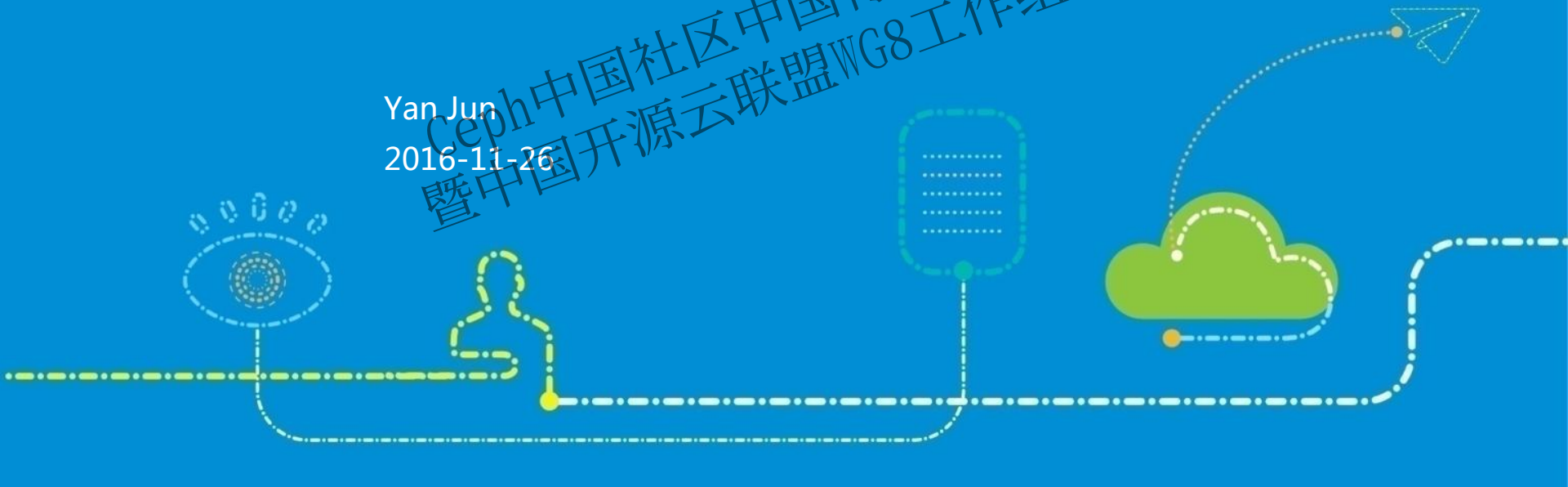
Ceph QoS based on dmClock

---基于 dmClock 的 Ceph Qos 功能实现

Yan Jun

2016-11-26



- 什么是QoS ？
  - 1）Quality of Service 服务质量
  - 2）对存储系统的 IO 资源控制机制，合理分配的策略
- Ceph为什么要用QoS？
  - 1）系统IO资源抢占，用户体验不好
  - 2）来自客户的应用需求

为什么是 dmClock

Algorithm class	Proportional allocation	Latency control	Reservation Support	Limit Support
Proportional Sharing (PS) Algorithms	Yes	No	No	No
PS + Latency support	Yes	Yes	No	No
PS + Reservations	Yes	Yes	Yes	No
mClock/dmClock	Yes	Yes	Yes	Yes

