

# ??

□□□□□

- [CAP](#)
  - [CAP □□](#)
- [eureka□ zookeeper□ nacos□□□□](#)
  - [eureka□ zookeeper□ nacos □□□](#)
- [□□](#)
- [SAP □□□](#)

CAP

CAP

# CAP ??

1. CAP Consistency Availability Partition tolerance



2. Consistency: Consistency Availability Partition tolerance

3.

Availability:



4.

Partition tolerance:



~ ~

eureka?zookepeer?nacos?????

# eureka? zookeeper? nacos ????

## 1. eureka

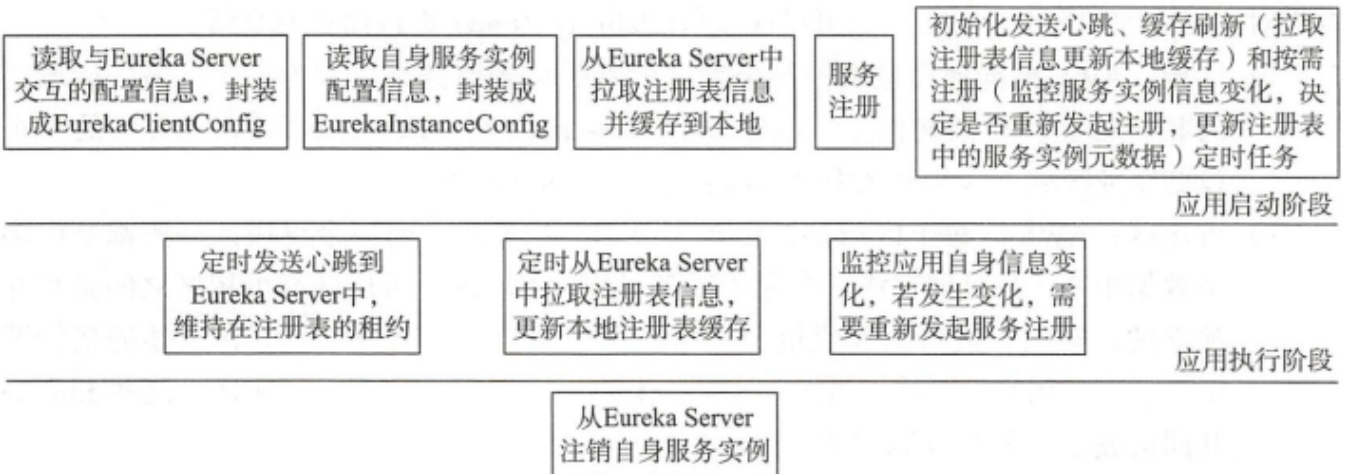


[https://blog.csdn.net/weixin\\_45018587](https://blog.csdn.net/weixin_45018587)

eureka: [ ]

[ ] : [ ] eureka [ ]

[ ] : [ ] eureka [ ] httpclient [ ] eureka [ ]



[http://blog.csdn.net/weixin\\_45018587](http://blog.csdn.net/weixin_45018587)

1.2.1

```

1. 1.2.1 eureka server
2. 1.2.1 InstanceInfo
server
3. 1.2.1 eureka server
4. 1.2.1 InstanceInfo eureka server
5. 1.2.1

