

美团云

分布式块存储系统

设计与实现

李慧霸 博士

2015 10 17

上海

个人简介



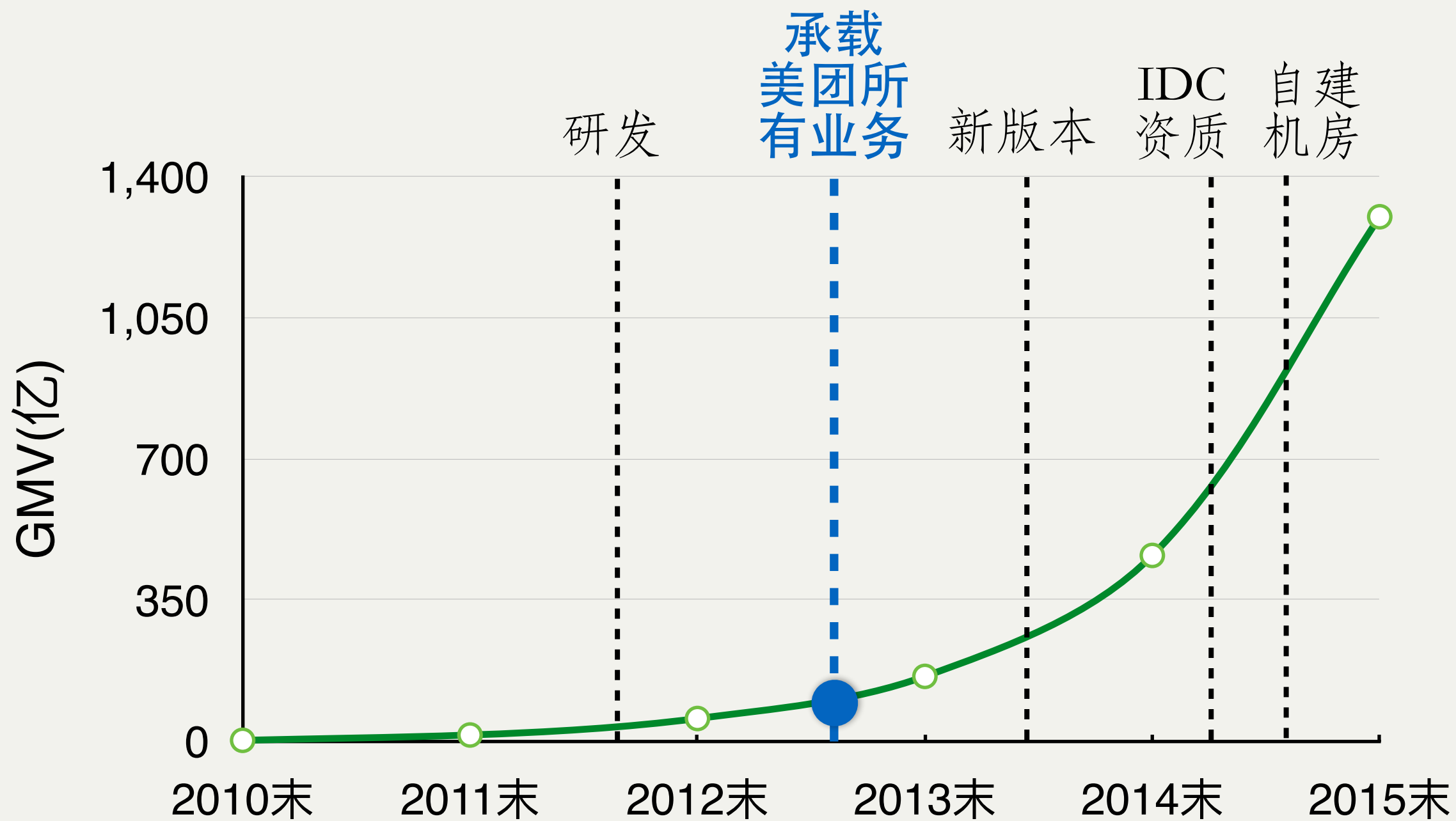
- 李慧霸
- 美团云
 - ❖ 分布式块存储项目负责人
- 国防科学技术大学
 - ❖ 并行与分布处理国家重点实验室
 - ❖ 研究员

内容提要



- 个人简介
- 云计算@美团
- 云计算对块存储的需求
- 设计与实现
 - ❖ 设计理念、存储结构、并发模型、
 - ❖ 系统架构、写入策略、错误防护
- 性能测试

美团云



稳定承载美团所有业务



运维

前端接入 请求转发 请求过滤 内容缓存 验证码 ...

业务逻辑 团购 电影 酒店 外卖 ...

业务组件 用户中心 支付 搜索 推荐 地理位置 风控 ...

基础组件 配置 队列服务 注册中心 NoSQL MySQL ...

云平台 云主机 EBS SDN ELB ...

安全

云计算承载研发环境



- 方案验证
- 新技术探索
- 压力测试
- 鲁棒性测试
- “小白鼠”平台
- 配额内自助使用

极高的**交付效率**

极高的**资源利用率**

月**资源“换手”率** > 50%

美团云 - 公有云服务

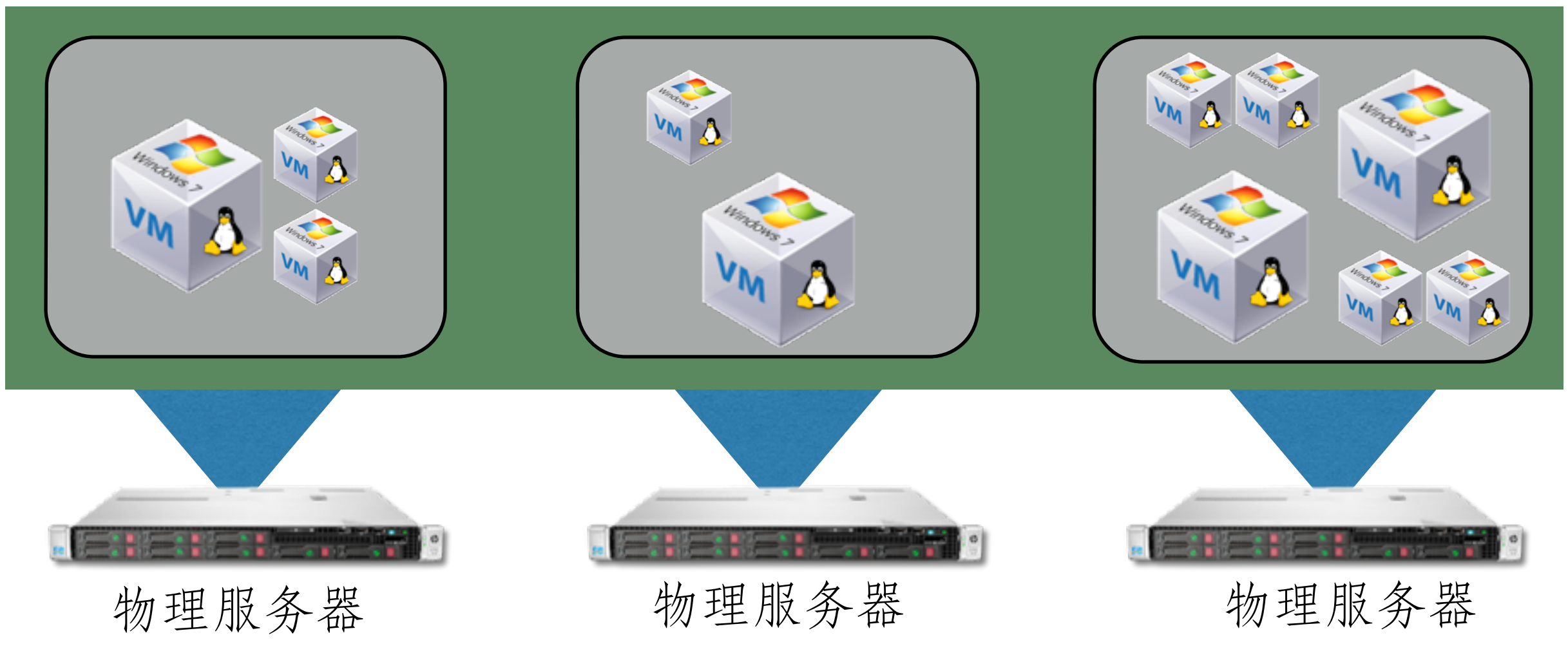


内容提要

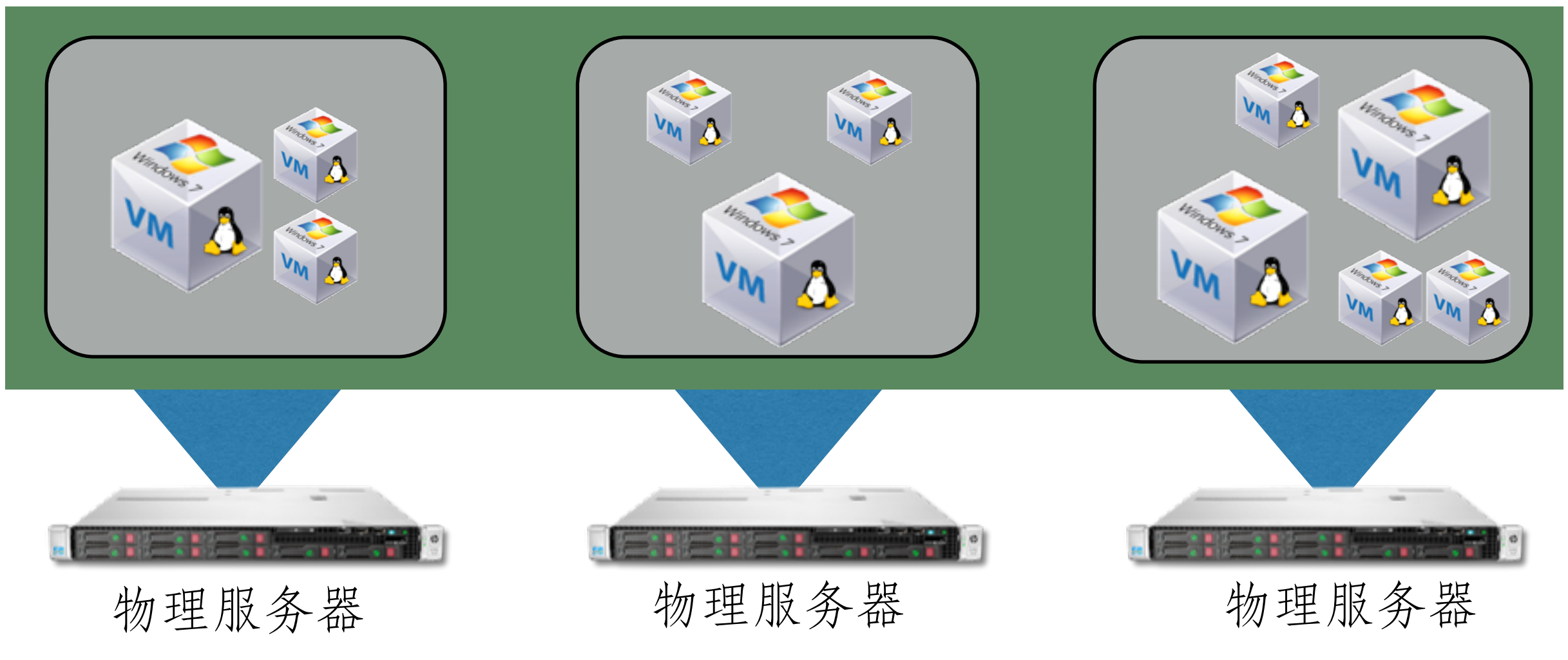


- 个人简介
- 云计算@美团
- 云计算对块存储的需求
- 设计与实现
 - ❖ 设计理念、存储结构、并发模型、
 - ❖ 系统架构、写入策略、错误防护
- 性能测试