- https://github.com/ceph/dmclock.git

提要

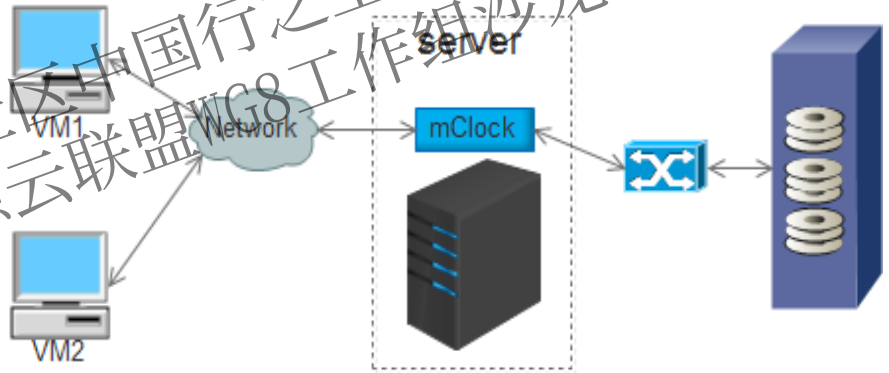
- dmClock算法
- Ceph中的QoS设计实现
- 结果和分析
- 后续工作

dmClock 算法

- mClock: Handling Throughput Variability for Hypervisor IO Scheduling
  - 预留：reservation
  - 权重：weight
  - 上限：limit

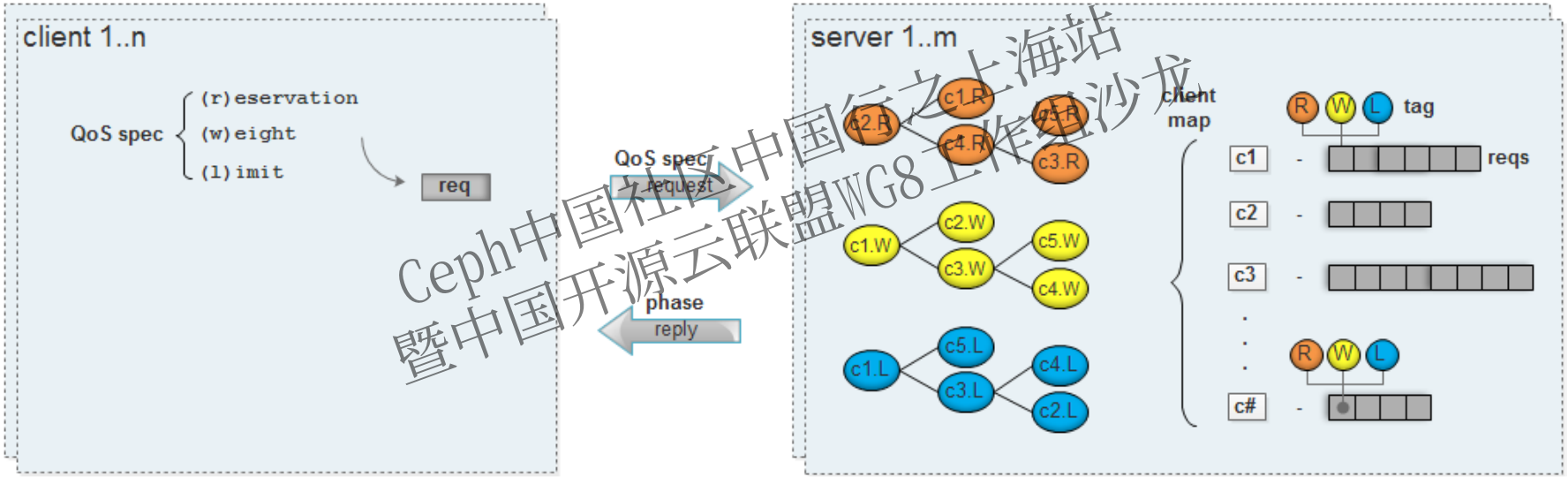
a) constraint - based phase

b) weight - based phase



dmClock 算法

- distributed mClock



Tag 计算

Symbol	Meaning
r_i	reservation of client c_i
w_i	weight of client c_i
l_i	limit service allowance for c_i
R_i^r	Reservation tag of req r from c_i
W_i^r	Weight tag of req r from c_i
L_i^r	Limit tag of req r from c_i

