
KubeCon
CloudNativeCon
North America 2024


KubeCon | CloudNativeCon

North America 2024

Service Profiling Based Resource Management and Scheduling

Jia Deng @ ByteDance
Mingmeng Luo @ ByteDance
Cong Xu @ ByteDance

Overview

Project Introduction

Service Profiling

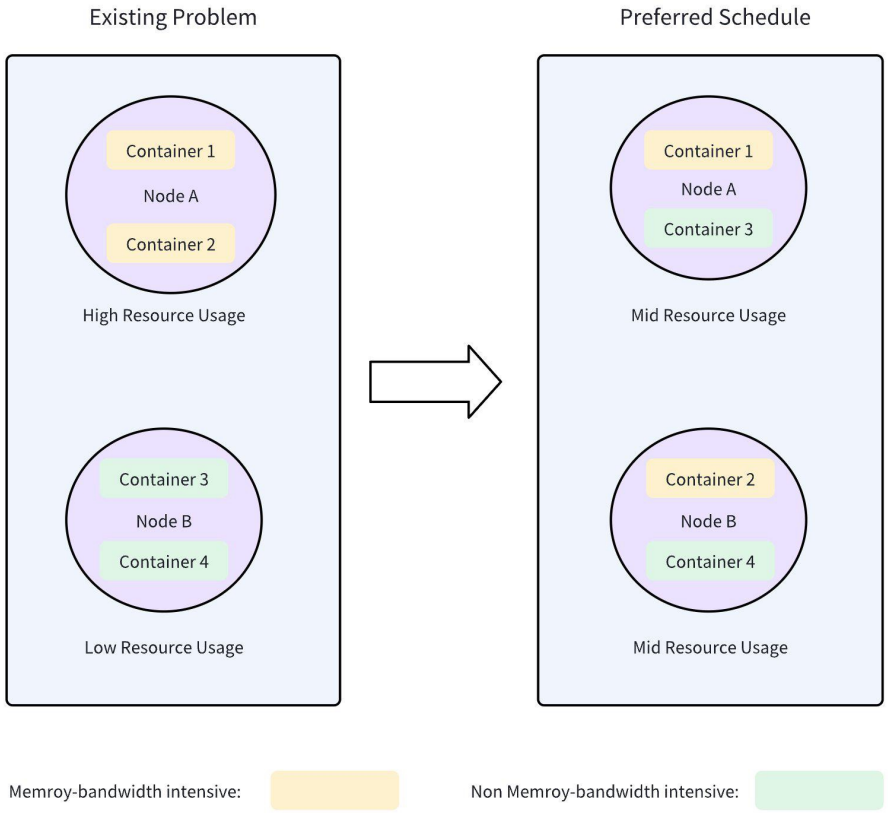
Scheduling

Katalyst

Future Plan

Project Introduction

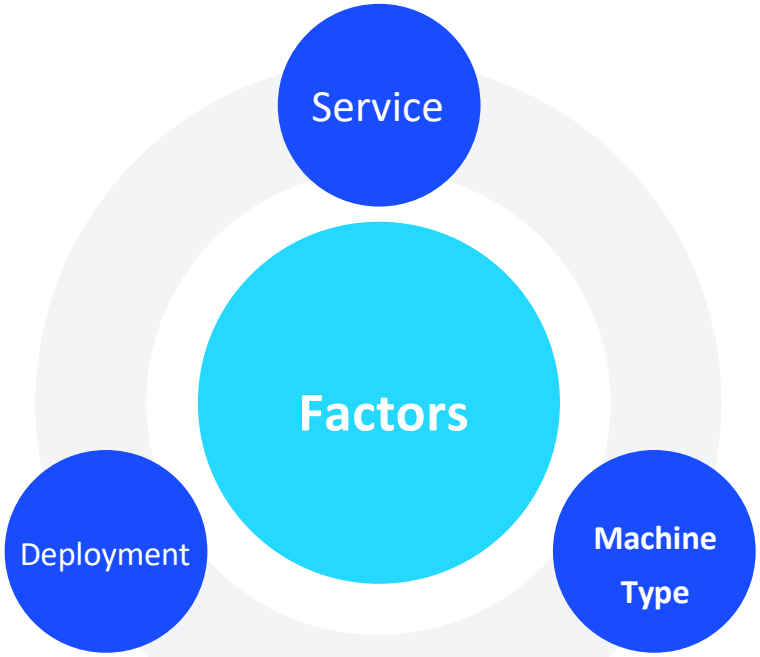
Project Introduction



- **Managing Various Machine Types and Workload Types in One Cluster**
- **Scheduling on K8S Unsupported Resources**
  - Disk I/O
  - Memory Bandwidth
  - etc
- **Hard to Quantify by Users**