```

1.3

```

1. 1.3 Eureka Client
2. 1.3 30
3. 1.3 90 Eureka Server

```

1.4

```

1. 1.4 Eureka Client cancel
2. 1.4 Eureka Server cancel

```

1.5

```

1. 1.5 Eureka Server
2. 1.5 Eureka Server
3. 1.5 Eureka Server

```

## 2 zookeeper

```

1. zookeeper= +

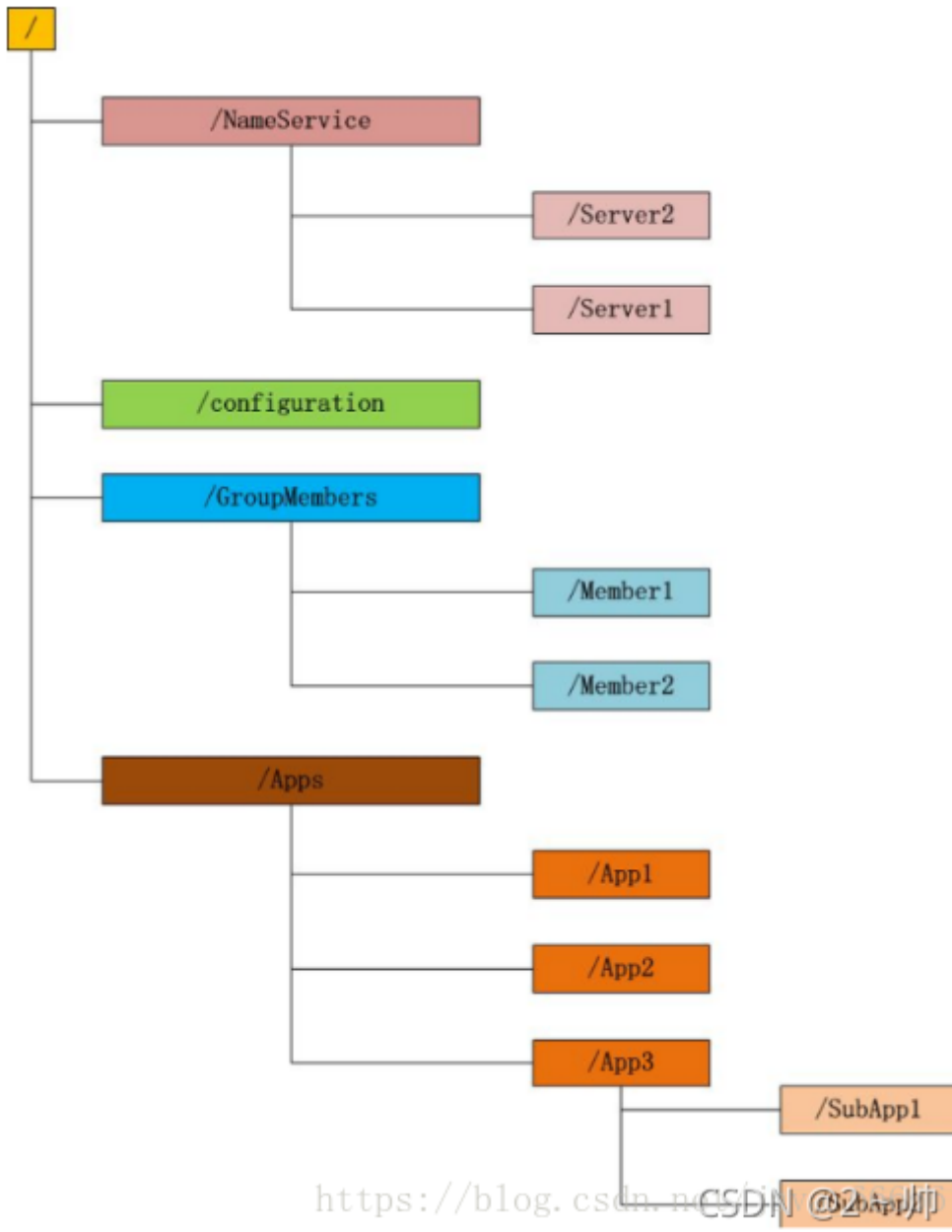
```

3.1

```

1. zookeeper

```



```

NameService znode( )
znode znode znode znode

```

```

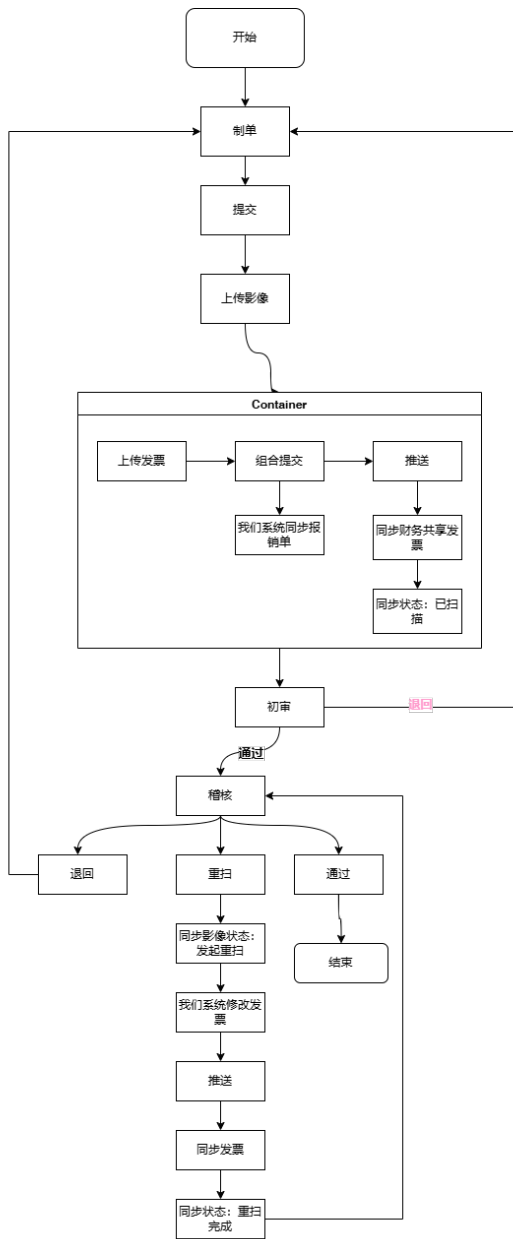
znode
1 PERSISTENT-
client zookeeper client zookeeper zookeeper
2 PERSISTENT_SEQUENTIAL-
user zookeeper zookeeper user
3 EPHEMERAL-
client zookeeper ,zookeeper
4 EPHEMERAL_SEQUENTIAL-

```



	Nacos	Eureka	Consul	CoreDNS	Zookeeper
一致性协议	CP+AP	AP	CP	—	CP
健康检查	TCP/HTTP/MYSQL/Client Beat	Client Beat	TCP/HTTP/gRPC/Cmd	—	Keep Alive
负载均衡策略	权重/ metadata/Selector	Ribbon	Fabio	RoundRobin	—
雪崩保护	有	有	无	无	无
自动注销实例	支持	支持	支持	不支持	支持
访问协议	HTTP/DNS	HTTP	HTTP/DNS	DNS	TCP
监听支持	支持	支持	支持	不支持	支持
多数据中心	支持	支持	支持	不支持	不支持
跨注册中心同步	支持	不支持	支持	不支持	不支持
SpringCloud集成	支持	支持	支持	不支持	支持
Dubbo集成	支持	不支持	支持	不支持	支持
K8S集成	支持	不支持	支持	支持	不支持

???