- request tag

$$tag_i^r = \max\{tag_i^{r-1} + n_i * 1/q_i, time\}$$

$$tag_i^r \in \{R_i^r, W_i^r, L_i^r\}; n_i \in \{\rho_i, \delta_i\}; q_i \in \{r_i, w_i, l_i\}$$

- clients weight tag adjustment

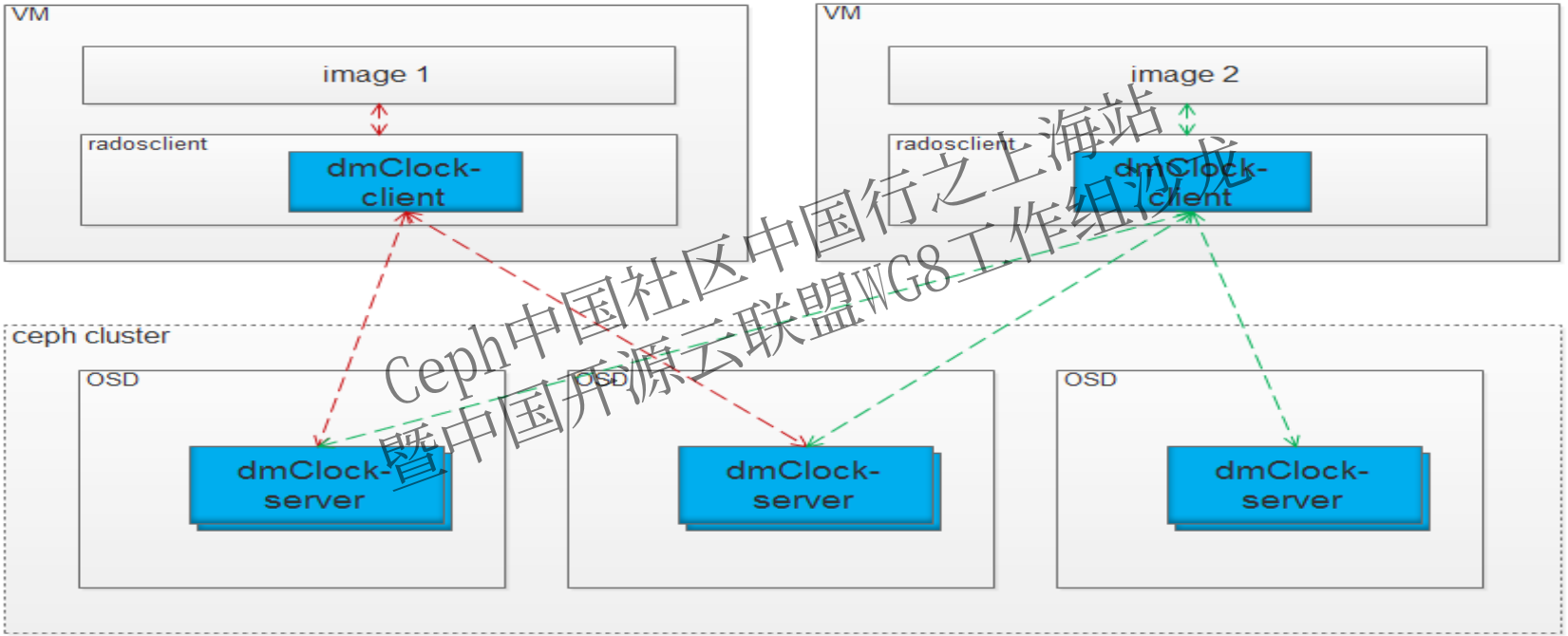
$$W_i^{r'} = W_i^r + \sigma_i$$

$$\sigma_i = \min\{W_i^{r-1}\} - time$$

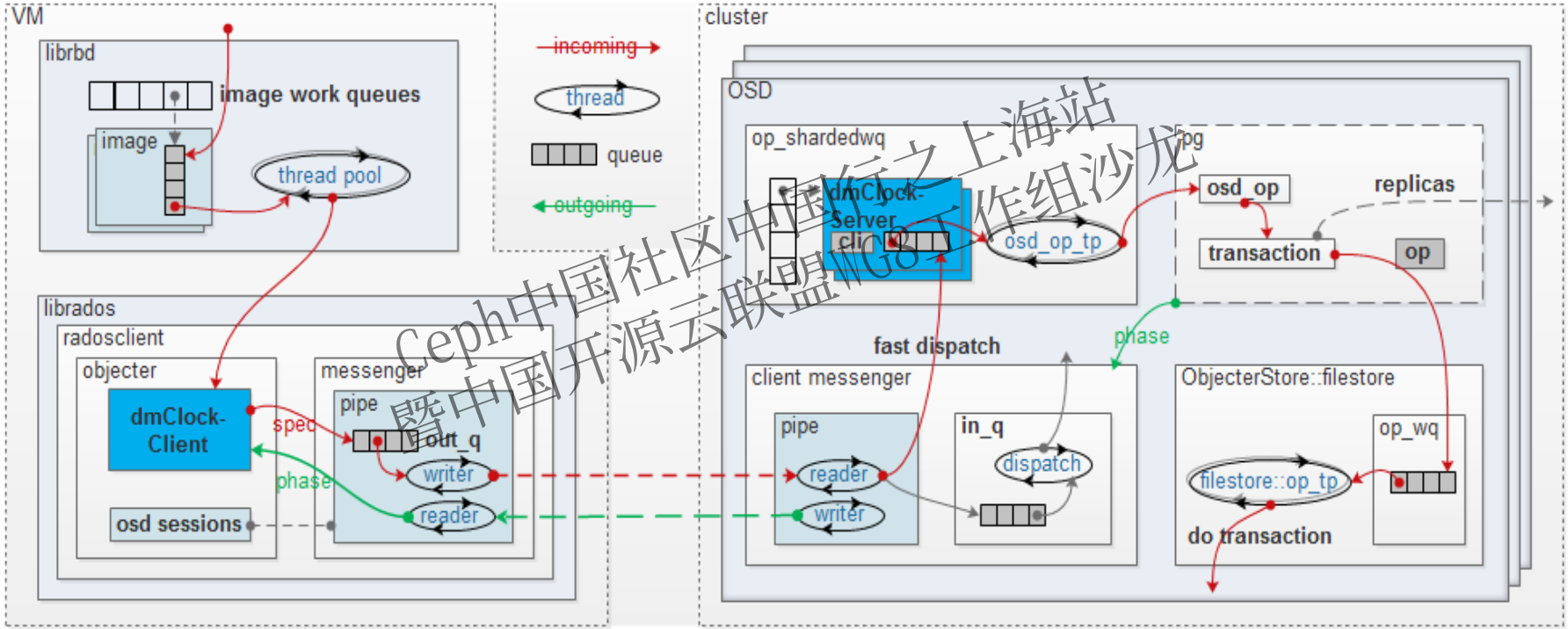
提要

- dmClock算法
- Ceph中的QoS设计实现
- 结果和分析
- 后续工作

