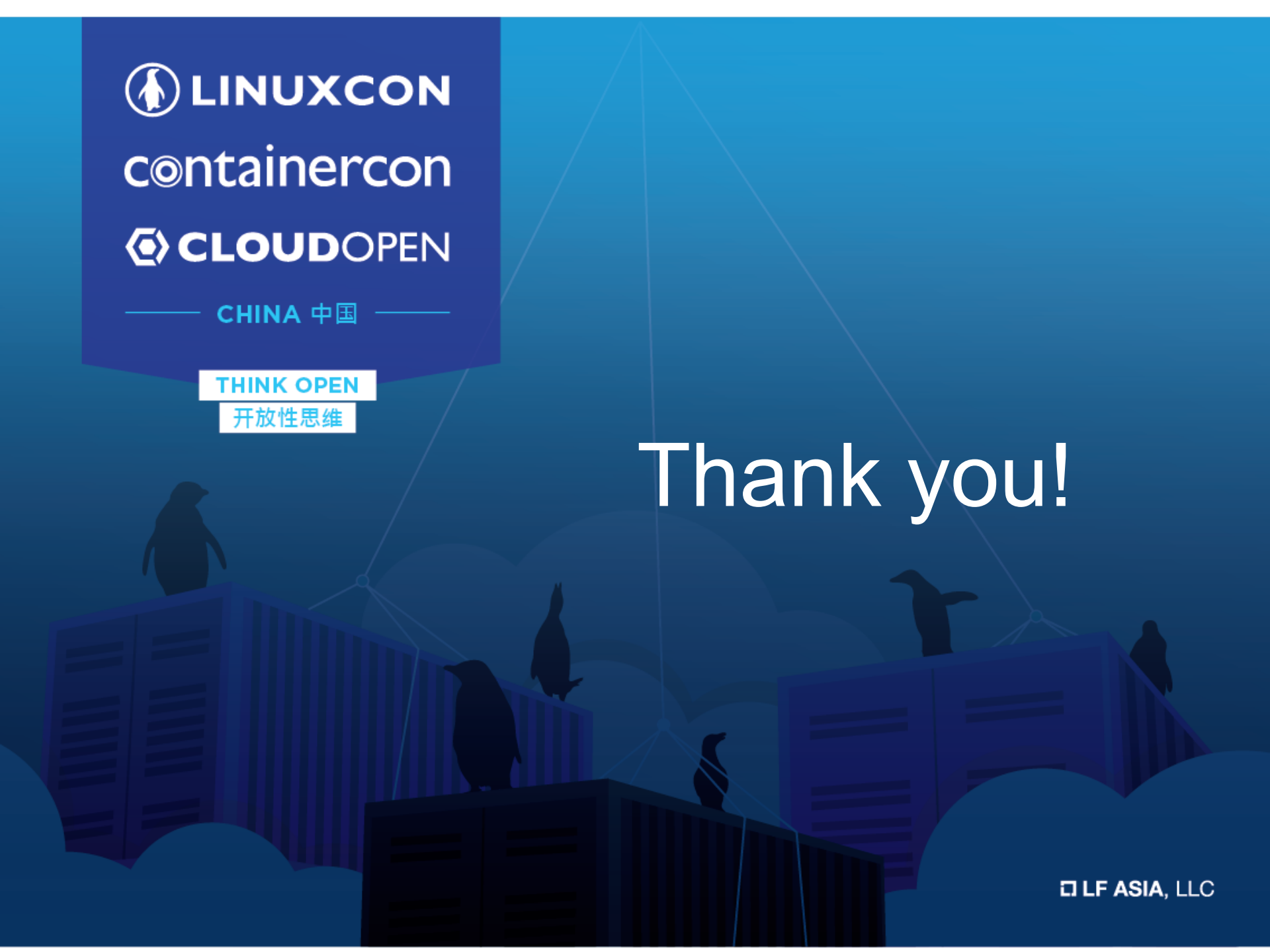
No	Nodes	Pods	Poseidon	Default Scheduler
1	200	3800	26027	761
2	400	7600	15200	361
3	600	11400	12351	265

Future plans

- Under development
  - Max allowed pods for nodes.
  - Taints & Tolerations.
  - Another round of benchmarking for scalabilities, performances.
- Longer future:
  - Transitioning to Metrics server API (Heapster is going to be deprecated).
  - High Availability / Failover for in-memory Firmament/Poseidon processes.
  - Priority Pre-emption support.
  - Gang Scheduling.
  - Resource Utilization benchmark.
  - Better cooperating with the default scheduler. (enhancements on multi-scheduler framework)
  - Checkout https://github.com/kubernetes-sigs/poseidon/issues for more…

Join us!

- Scheduling SIG
  - https://groups.google.com/forum/#!forum/kubernetes-sig-scheduling
- Poseidon Project
  - https://github.com/kubernetes-sigs/poseidon
- Follow Huawei Container team on WeChat


LINUXCON
containercon
CLOUDOPEN
CHINA 中国
THINK OPEN
开放性思维
Thank you!
LF ASIA, LLC


LINUXCON
containercon
CLOUDOPEN
CHINA 中国
THINK OPEN
开放性思维
LF ASIA, LLC