Service Profiling

Workload Identifier



cpu request
memory request
env variables

eg:
Intel 5th generation
Intel 6th generation
AMD genoa 4th generation

- **Workload + Deployment:**
  - mb_per_core = (read_bw + write_bw)/cpu_request
  - Predicted MB: mb_per_core * cpu_request

deployment	cpu_request	mem_bw_per_core
	30.0	
	30.0	1537.10769807759
deploy_A	30.0	1528.7955716419256
	30.0	1517.9658808135441
	30.0	
	30.0	1924.4579527495557
	30.0	1967.8622223568912
deploy_B	30.0	1937.2548225453947
	30.0	1933.0827505650968
	46.0	1503.6473966063602
	30.0	1741.4002509273969
	30.0	1685.6591864851118
deploy_C	30.0	1762.3731886296341
	30.0	1756.5971375515662
	30.0	1677.5598082394074

mb_usage of the same workload

- **Machine Type:**
  - mb_per_core * machine_weight

Machine Generation Ratio

Machine Physical Memory Module Bandwidth

(Mega Transfer/Second) (Illustraive Data)

M1: 1000

M2: 2000

M3: 4000

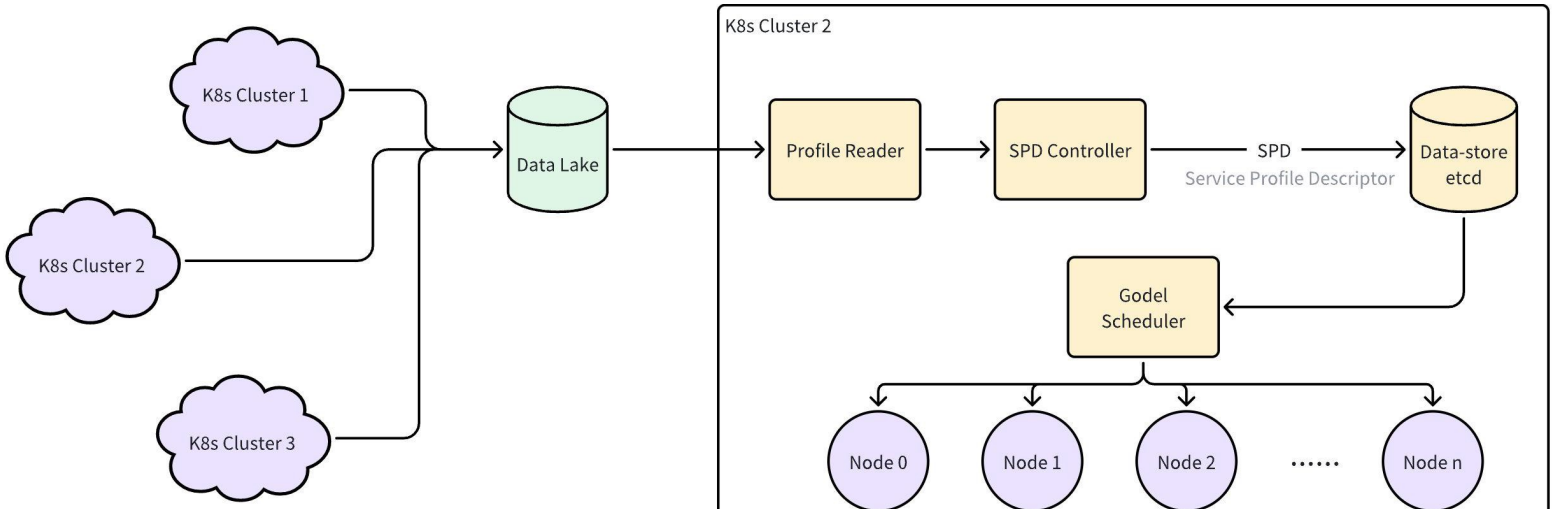
suppose workload W's profile on M1 = a,

workload W's predicted resource needed on M2 = 2a

Machine Memory Modual **Physical** Bandwidth Ratio			
	M1	M2	M3
M1	1	2	4
M2	0.5	1	2
M3	0.25	0.5	1