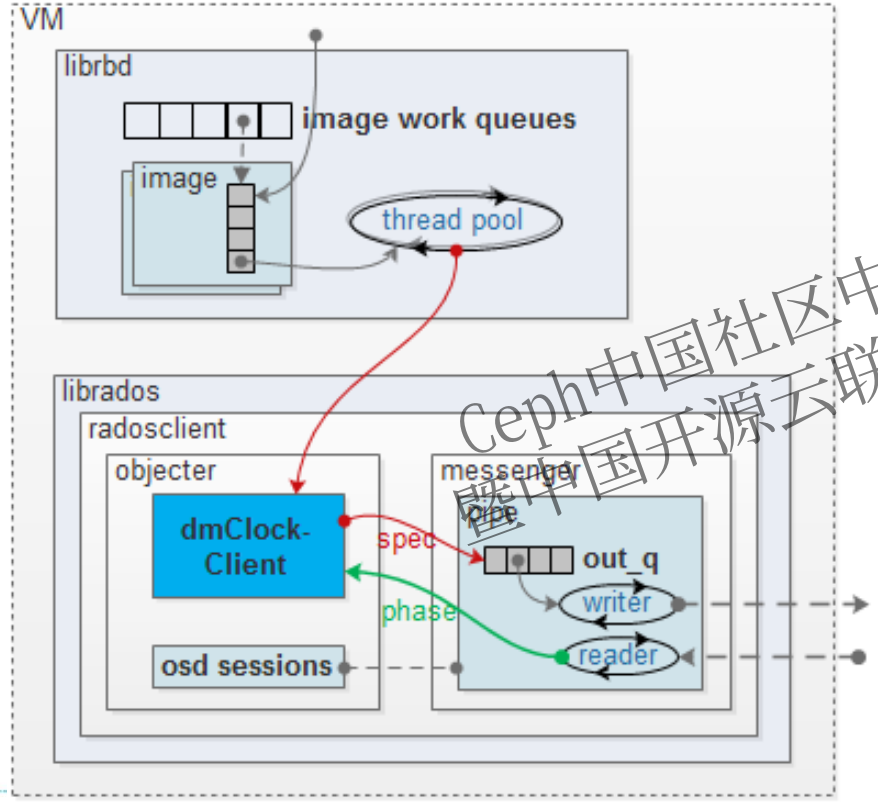
Ceph QoS 设计实现



Ceph QoS 设计实现



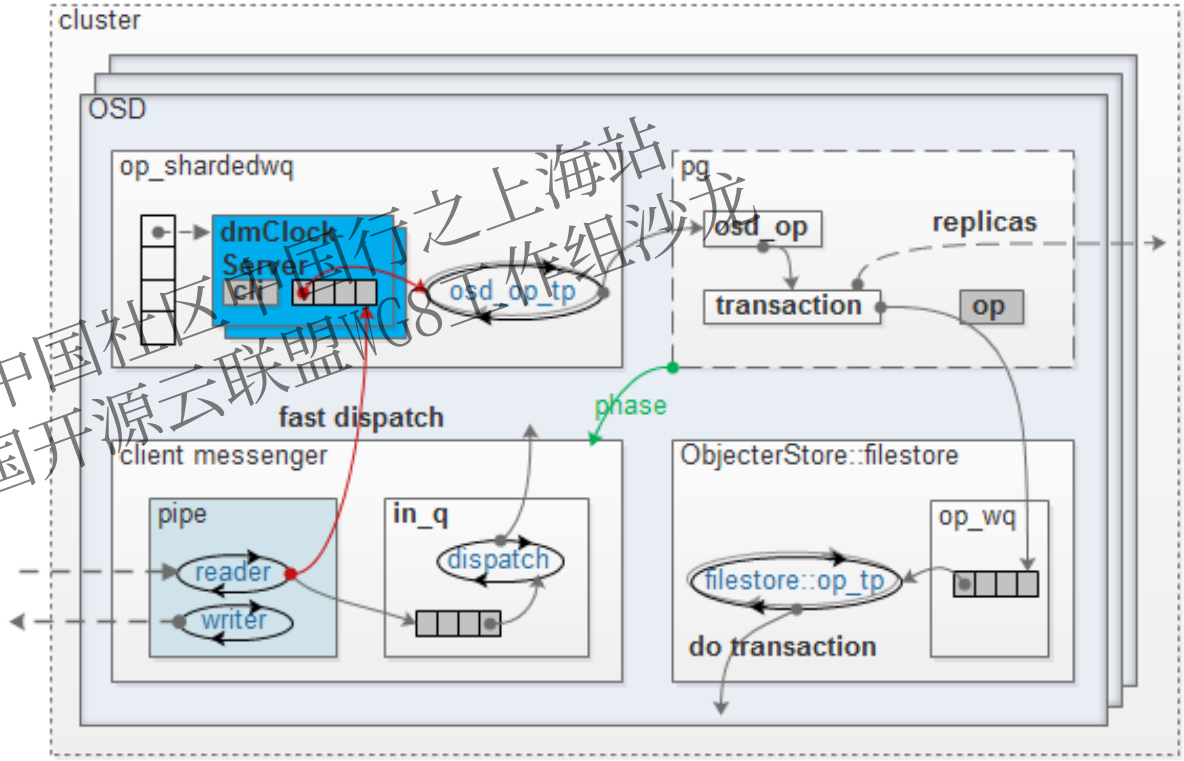
Ceph QoS 设计实现



- dmClock-client 驻留于 objecter
- 请求下发经过dmc-Client添加Qos值、pho/delta等信息，在请求回应reply 中把 phase 信息带回来；