IaaS 云计算



IaaS 云计算



三类存储架构



- 本地磁盘存储
 - ❖ 技术简单，性能高，成本低
 - ❖ VM迁移困难，可用性低
- 集中共享存储
 - ❖ 技术简单，VM易于迁移
 - ❖ 成本高，可用性低，扩展性低
- 分布共享存储
 - ❖ 可靠性高，可用性高，VM易于迁移
 - ❖ 技术困难

三类存储架构

- 本地磁盘存储
 - ❖ 技术简单，性能高，成本低
 - ❖ VM迁移困难，可用性低
- 集中共享存储
 - ❖ 技术简单，VM易于迁移
 - ❖ 成本高，可用性低，扩展性低
- 分布共享存储 ✓
 - ❖ 可靠性高，可用性高，VM易于迁移
 - ❖ 技术困难

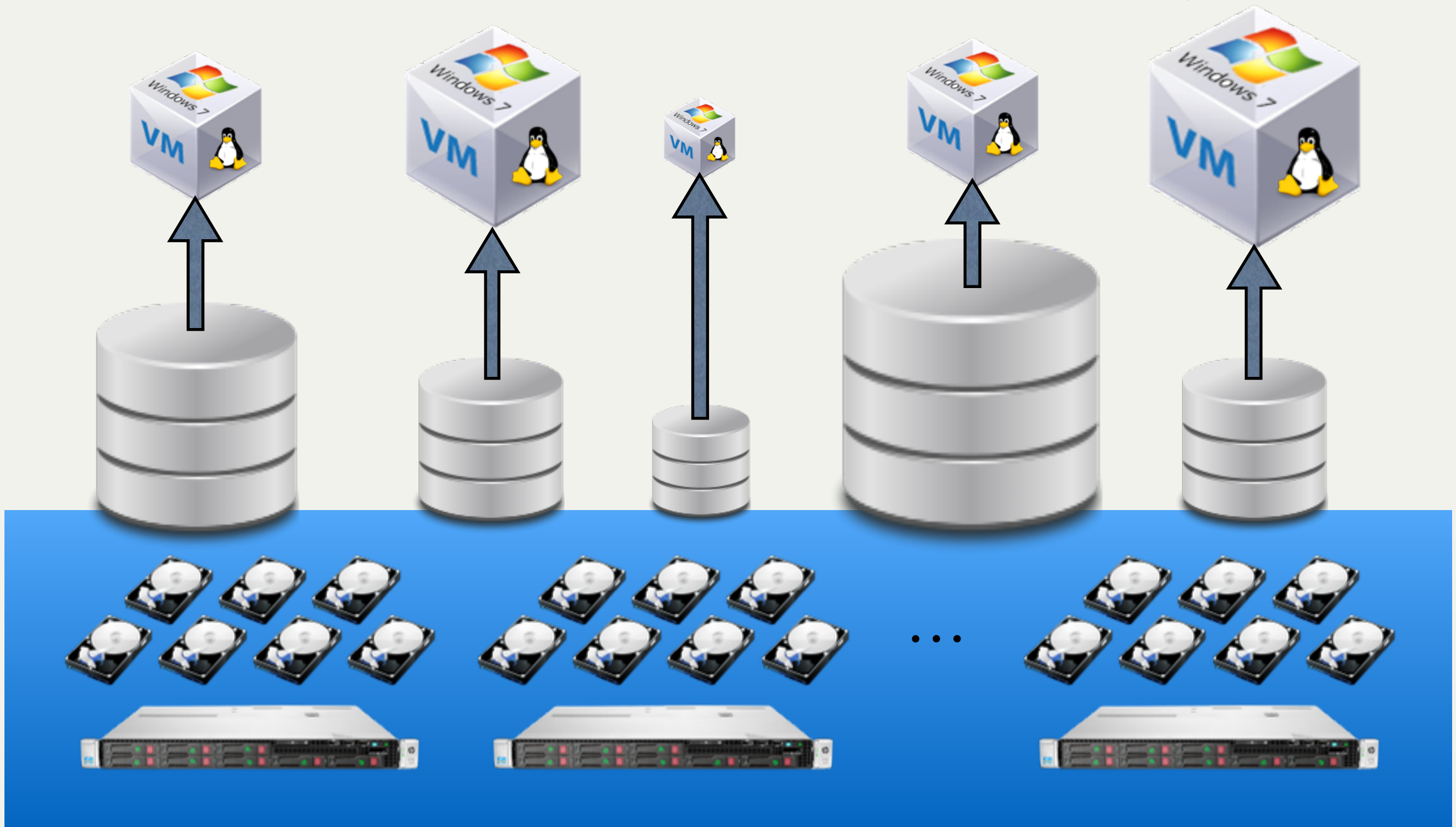
IaaS中的存储需求



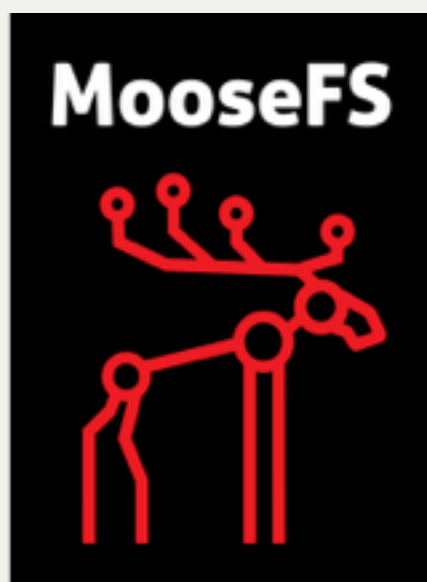
IaaS中的存储需求



IaaS中的存储需求



相关开源项目



Ceph的效率问题



服务端:

21282 root	20	0	809m	80m	11m	S	37.4	0.1	6:38.88	ceph-osd
17771 root	20	0	802m	80m	10m	S	27.8	0.1	4:54.46	ceph-osd
19448 root	20	0	839m	79m	10m	S	27.1	0.1	6:42.14	ceph-osd
18580 root	20	0	824m	82m	10m	S	26.5	0.1	5:52.29	ceph-osd
20392 root	20	0	798m	77m	10m	S	13.9	0.1	4:02.15	ceph-osd

123.7% @ 4,101读IOPS

客户端: 5xx% @16,407读IOPS

Sheepdog的数据丢失问题



```
[root@node3-10gtest ~]# collie cluster check
fix vdi test1-3
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fca
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fec
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e5630000026f5
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000002da6
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001e8c
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001530
...
fix vdi test2-9
50.9 % [=====>] 25 GB / 50 GB
no majority of d781e300000123
51.2 % [=====>] 26 GB / 50 GB
no majority of d781e30000018a
53.2 % [=====>] 27 GB / 50 GB
...
```

Sheepdog的数据丢失问题



```
[root@node3-10gtest ~]# collie cluster check
fix vdi test1-3
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fca
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fec
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e5630000026f5
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000002da6
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001e8c
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001530
...
fix vdi test2-9
50.9 % [=====>] 25 GB / 50 GB
no majority of d781e300000123
51.2 % [=====>] 26 GB / 50 GB
no majority of d781e30000018a
53.2 % [=====>] 27 GB / 50 GB
...
```

Sheepdog的数据丢失问题



```
[root@node3-10gtest ~]# collie cluster check
fix vdi test1-3
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fca
99.9 % [=====>] 50 GB / 50 GB
fixed replica 3e56300000fec
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e5630000026f5
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000002da6
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001e8c
100.0 % [=====>] 50 GB / 50 GB
fixed replica 3e563000001530
...
fix vdi test2-9
50.9 % [=====>] 25 GB / 50 GB
no majority of d781e300000123
51.2 % [=====>] 26 GB / 50 GB
no majority of d781e30000018a
53.2 % [=====>] 27 GB / 50 GB
...
```

内容提要



- 个人简介
- 云计算@美团
- 云计算对块存储的需求
- 设计与实现
 - ❖ 设计理念、存储结构、并发模型、
 - ❖ 系统架构、写入策略、错误防护
- 性能测试

设计理念



- 稳定可靠
 - ★ 在做trade off时，稳定压倒一起
- 高性能
 - ★ 能发挥万兆网和SSD的潜力
- 勤俭
 - ★ 低人力物力投入
 - * 能使用单路低端CPU
 - * 尽可能重用已有服务

设计理念



- CAP取舍
 - ❖ 舍C，取AP
- 服务无状态
 - ❖ stateless, timeless or functional
- 简化并发模型
- 模块化、层次化

系统架构



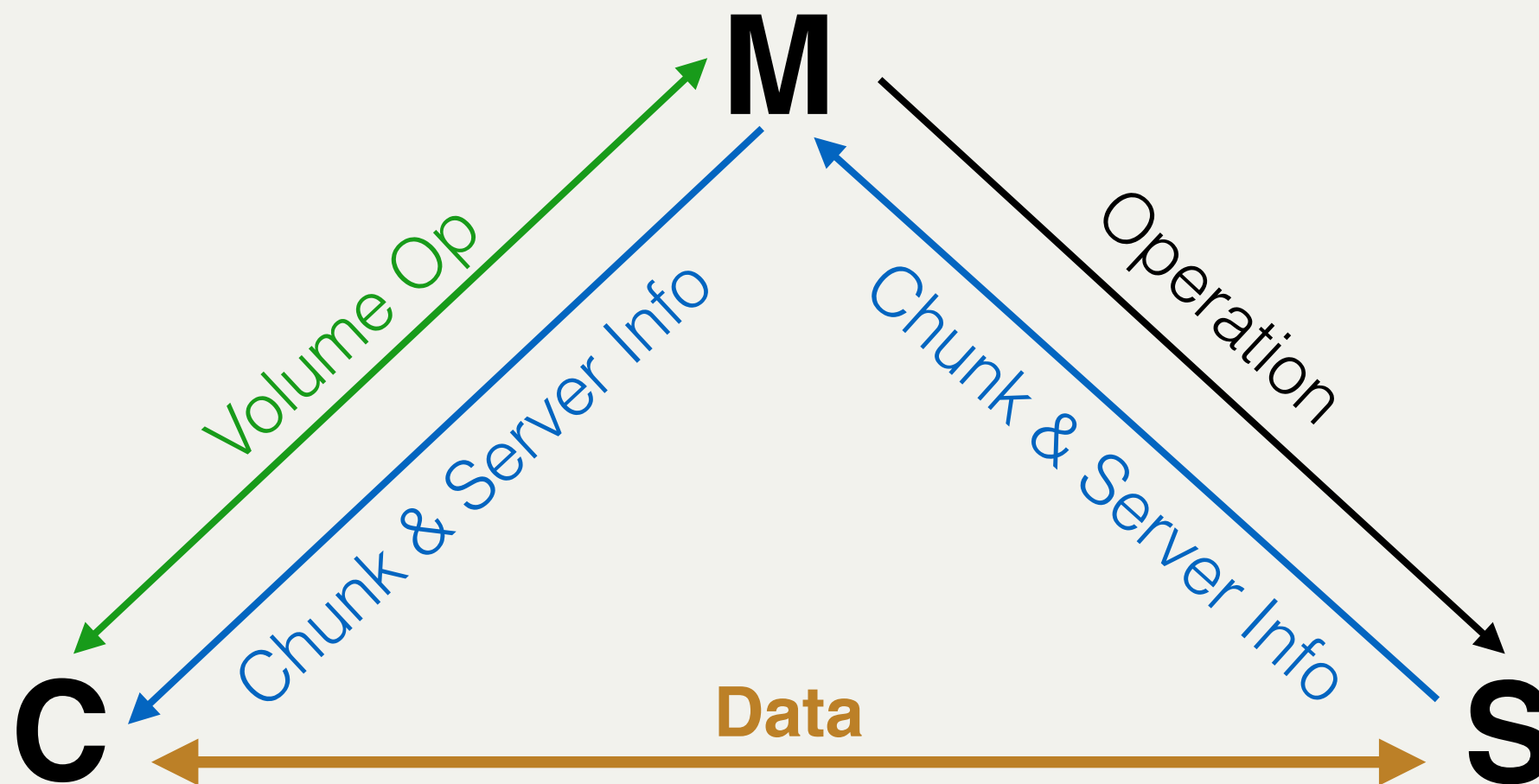
- 有中心（元数据服务器）
 - ❖ HDFS/GFS, MooseFS, ...
- 无中心（元数据服务器）
 - ❖ ceph, gluster, sheepdog, ...