Scheduling

Scheduling

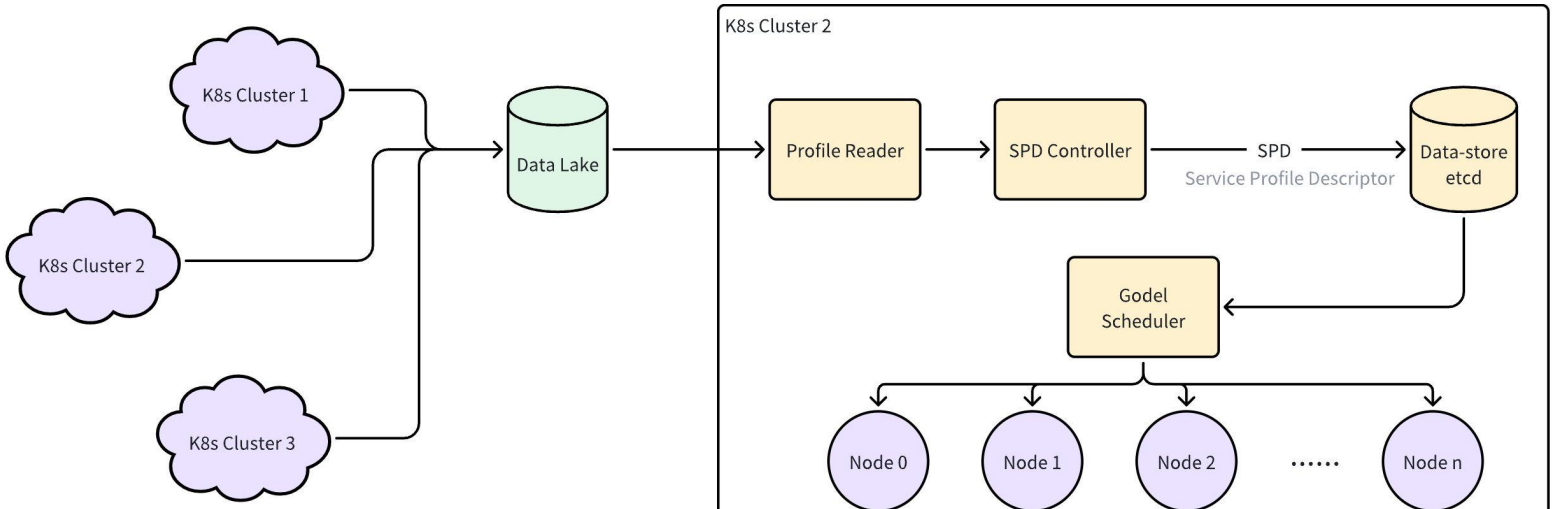


	MB Left	Rating
node 0	1000	0
node 1	1500	50
node 2	2000	100

Service Profile Descriptor

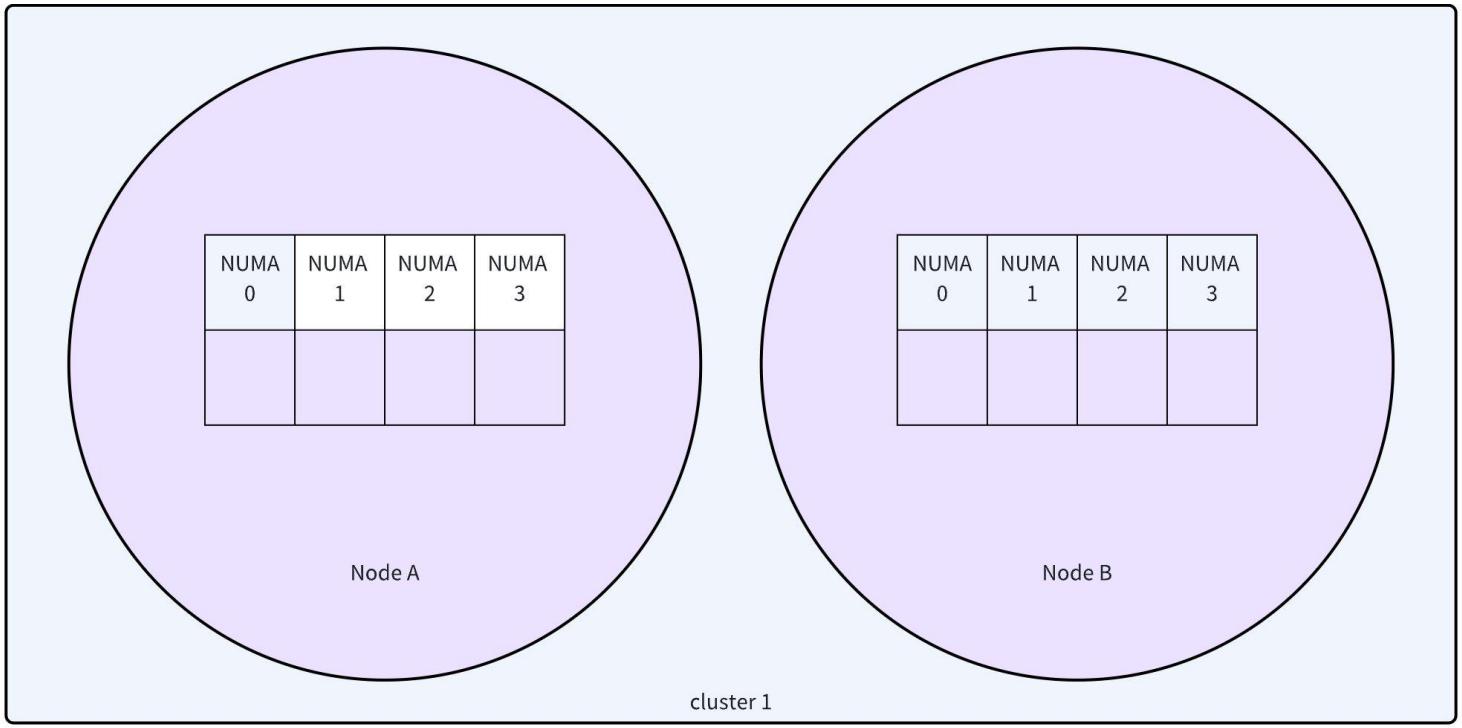

```
status:
  aggMetrics:
  - aggregator: avg
    items:
    - containers:
      - name: '*'
        usage:
          [redacted]
          memory_bandwidth-M1: [redacted]
          memory_bandwidth-M2: [redacted]
          memory_bandwidth-M3: [redacted]
          [redacted]
          [redacted]
      metadata: {}
      timestamp: "2024-08-22T16:00:00Z"
      window: 1h0m0s
```