- 提供 librados 层和 librbd 层的QoS模板 配置接口，可以动态修改QoS值；

Ceph QoS 设计实现

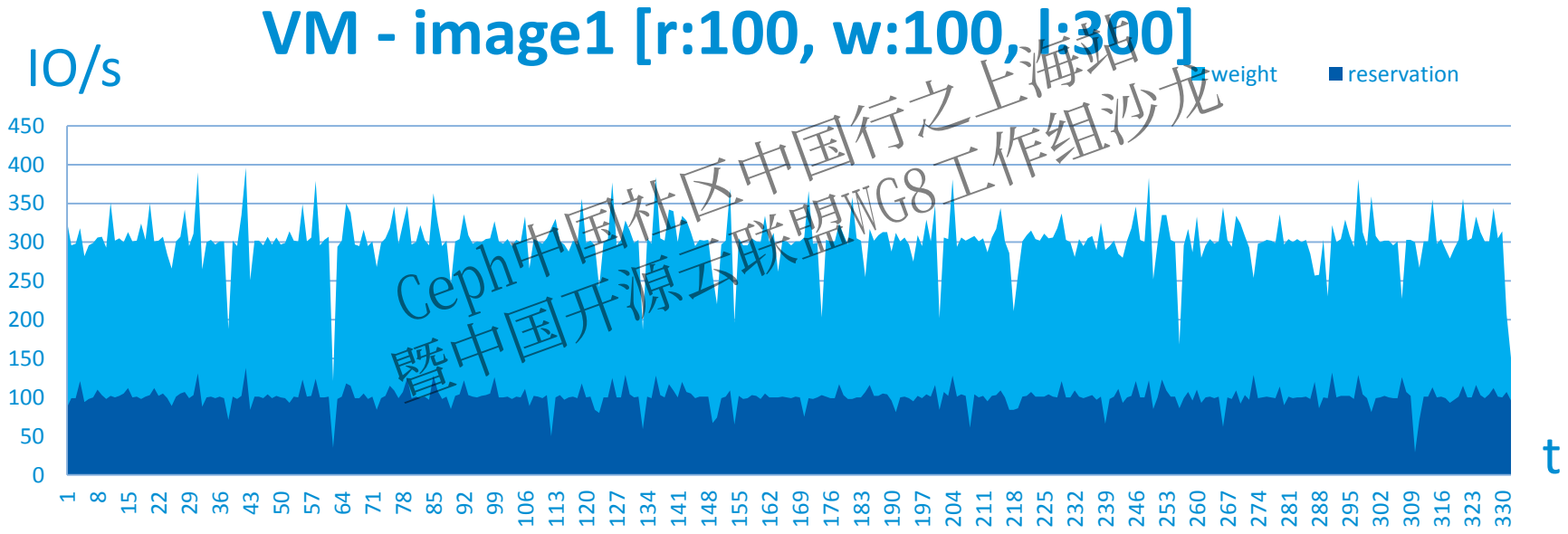
- dmClock-Server 基于时间标签的队列取代 prio/wpq 队列；
- 在reply中将 phase 信息带回来；