系统架构

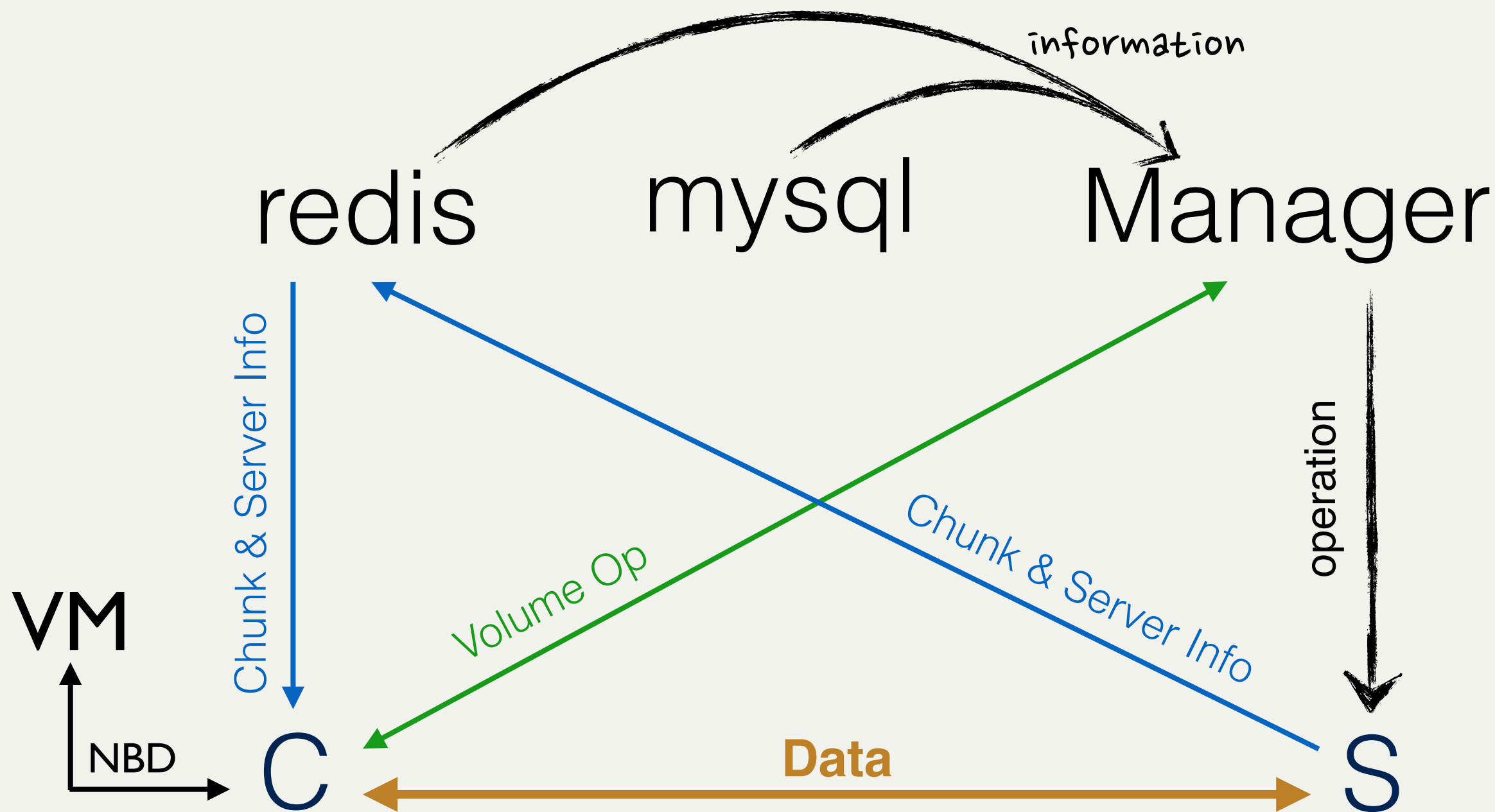


- 有中心（元数据服务器）
 - ❖ HDFS/GFS, MooseFS, ... ✓
- 无中心（元数据服务器）
 - ❖ ceph, gluster, sheepdog, ...

系统架构



系统架构



并发模型



- 事件驱动
 - ❖ `select()`, `poll()`, `epoll()`, ...
- 多线程
- 多进程
 - ❖ 每块硬盘对应一个进程
- 多协程

并发模型



- 事件驱动 ✓
 - ❖ `select()`, `poll()`, `epoll()`, ...
- 多线程
- 多进程
 - ❖ 每块硬盘对应一个进程
- 多协程

并发模型



- 事件驱动 ✓
 - ❖ `select()`, `poll()`, `epoll()`, ...
- 多线程
- 多进程 ✓
 - ❖ 每块硬盘对应一个进程
- 多协程

并发模型



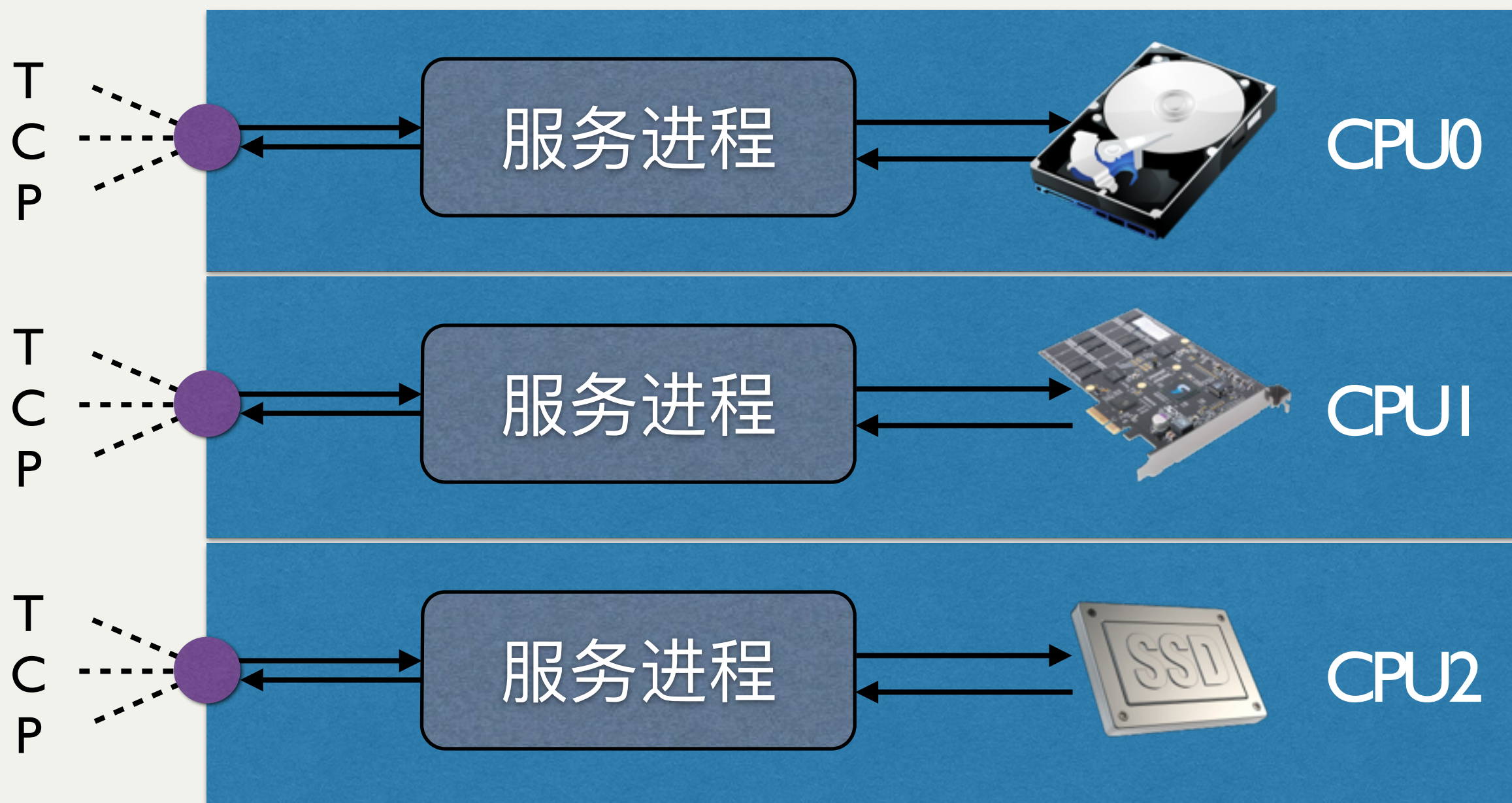
- 事件驱动 ✓
 - ❖ `select()`, `poll()`, `epoll()`, ...
- 多线程
- 多进程 ✓
 - ❖ 每块硬盘对应一个进程
- 多协程 ✓

并发模型

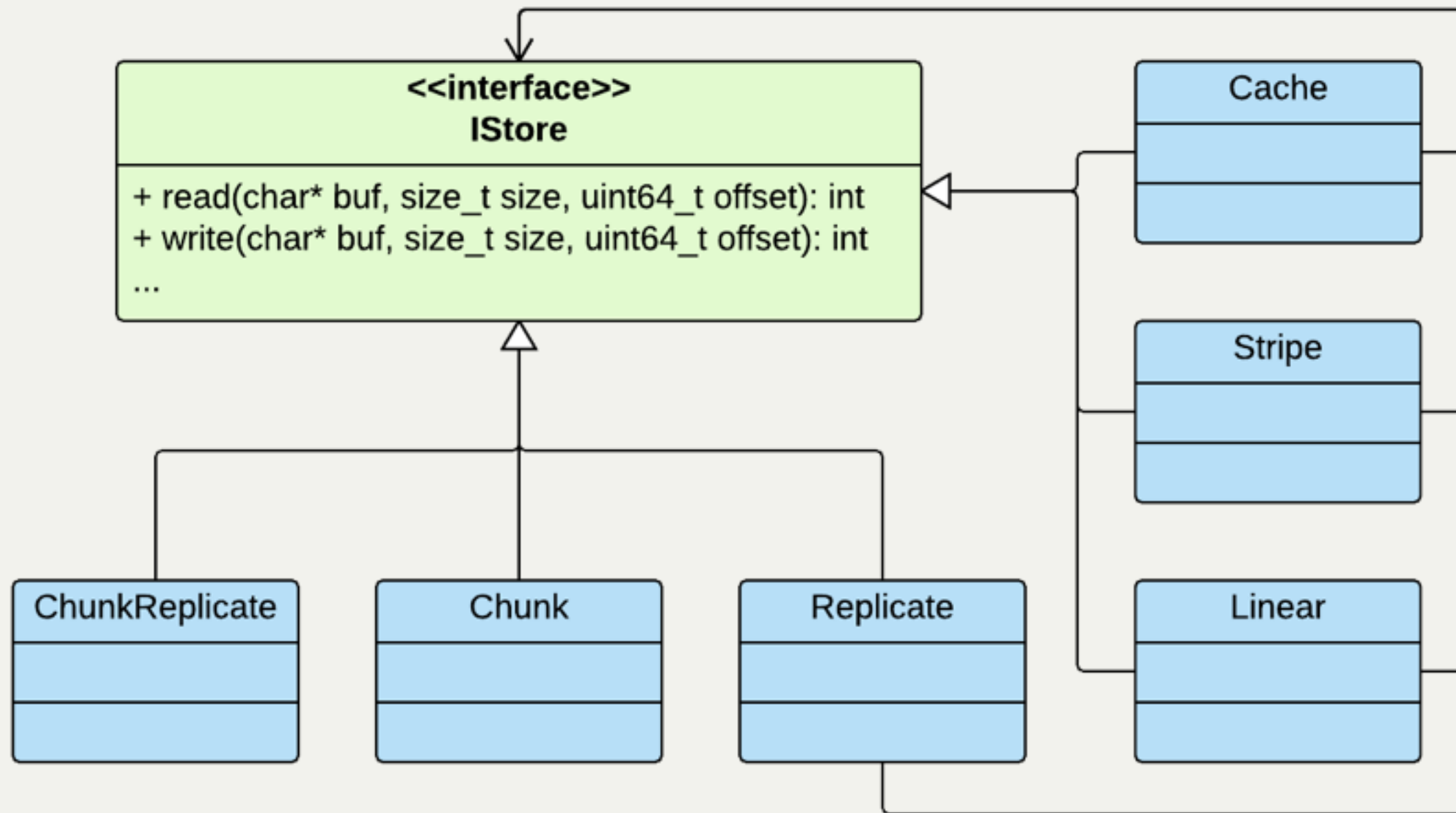
- 事件驱动 ✓
 - ❖ `select()`, `poll()`, `epoll()`, ...
- 多线程
- 多进程 ✓
 - ❖ 每块硬盘对应一个进程
- 多协程 ✓

