China Lustre Workshop 2011

Lustre开发使用经验及展望

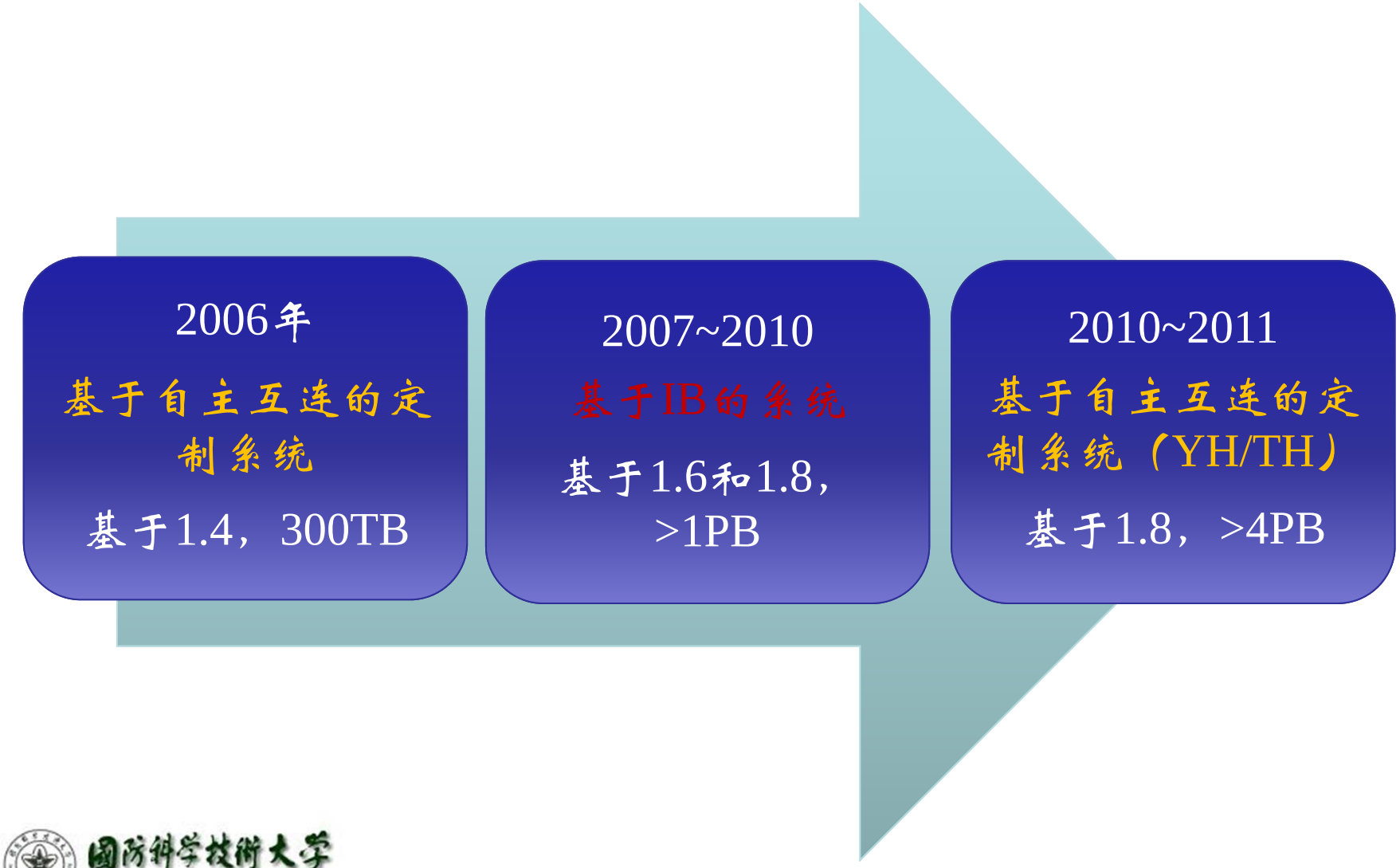
周恩强
国防科技大学计算机学院计算机所
eqzhou@nudt.edu.cn

國防科学技術大学
National University of Defense Technology

研究小组基本情况

- NUDT并行文件系统研究小组
  - 教员+学生，工程任务为主，兼顾学术研究
  - 主要工作为**YH（TH）**系统定制并行文件系统
  - **2003**年以前，研究使用**NFS**和**PVFS**
- 2004年开始评估、分析和定制开发lustre
  - 移植，体系结构适配
  - 优化，性能最大化
  - 新功能开发