Scheduling



	most MB Left NUMA	Rating
node 0	1000	0
node 1	1500	50
node 2	2000	100

Scheduling Example

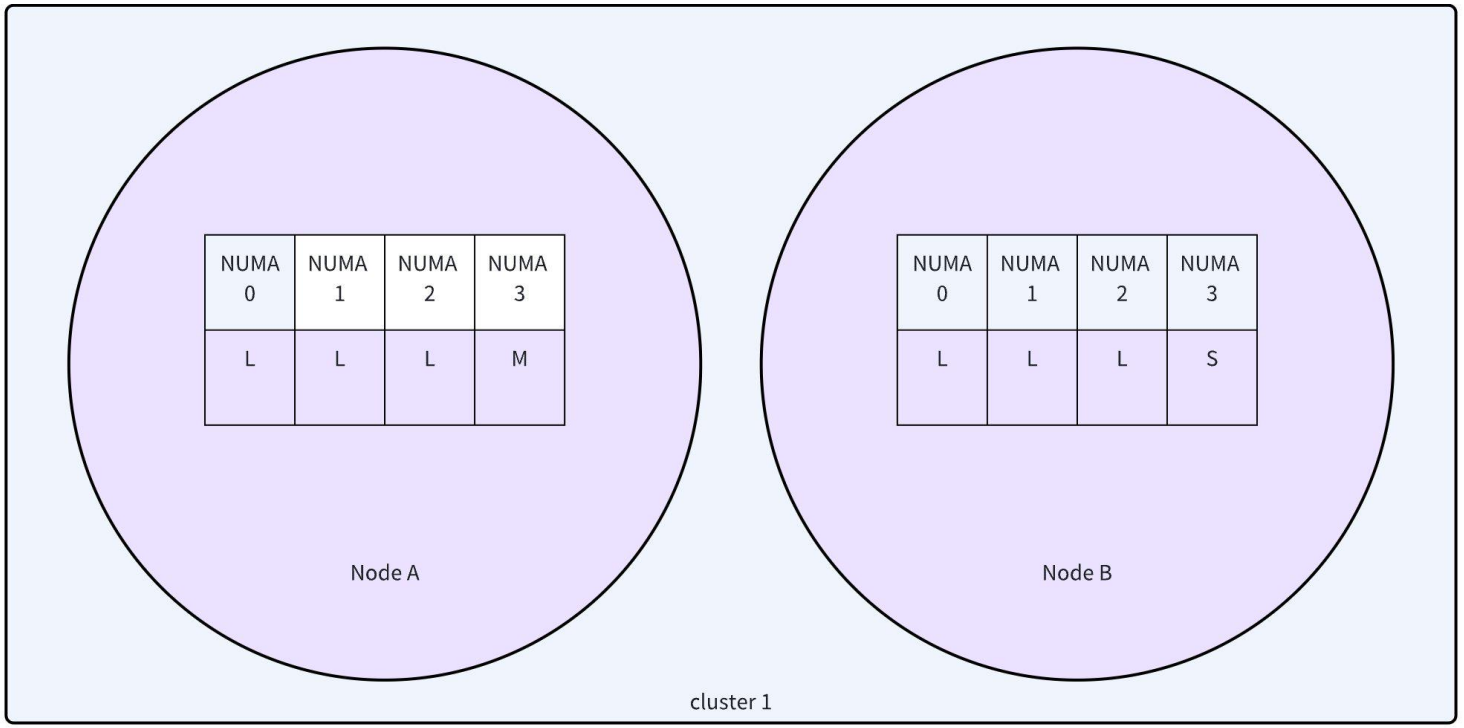


NUMA limit < Large workload

half NUMA limit < Medium workload < NUMA limit

Small workload < half NUMA limit

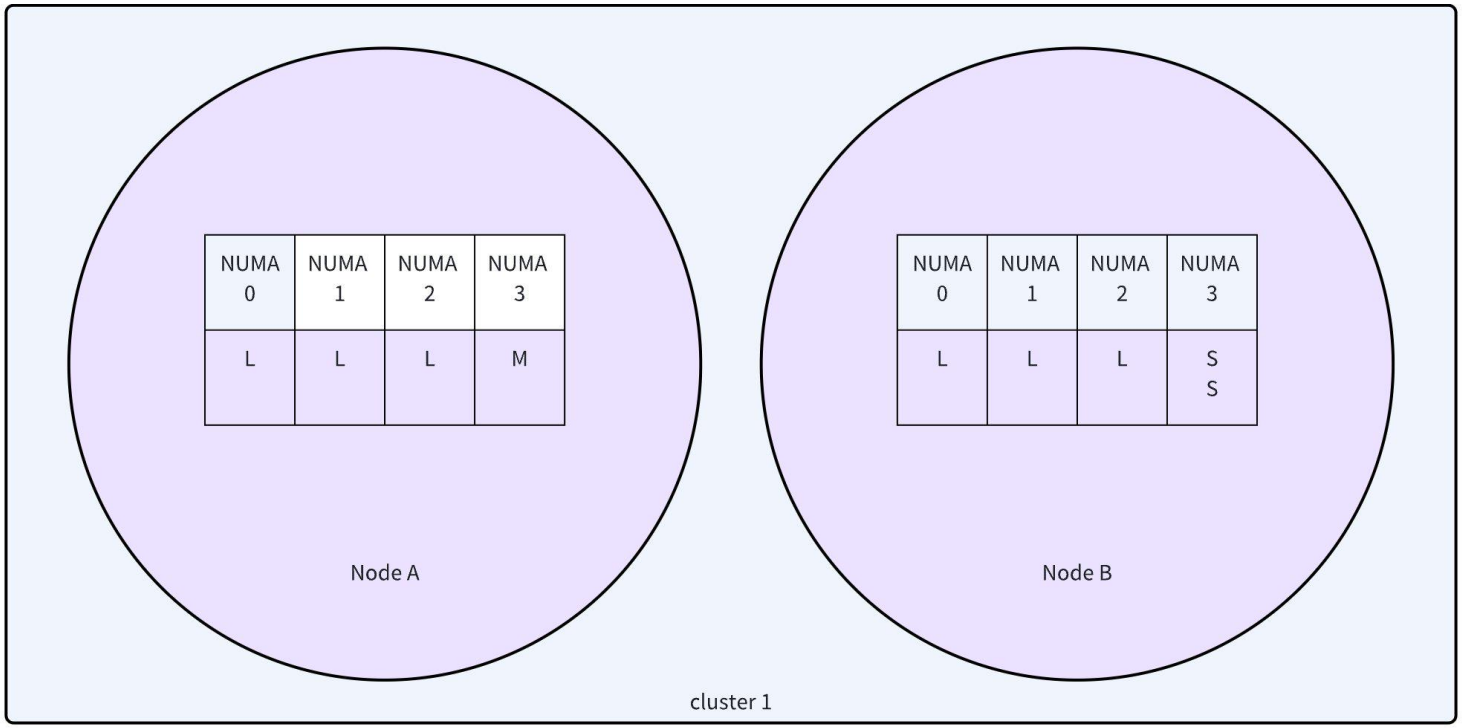
Scheduling Example



Schedule 3 large size pods + 1 small size pod, 1 by 1, on node A

Schedule 3 large size pods + 1 medium size pod, 1 by 1, on node B

Scheduling Example

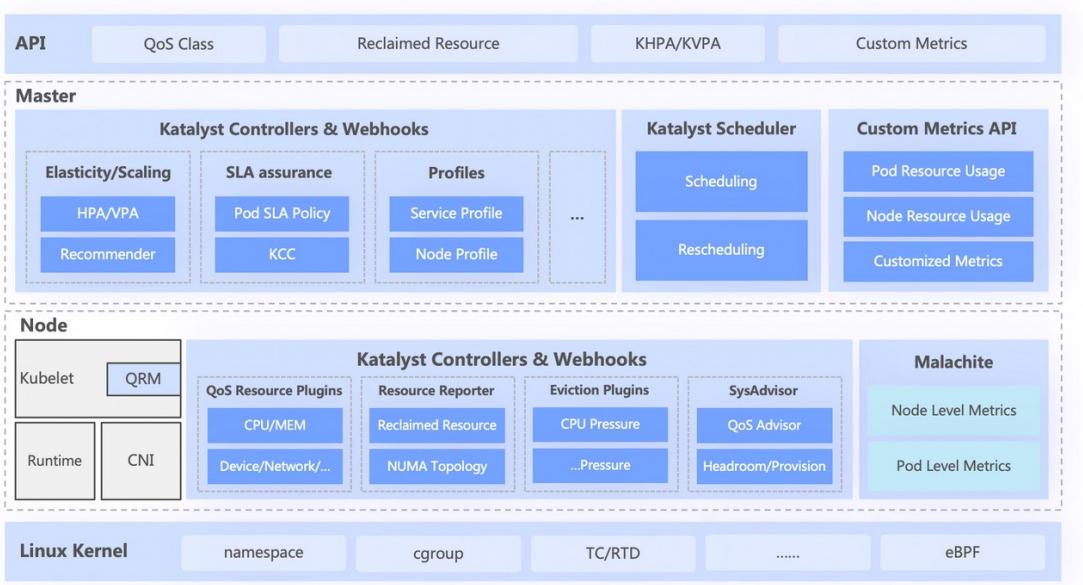