同步化

CPU核心绑定

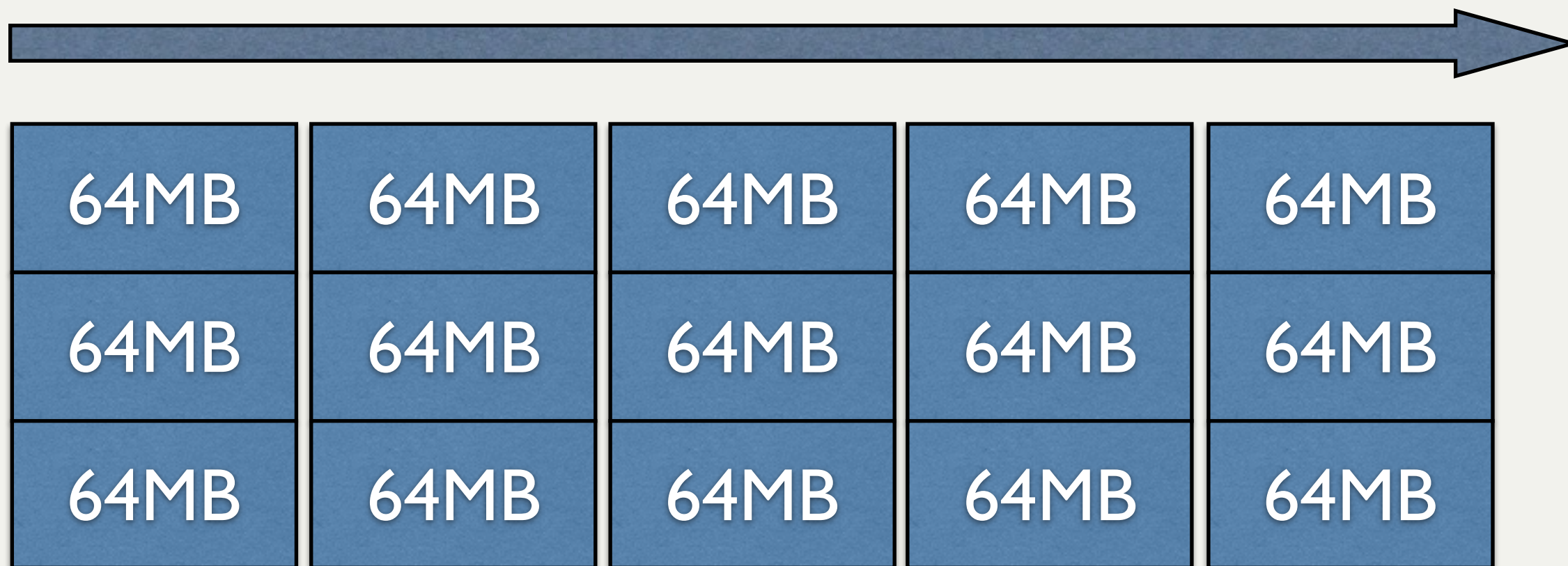


可组合的模块结构



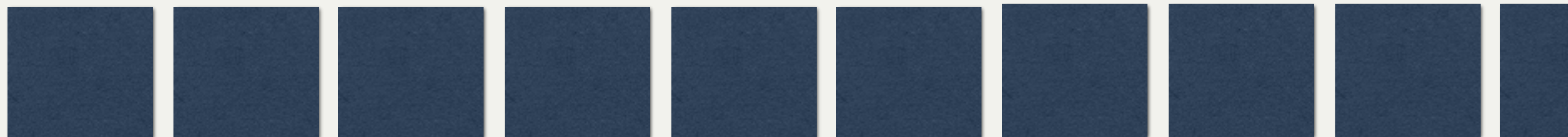
存储结构

卷/文件存储空间



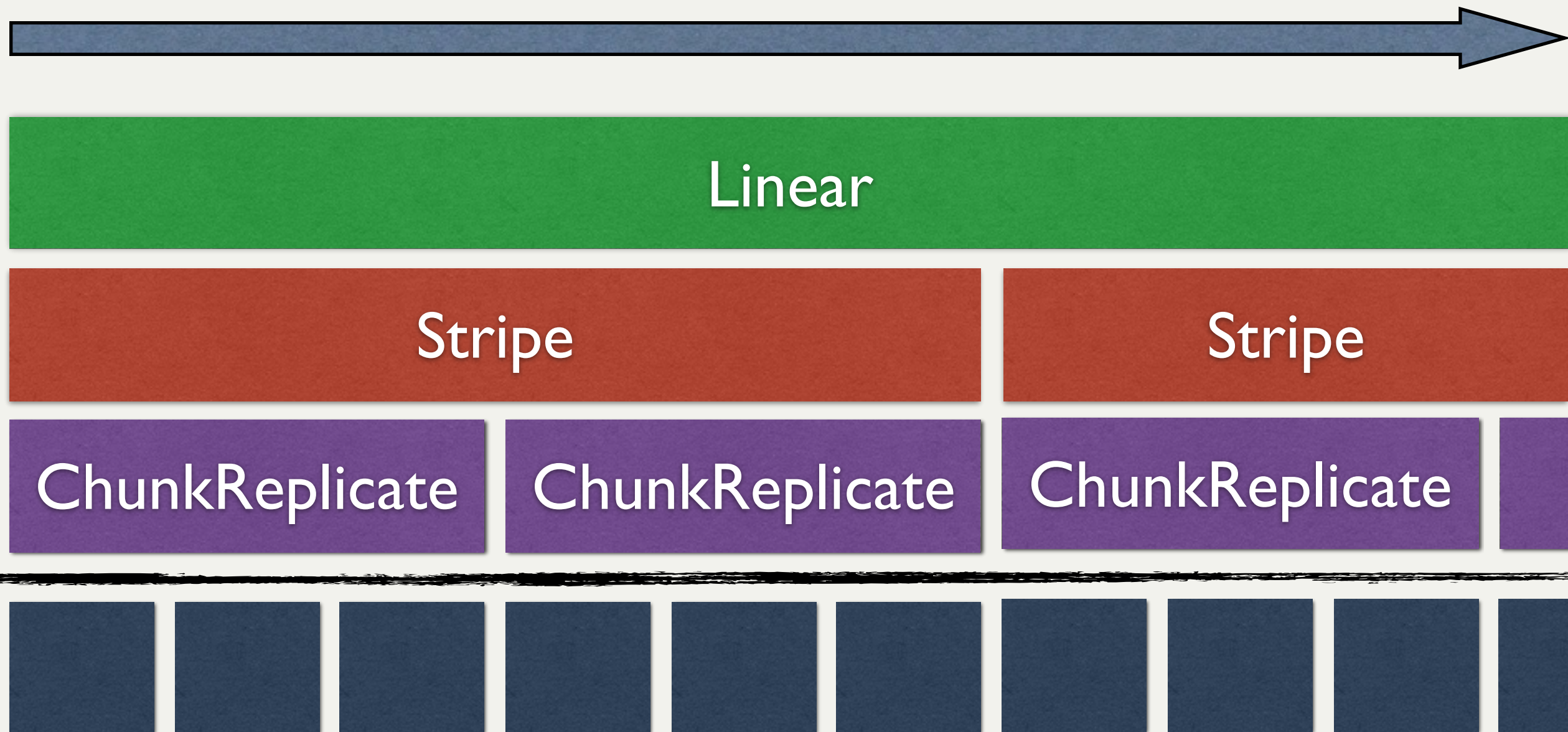
存储结构

卷/文件存储空间