基于Lustre的文件系统部署



Lustre的使用经验

部署

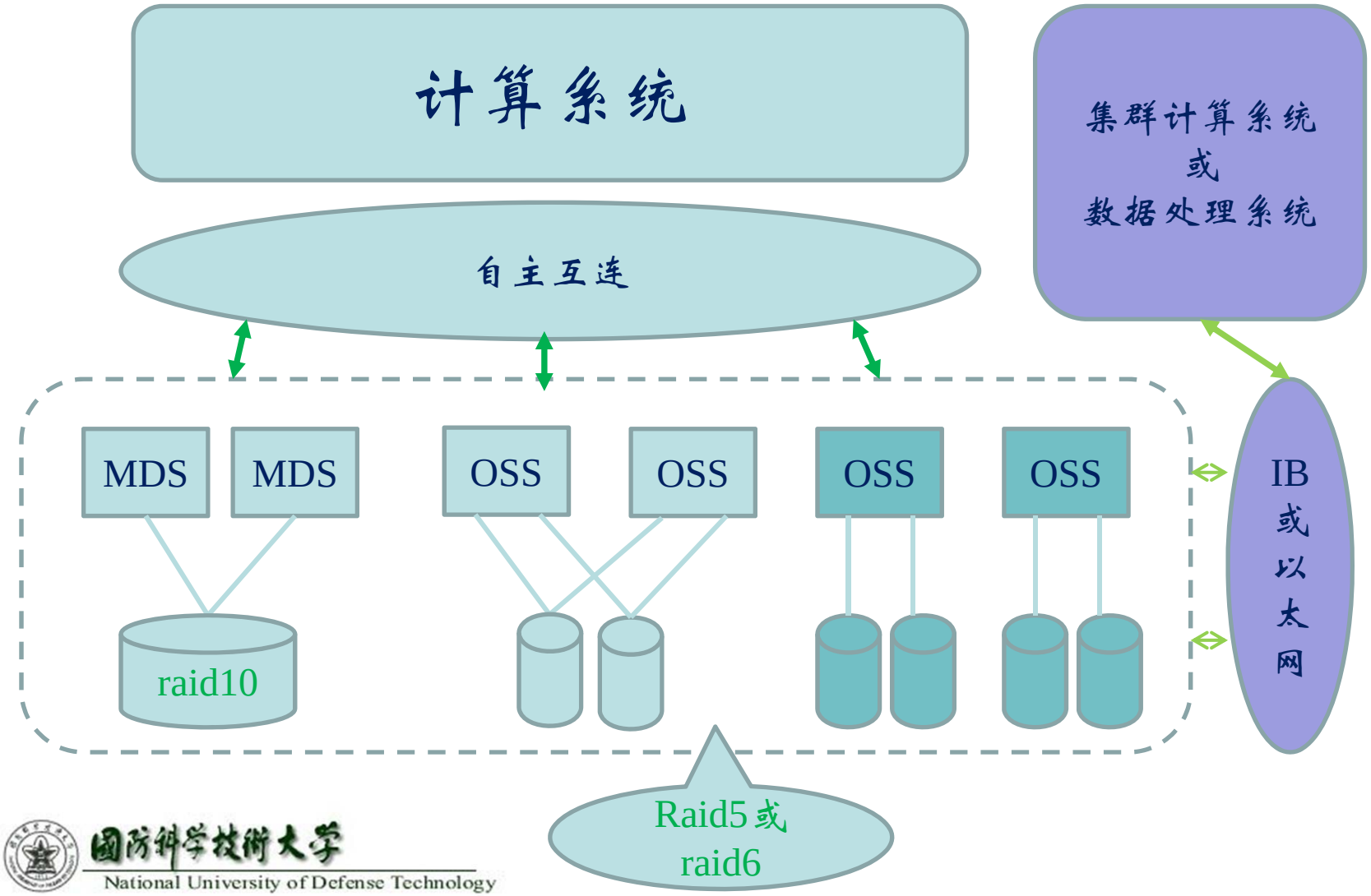
性能

管理

可靠性

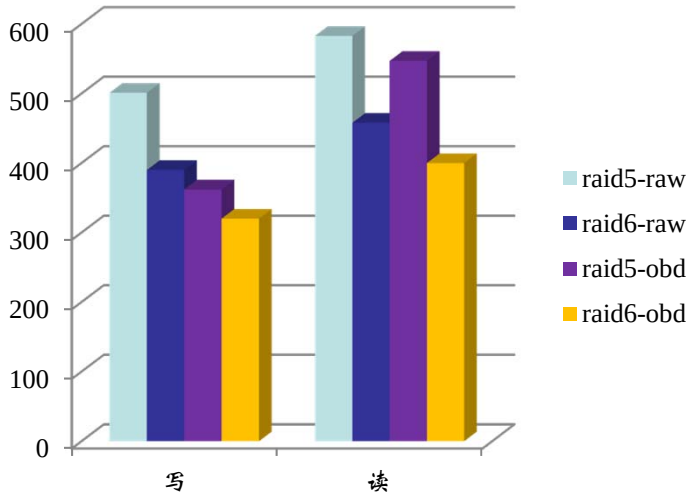
数据备份

典型部署形态


计算系统
集群计算系统
或
数据处理系统
自主互连
MDS
MDS
OSS
OSS
OSS
OSS
IB
或
以
太
网
raid10
Raid5或
raid6

部署-平台选型

- 硬件选型
  - **Obdfilter-survey测试**
    - 单设备最大带宽
    - 多线程条件下的变化曲线
  - **8~10个disk一个RAID**
    - 内置盘阵，较大规模部署
    - 外置盘阵，中小规模部署
    - **write-back cache enable，backend battery**