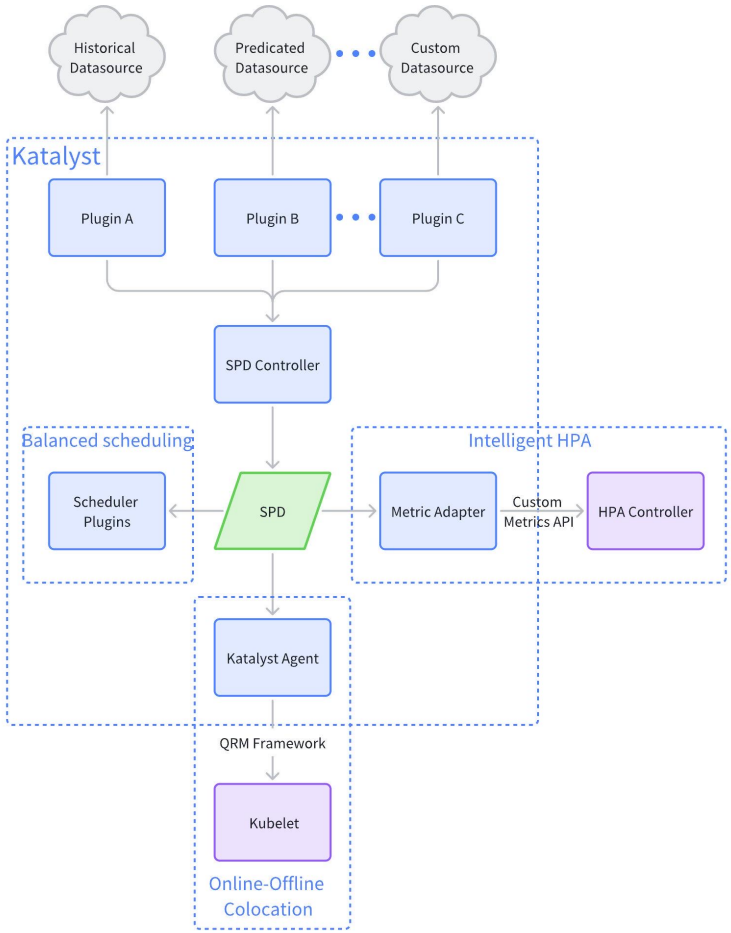
Schedule another small size workload

Katalyst

Katalyst: Resource Manage System

Katalyst and SPD Framework

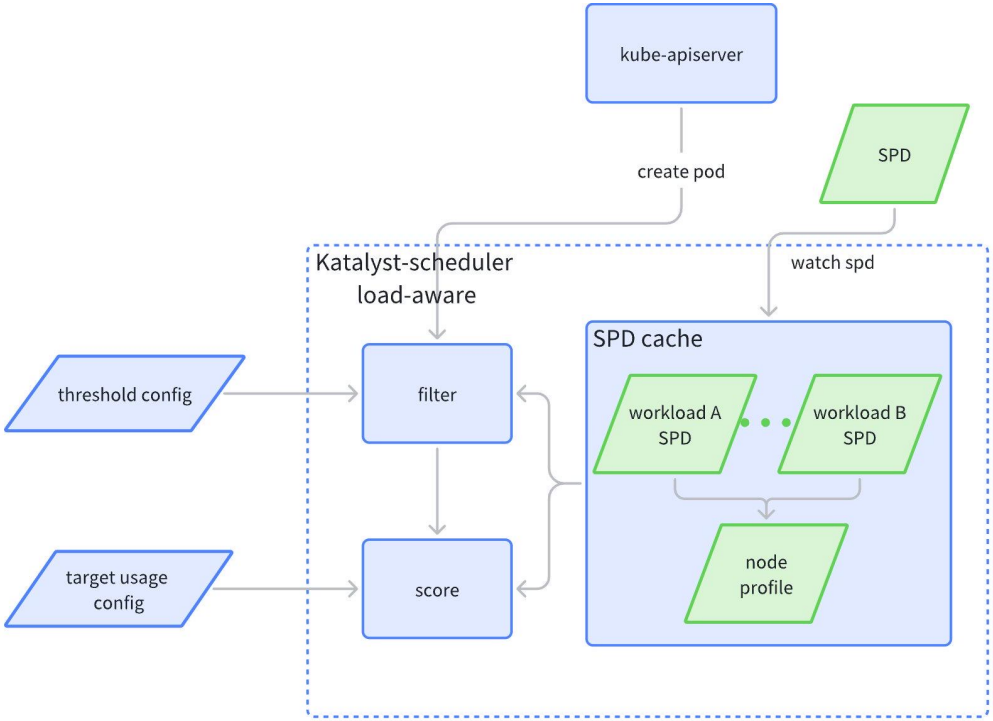




github.com/kubewharf/katalyst-core

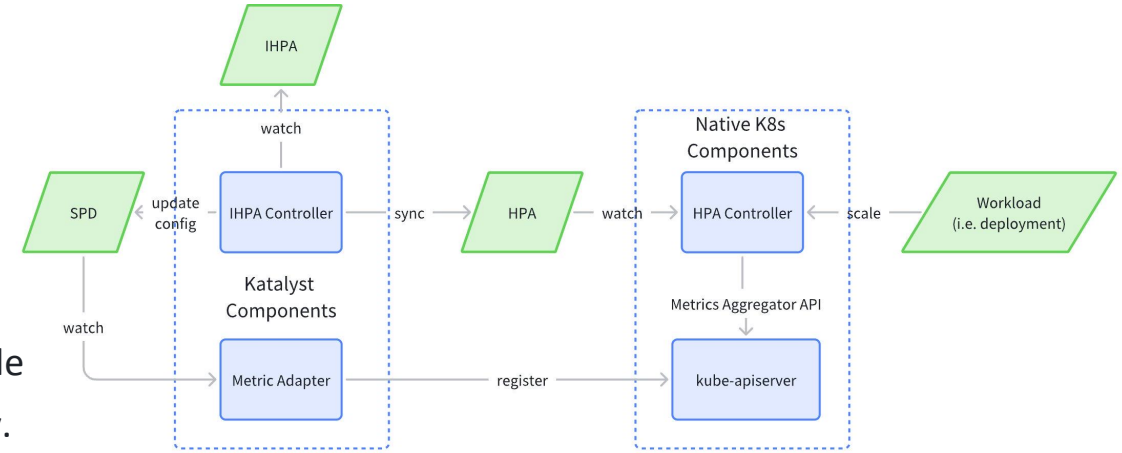
Case 1: Load-Aware Balanced Scheduling

- **Native Scheduler Limitations:**
  - Relies on Pod Requests.
  - Ignores historical loads. Can cause resource hotspots.
- **Load-Aware Balanced Scheduling:**
  - Balances resource usage.
  - Uses profiling to optimize placement.