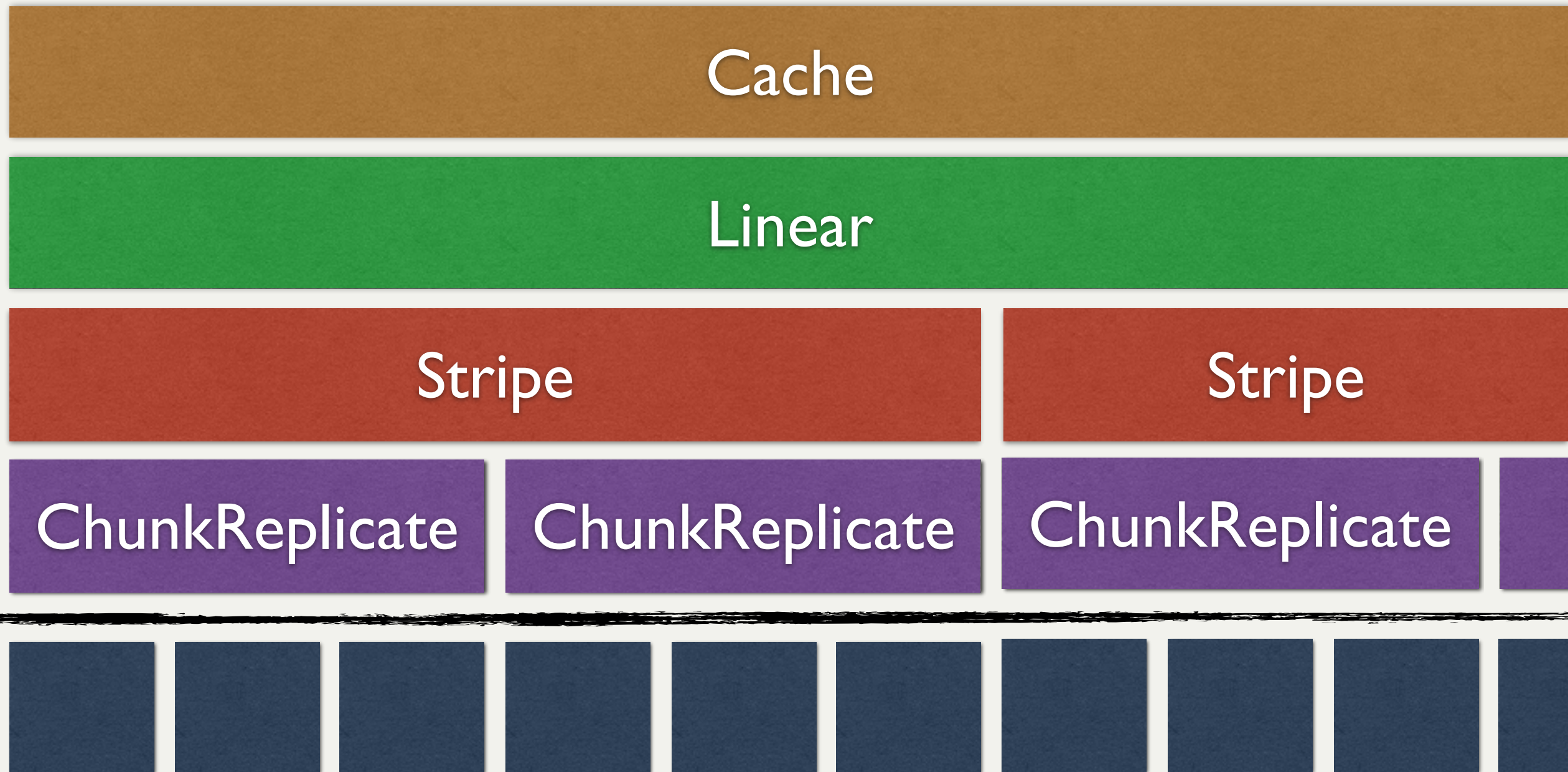


存储结构

卷/文件存储空间

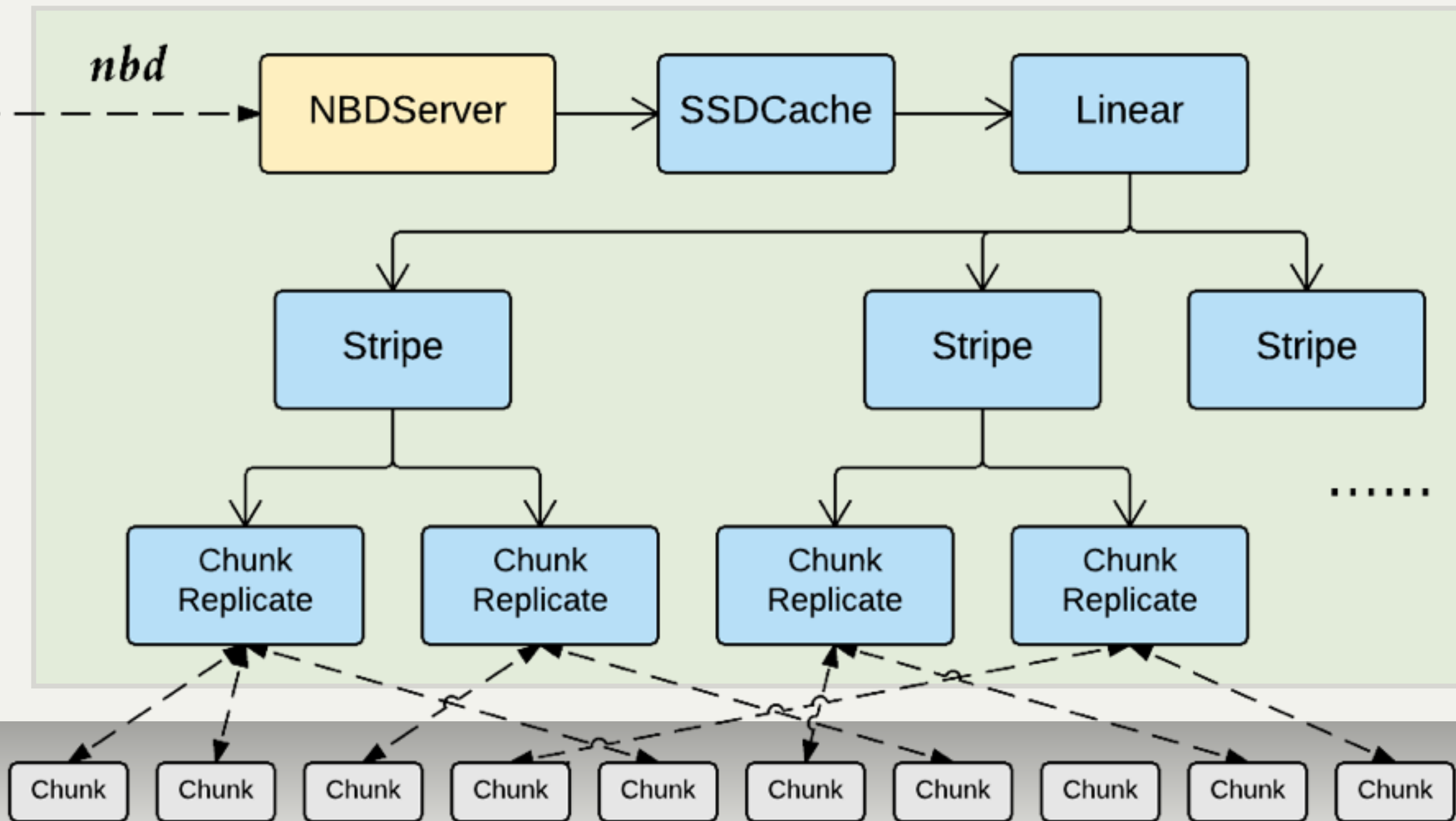


存储结构

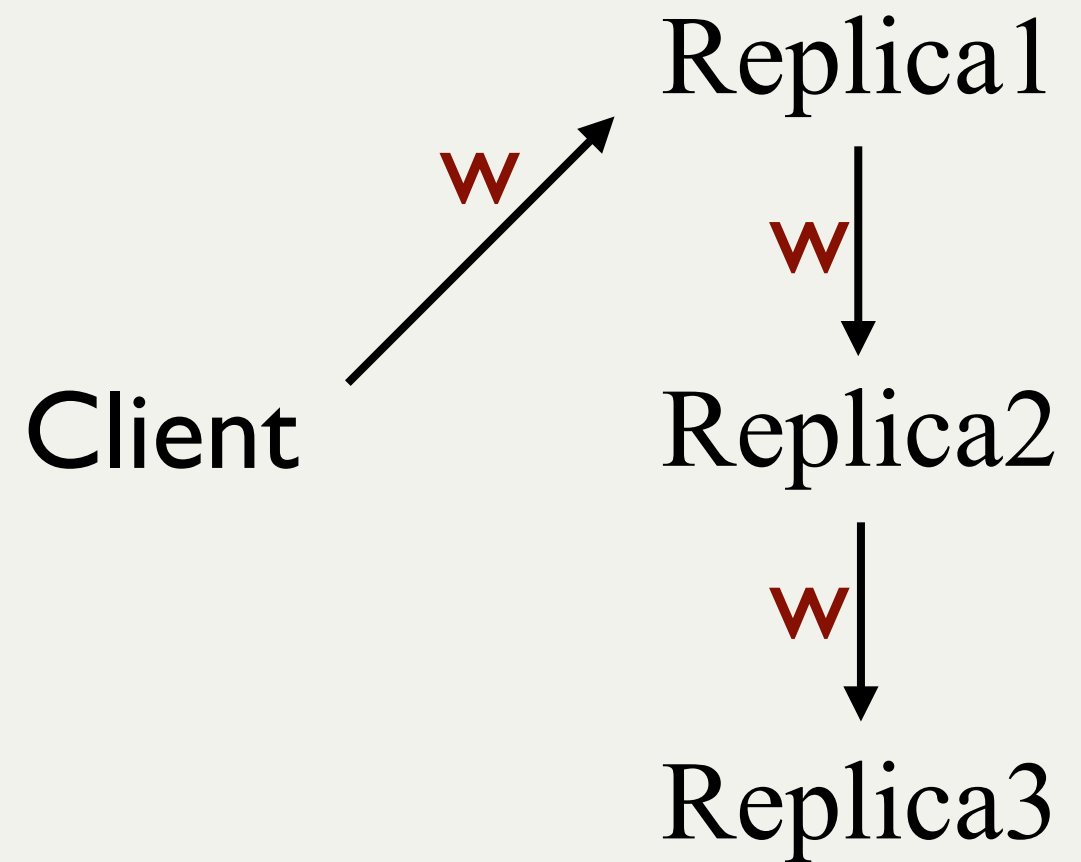
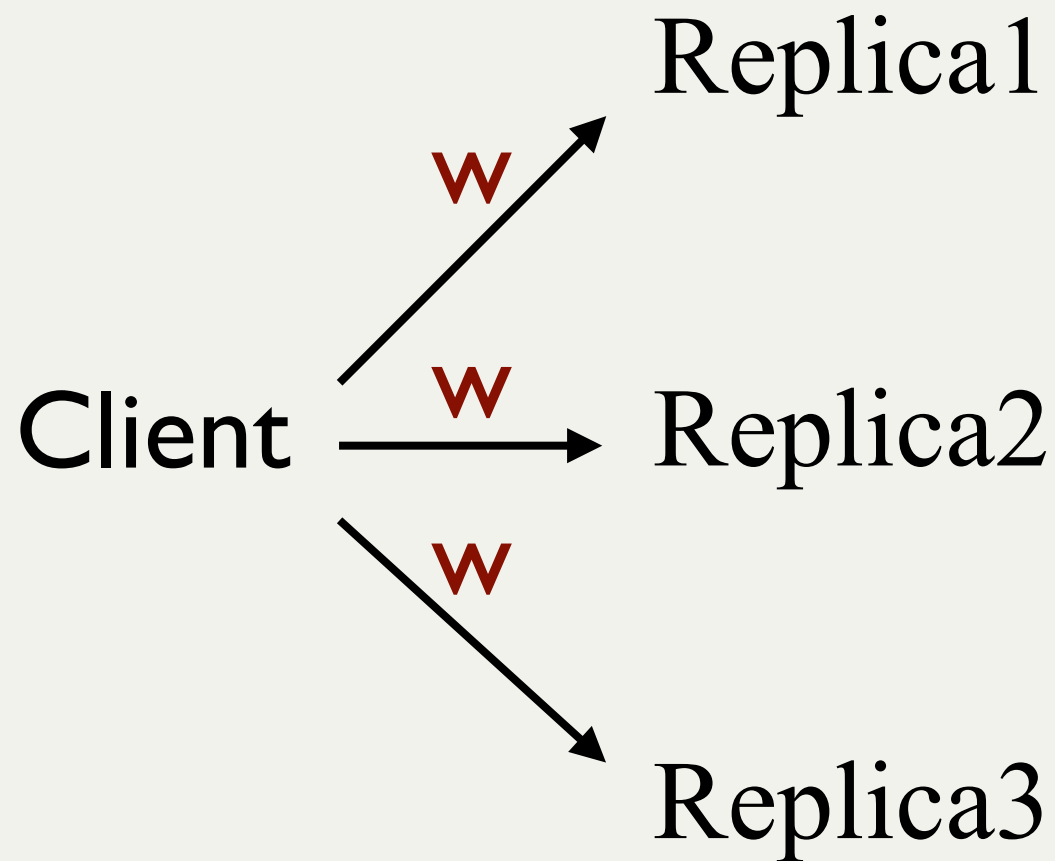