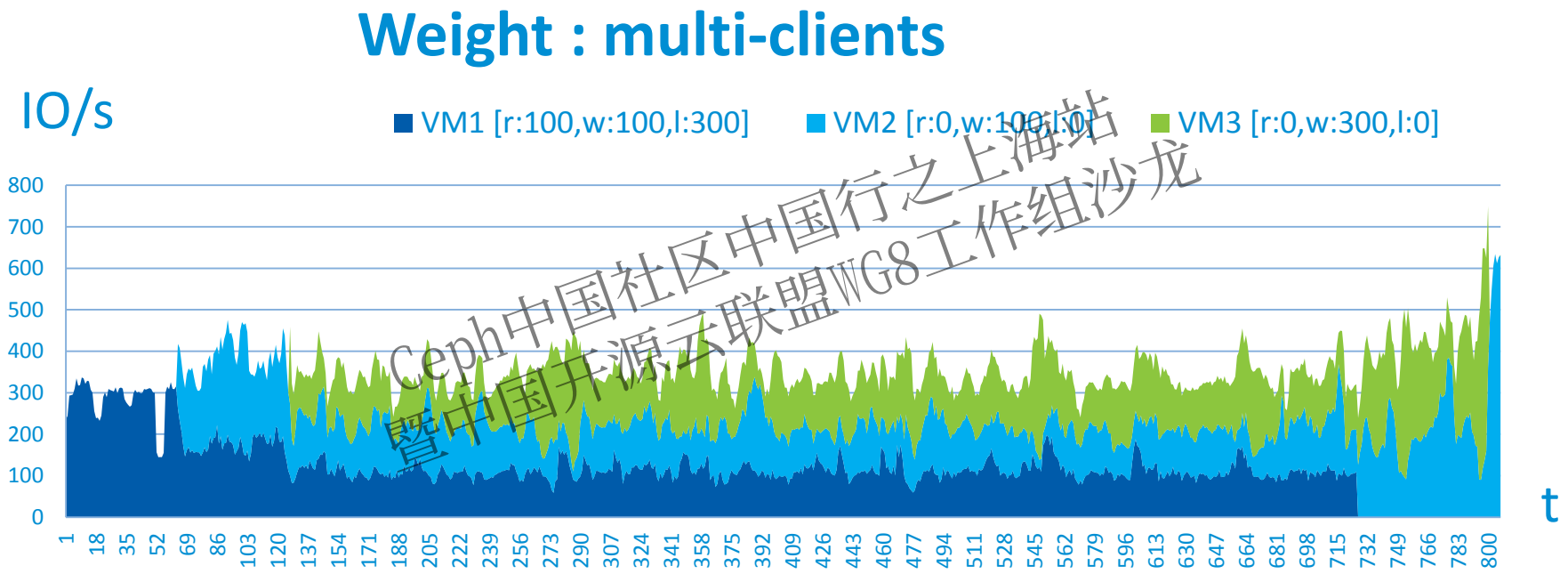
提要

- dmClock算法
- Ceph中的QoS设计实现
- 结果和分析
- 后续工作

结果与分析



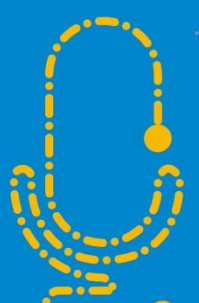
结果与分析



后续工作

- 多客户端的权重控制
- 集群节点间PG数据恢复 recovery IO 速率
- 系统QoS配置策略优化

谢谢！



Ceph中国社区中国行之上海站
暨中国开源云联盟WG8工作组沙龙

未来，不等待