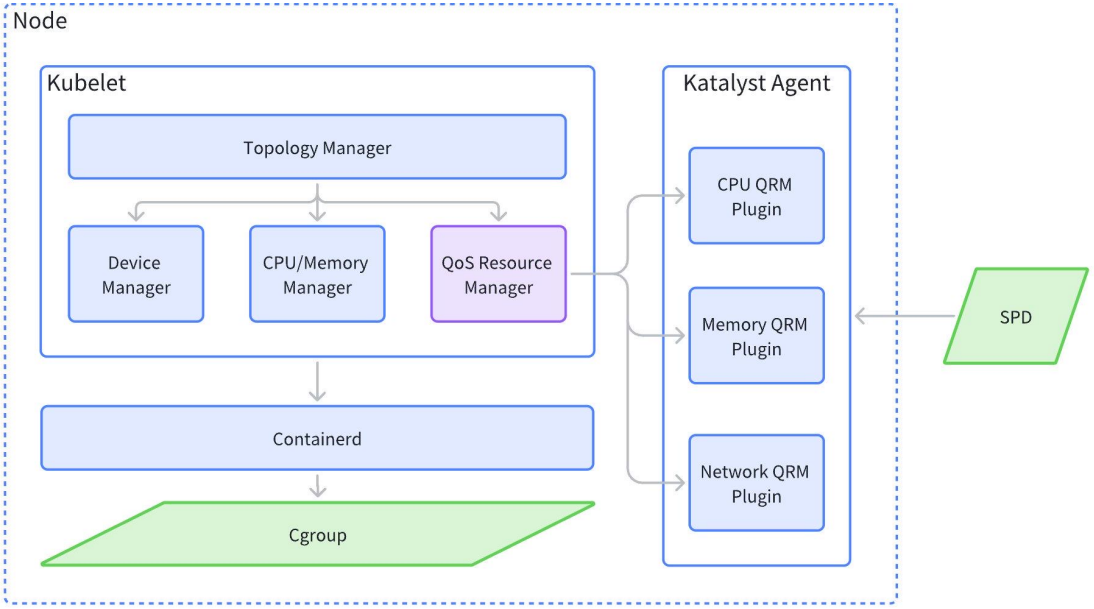
Case 2: Intelligent HPA

- **Service Profiling:** Dynamically generates profiles as external metrics, guiding timely and accurate scaling actions.
- **Dual Metrics:** Combines predictive and native metrics to enhance scalability and ensure stability.
- **Proactive Scaling**: Uses predictive metrics to scale workloads before demand spikes, minimizing delay.



Case 3: Online-Offline Colocation

- QRM (QoS Resource Manager)
  - Handling QRM plugins registration and enforcing real-time resource allocation and adjustments through the standard CRI interface.
  - Provides resource allocation suggestions to the Topology Manager.
- SPD for QRM plugins
  - Use SPD to understand the resource requirements of online workloads
  - Allocate remaining resources to offline tasks without impacting the performance of online services.



Future Plan

Future Plan

- More accurate profiling strategy
- Rescheduling based on actual usage
- Expand to other resources
  - network bandwidth
  - disk i/o
  - power
  - etc

github.com/kubewharf/katalyst-core