运行时模块结构

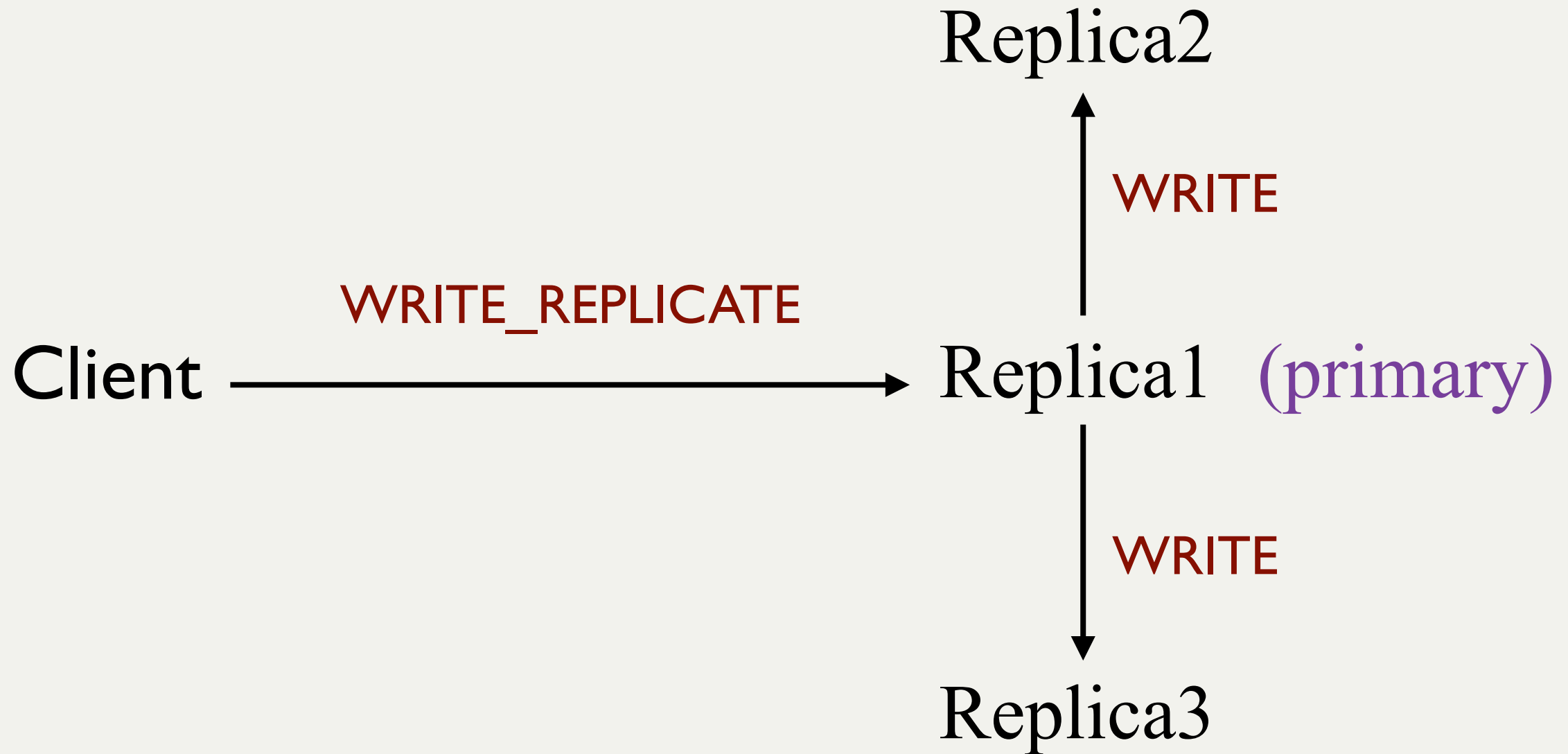
虚拟机



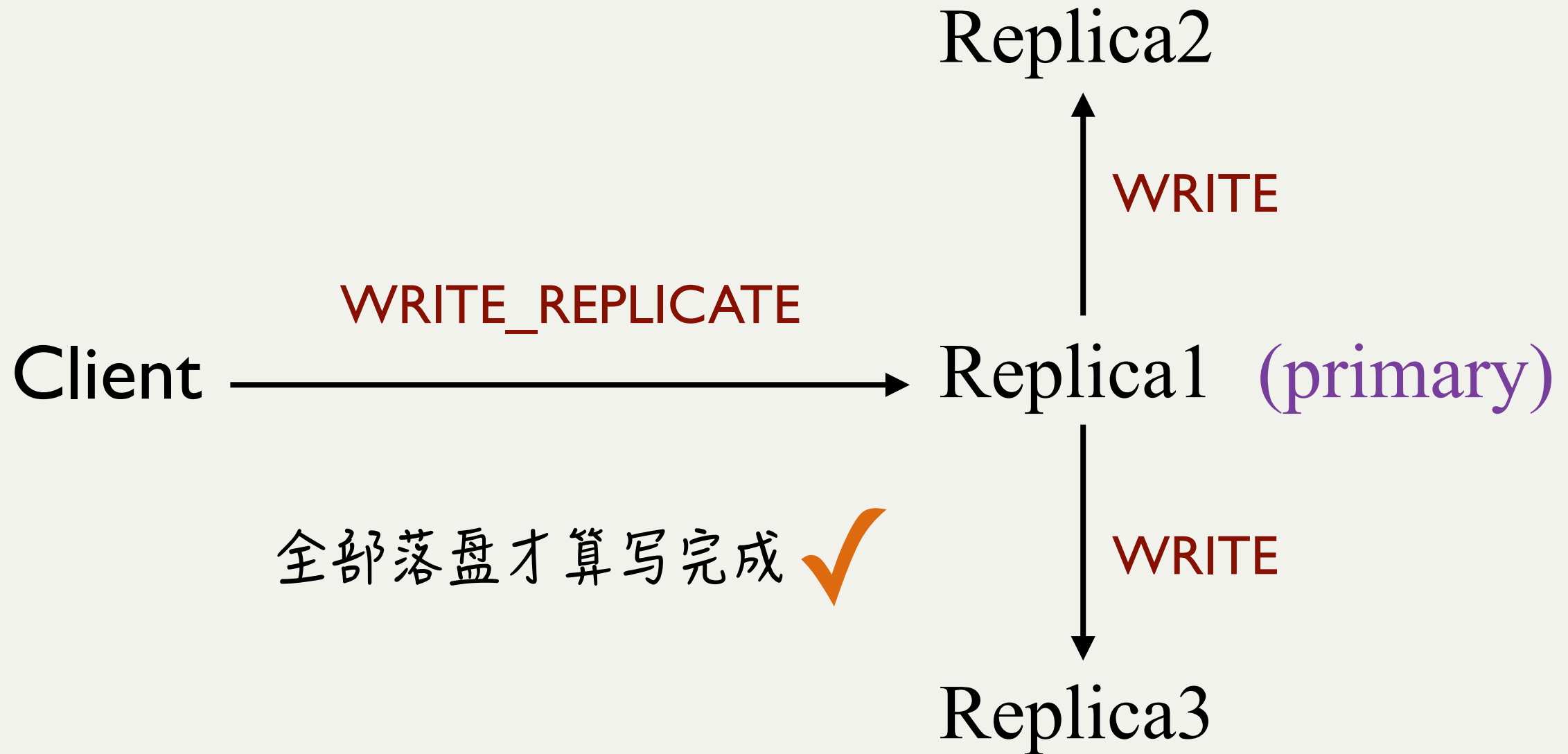
写入策略



写入策略



写入策略



一些“黑科技”



一些“黑科技”

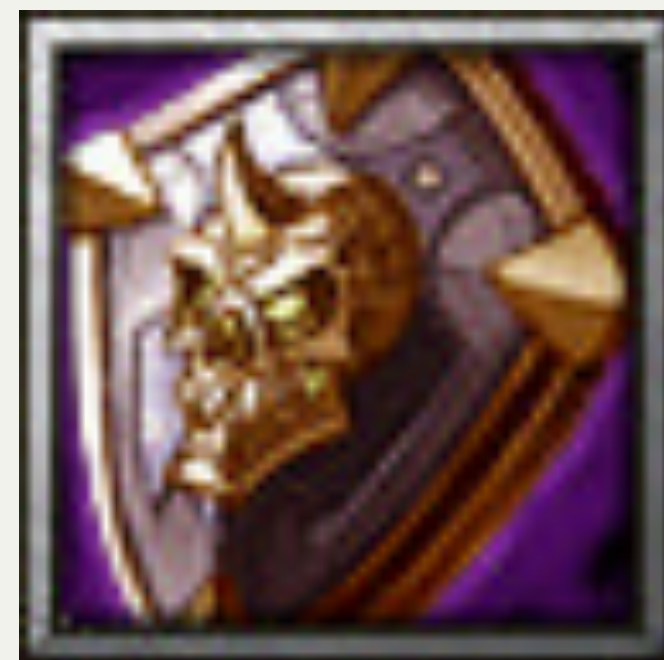


Nether Swap
移形换位

一些“黑科技”



Nether Swap
移形换位



Vanguard
先锋盾

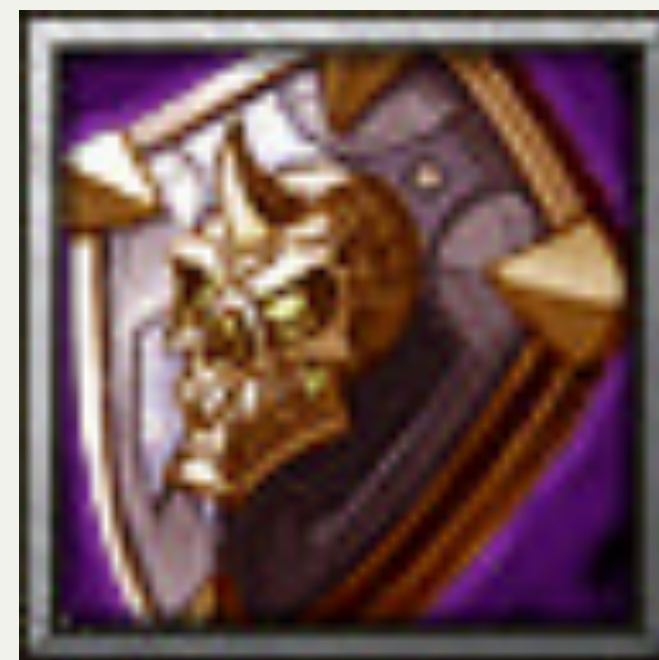
一些“黑科技”