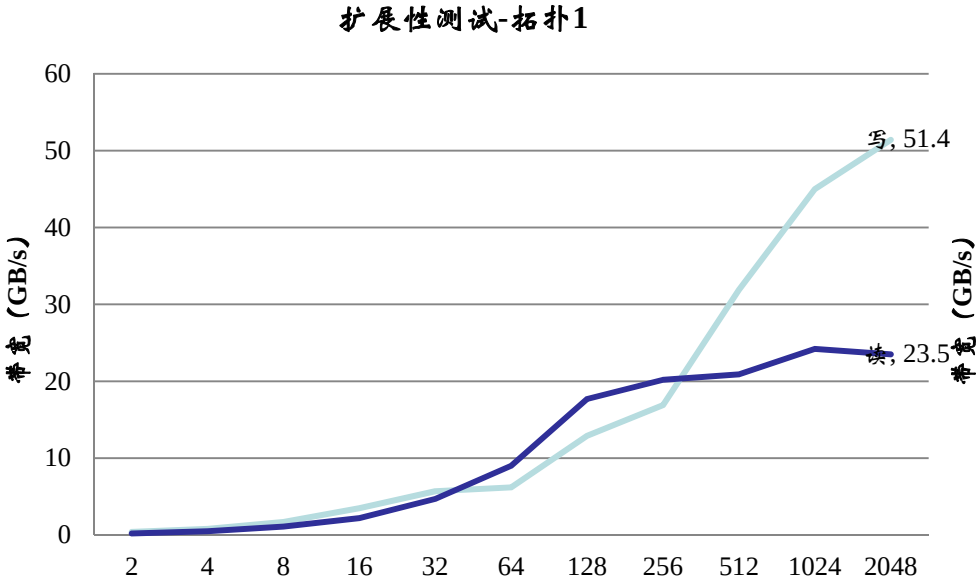
- tradeoff，性能、可靠性和成本

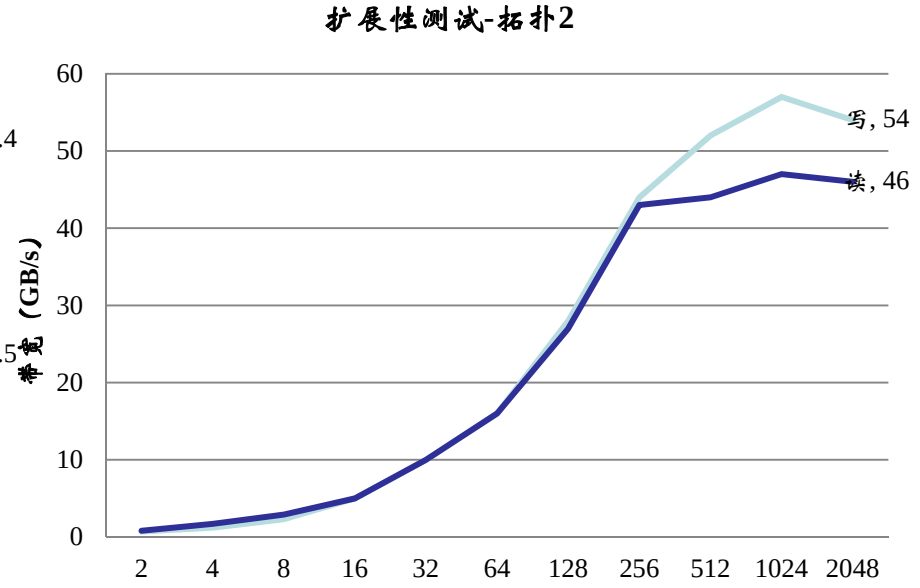
部署-使用原则

- 分区使用
  - 单个分区，64个OSS，2个MDS
  - HA分区和BW分区
  - 顶层虚拟化使用模式（视图和调度）
- 使用多套网络
  - 自主互连网络，计算结点高速I/O
  - Infiniband网络，共享数据
  - 以太网，共享数据和管理

性能-网络拓扑的影响

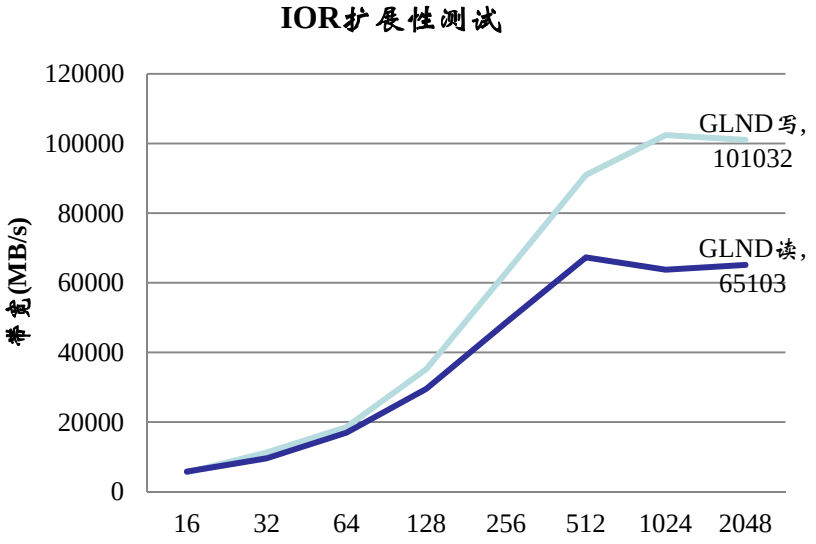
- IOR -b 128m -t 1m -w -r -F –C
- 较大规模系统的网络拓扑设计需综合考虑I/O链路的网络路径和带宽分配