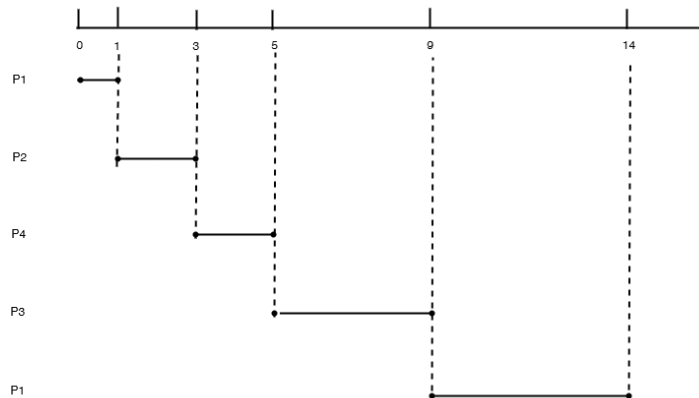


1. 最短作业优先算法定义

是一种抢占式的进程调度算法，其核心思想是根据进程的剩余执行时间动态选择最短者优先执行。

题 36 表

进程	到达时刻 t	运行时间
P1	0	6
P2	1	2
P3	1	4
P4	3	2



最短剩余时间算法

周转时间: 执行时间-到达时间

P1:  $14 - 0 = 14$

P2:  $3 - 1 = 2$

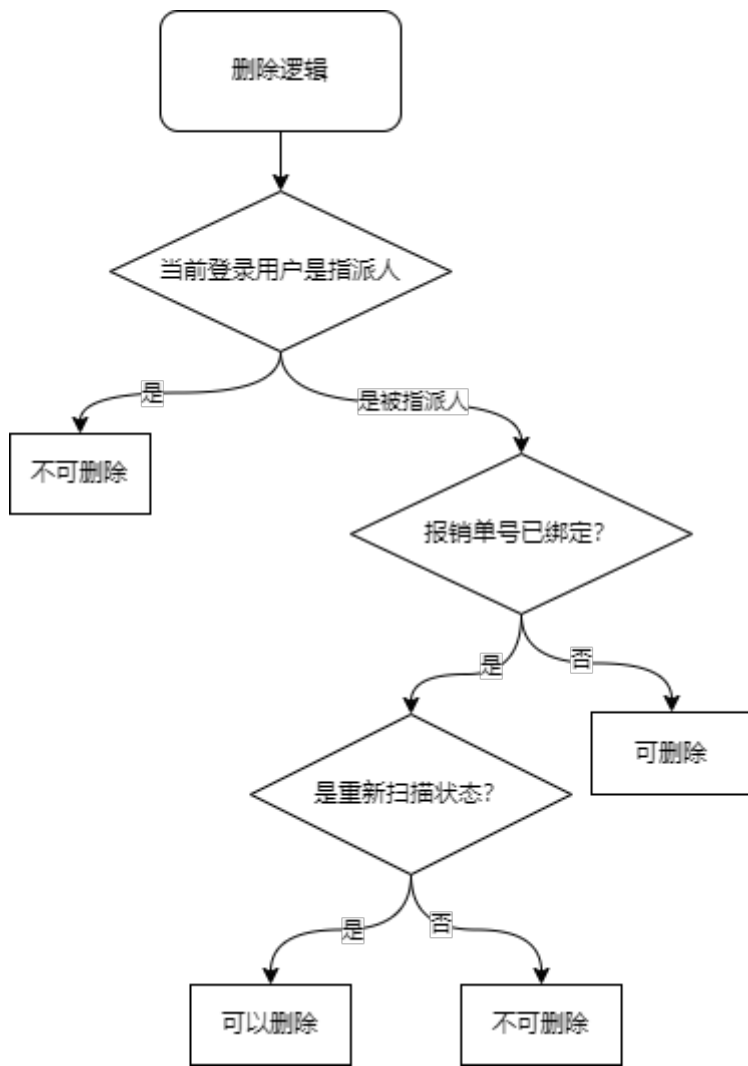
P3:  $9 - 1 = 8$

P4:  $5 - 3 = 2$

平均执行时间:  $(14+2+8+2) / 4 = 6.50$

1. 最短作业优先算法定义

SJF是一种非抢占式的作业调度算法，其核心思想是优先调度预计执行的作业最短的作业，以减少平均周转时间和提高系统吞吐量。若后来的短作业（进程）到达时，当前作业已开始执行，则不抢占，继续执行当前作业直到完成。



# SAP ????