Nether Swap
移形换位

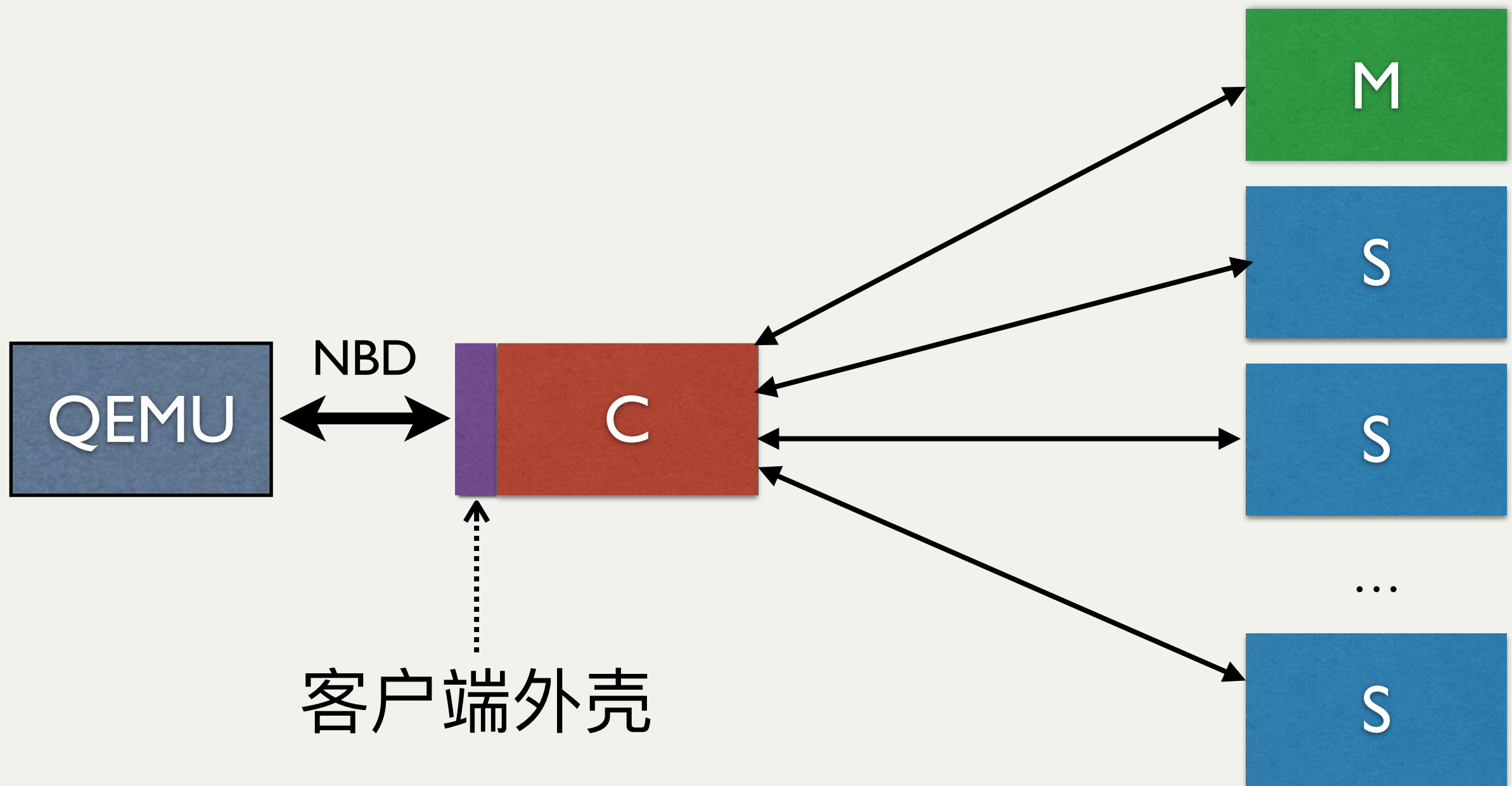


Aegis of the Immortal
不朽之盾

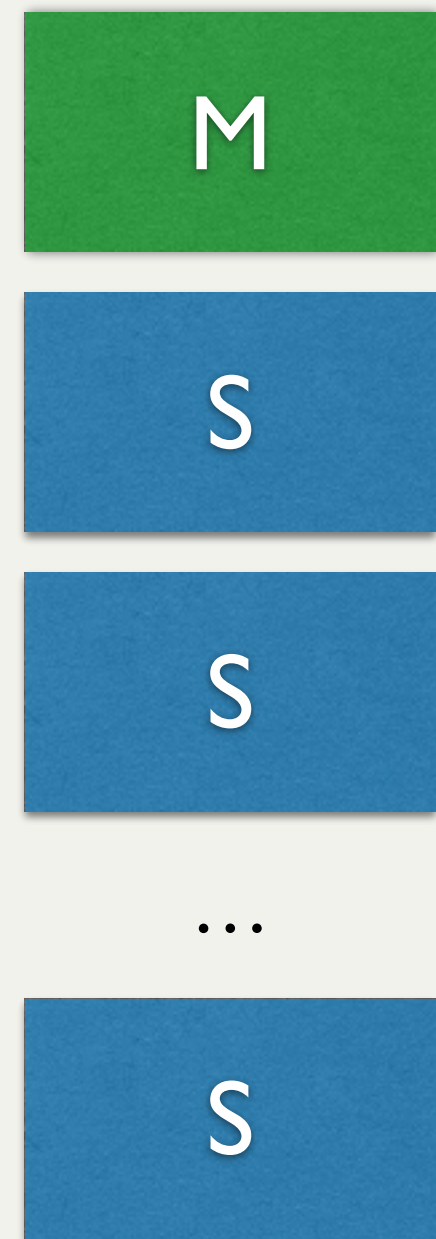
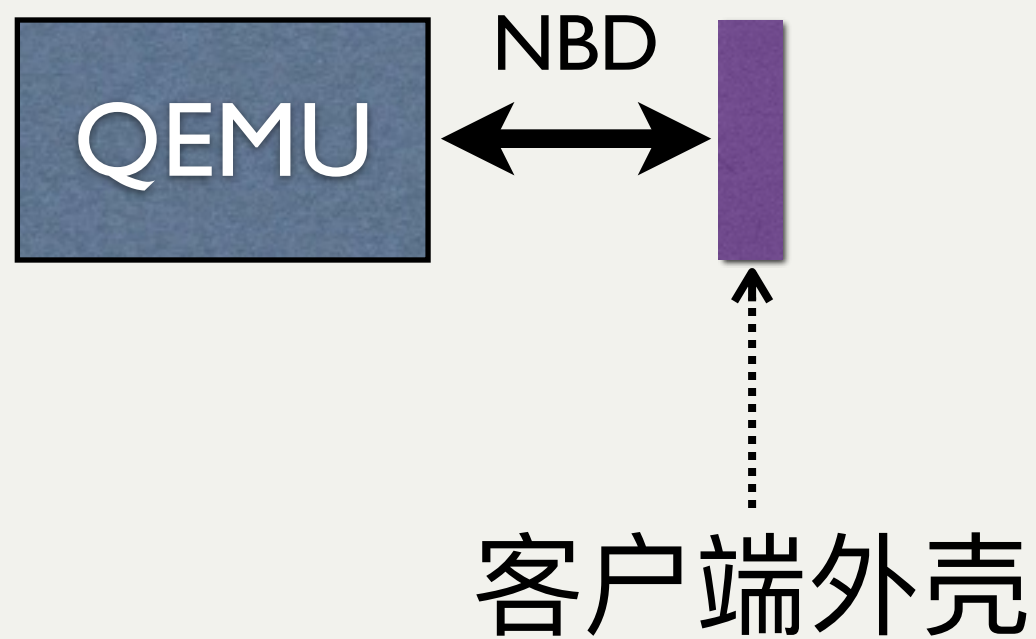


Vanguard
先锋盾

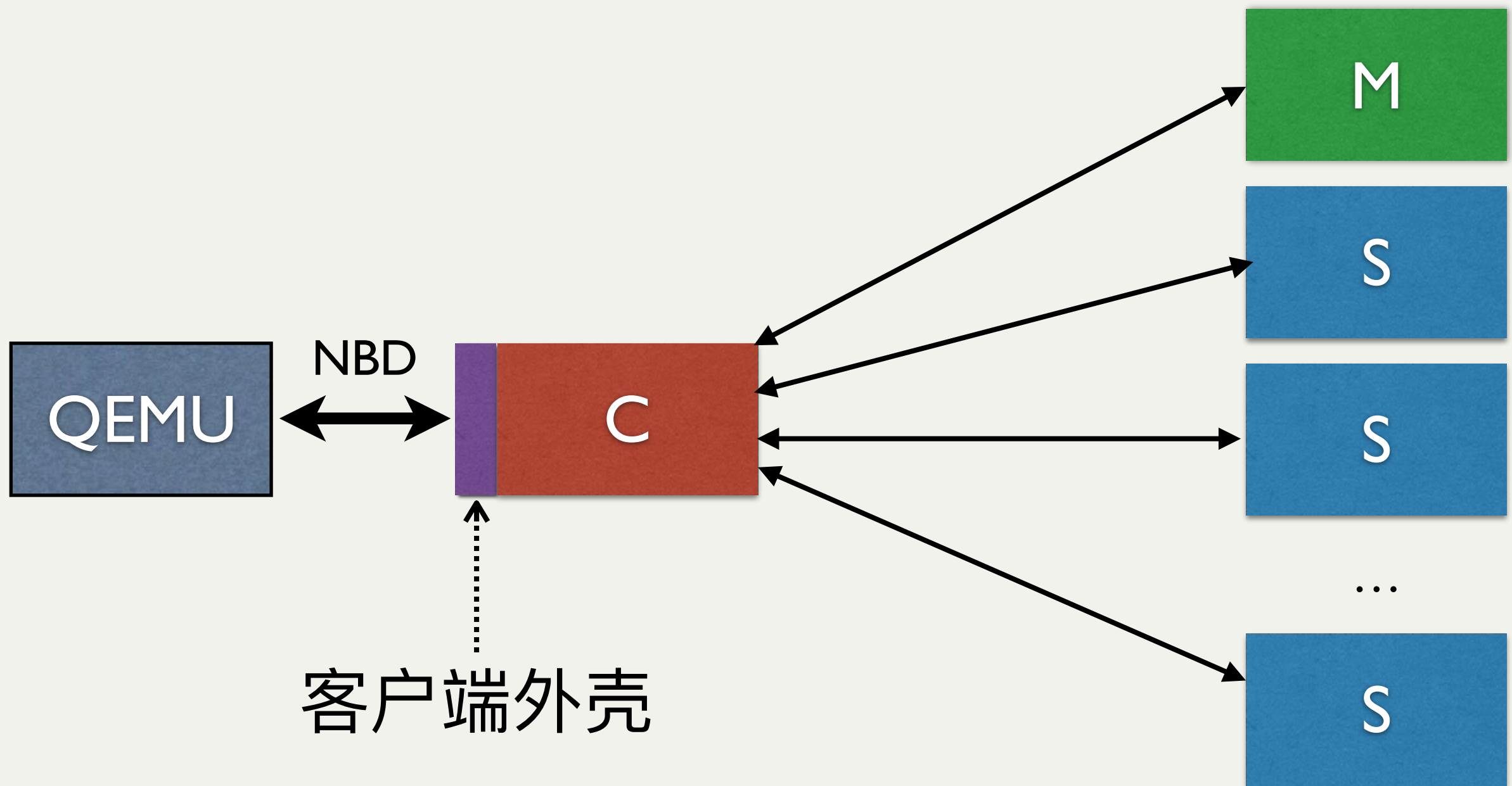
移形换位——热升级



移形换位——热升级



移形换位——热升级



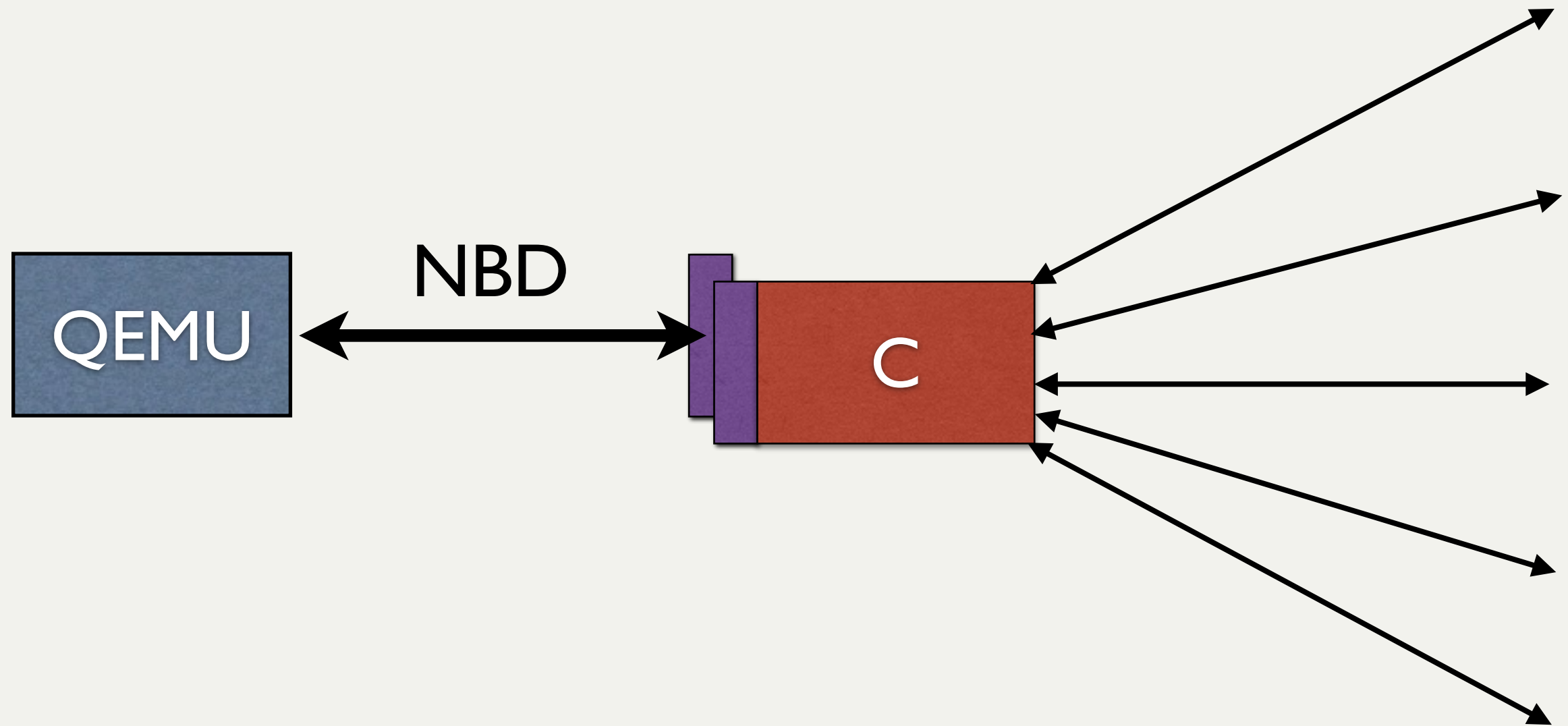
先锋盾——抵挡普通伤害



- 捕获一切异常信号
 - ❖ 除以0
 - ❖ 内存非法访问
 - ❖ ...
- 返回到正确场景（壳）
 - ❖ `setjmp() / longjmp()`
 - ❖ `try { throw } catch(...) {}`

不朽之盾——起死回生

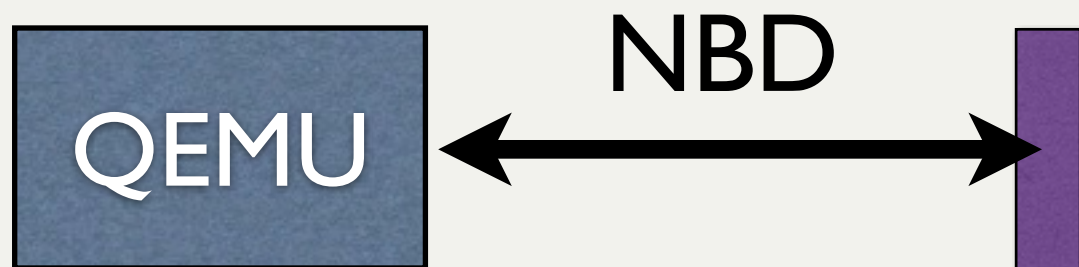
- 任何时候，恢复到绝对正确的运行状态!!!



不朽之盾——起死回生

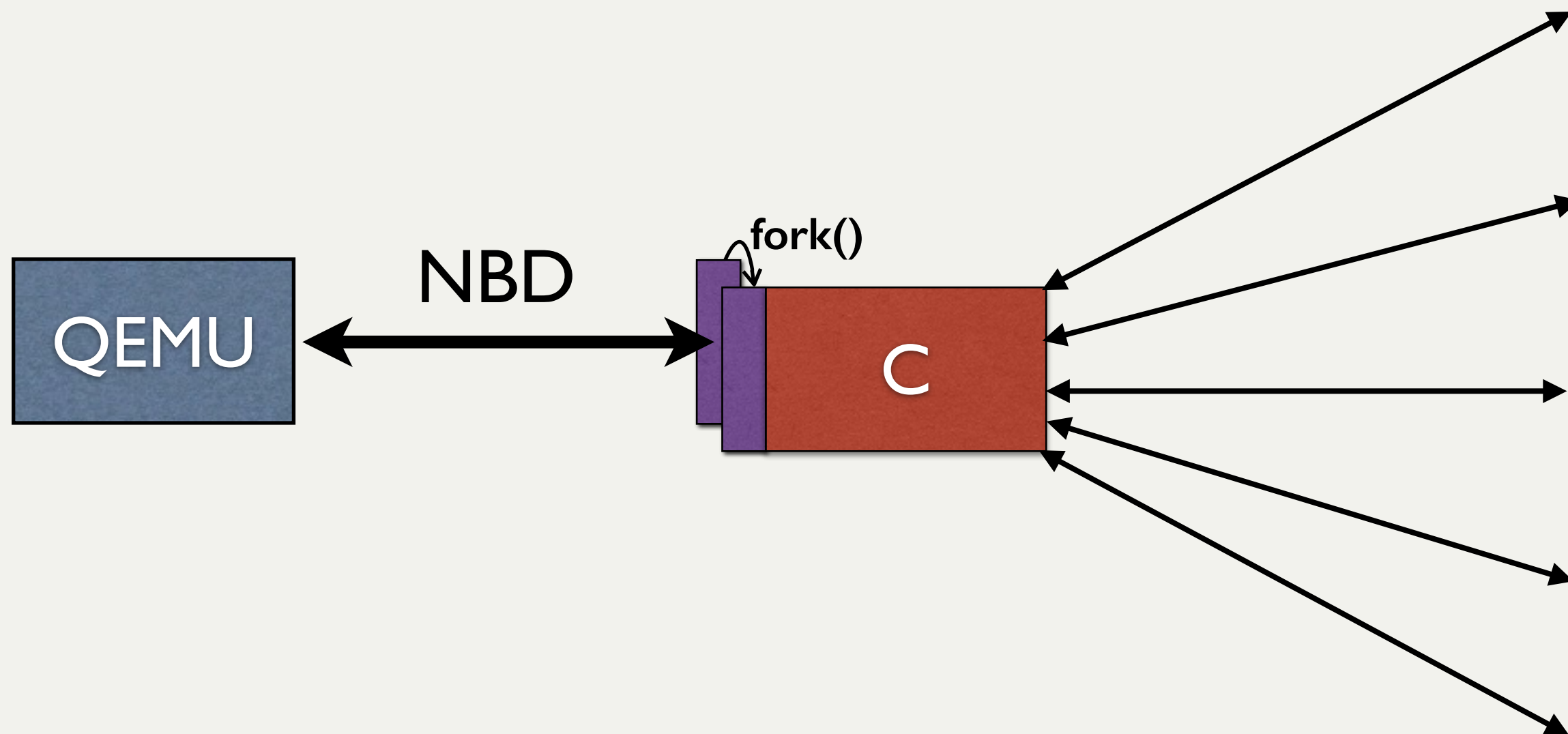


- 任何时候，恢复到绝对正确的运行状态!!!



不朽之盾——起死回生

- 任何时候，恢复到绝对正确的运行状态!!!

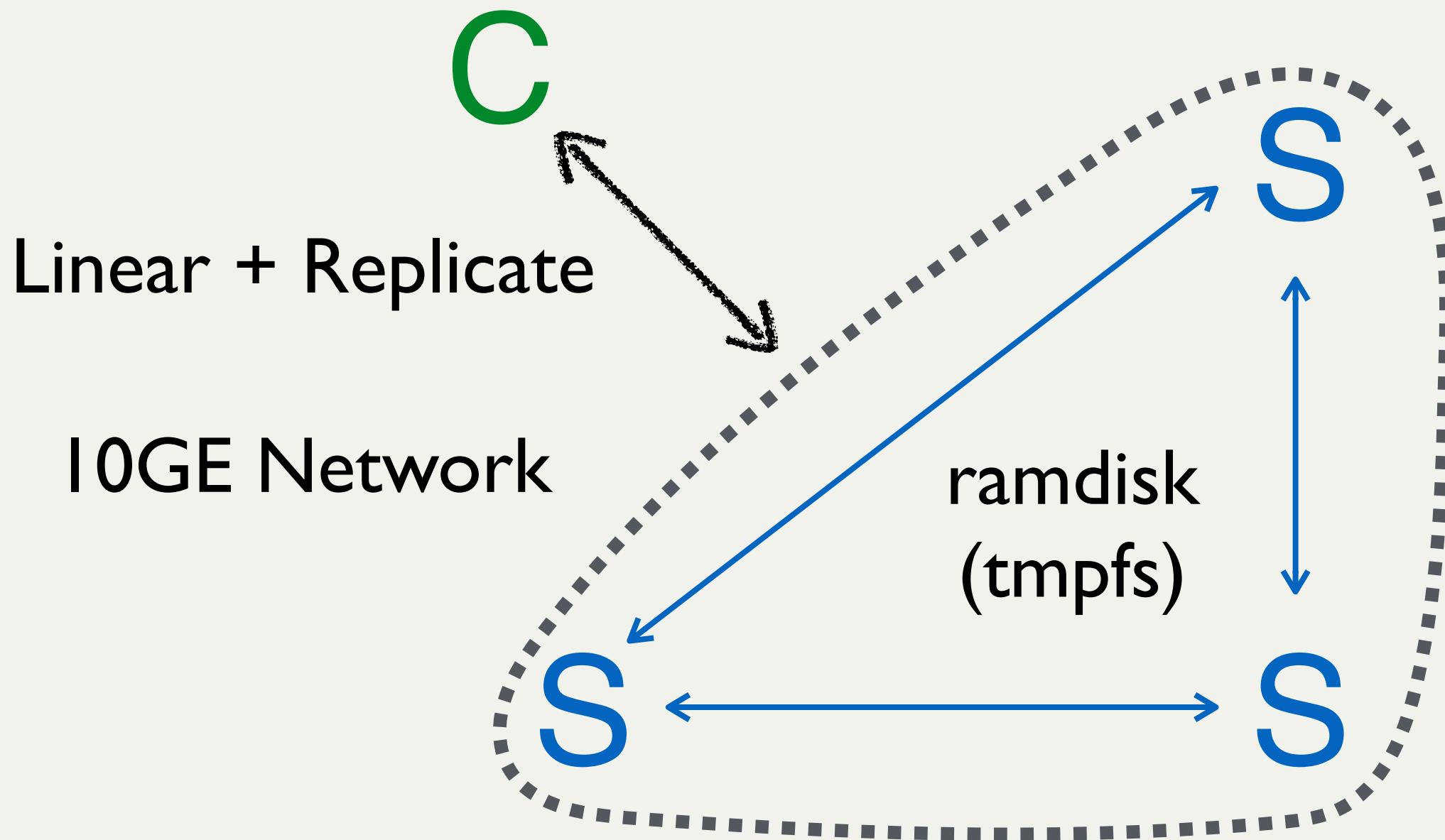


内容提要



- 个人简介
- 云计算@美团
- 云计算对块存储的需求
- 设计与实现
 - ❖ 设计理念、存储结构、并发模型、
 - ❖ 系统架构、写入策略、错误防护
- 性能测试

测试环境



fio测试结果



	吞吐率(KB/s)	客户端CPU	服务端CPU (x3)
读	1,168,282	98%	15%
写	976,428	90%	120%

	IOPS	客户端CPU	服务端CPU (x3)
读	61,340	96%	43%
写	40,438	95%	177%

与Ceph对比效率



服务端：

21282 root	20	0	809m	80m	11m	S	37.4	0.1	6:38.88	ceph-osd
17771 root	20	0	802m	80m	10m	S	27.8	0.1	4:54.46	ceph-osd
19448 root	20	0	839m	79m	10m	S	27.1	0.1	6:42.14	ceph-osd
18580 root	20	0	824m	82m	10m	S	26.5	0.1	5:52.29	ceph-osd
20392 root	20	0	798m	77m	10m	S	13.9	0.1	4:02.15	ceph-osd

123.7% @ 4,101读IOPS
对比：43% @20,446读IOPS
提升**14**倍效率

客户端：5xx% @16,407读IOPS
对比：96% @61,340读IOPS
提升**21**倍效率

测试结果

	延迟(ms)	
	均值	标准差
读	0.166	0.05911
写	0.279	0.04703

对比:

```
[root@ ~]# ping 10.10.10.10
PING 10.10.10.10 (10.10.10.10) 56(84) bytes of data:
64 bytes from 10.10.10.10: icmp_seq=1 ttl=61 time=0.141 ms
64 bytes from 10.10.10.10: icmp_seq=2 ttl=61 time=0.152 ms
64 bytes from 10.10.10.10: icmp_seq=3 ttl=61 time=0.169 ms
64 bytes from 10.10.10.10: icmp_seq=4 ttl=61 time=0.091 ms
64 bytes from 10.10.10.10: icmp_seq=5 ttl=61 time=0.117 ms
64 bytes from 10.10.10.10: icmp_seq=6 ttl=61 time=0.169 ms
^C
```



Q&A

QCon

全球软件开发大会

美团云
Meituan Open Services

谢谢!

Brought by **InfoQ**

International Software Development Conference