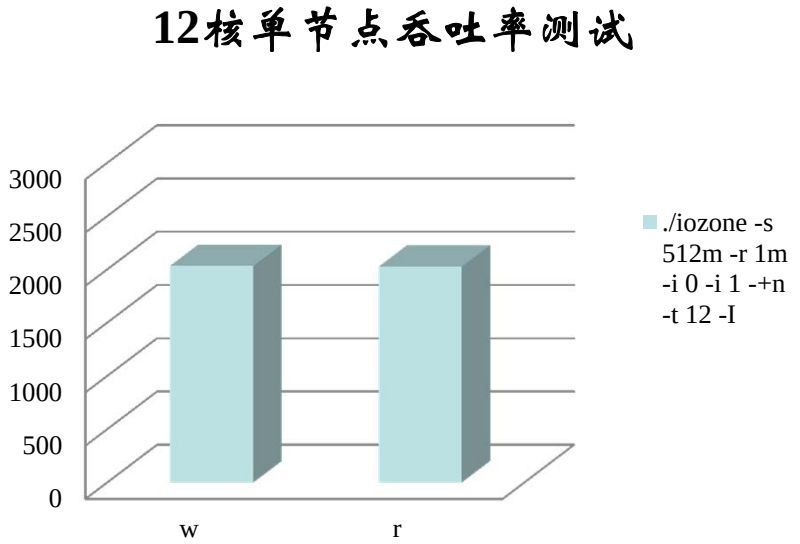




性能-顺序读写模式

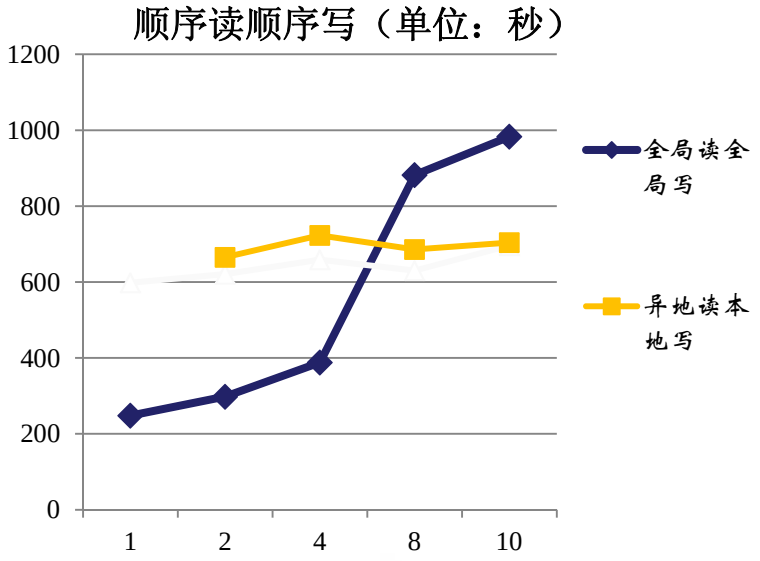
- 顺序读写性能非常好
  - Client write cache，Read ahead cache
  - cache+1MB-RPC+ldiskfs

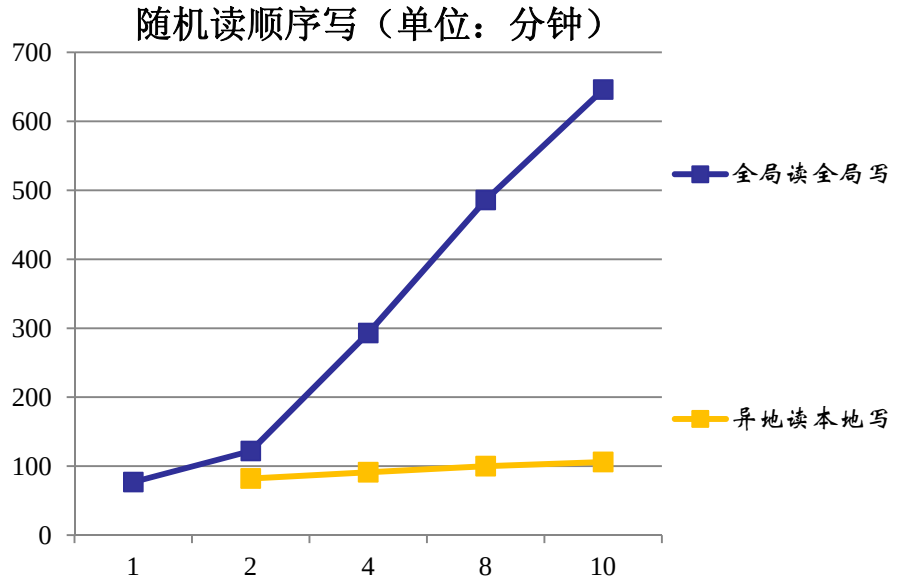




性能-随机读写模式

- 随机读写
  - Pipeline失效，read ahead带来代价
- 典型密集数据处理应用
  - 数据分析阶段，文件读成为瓶颈





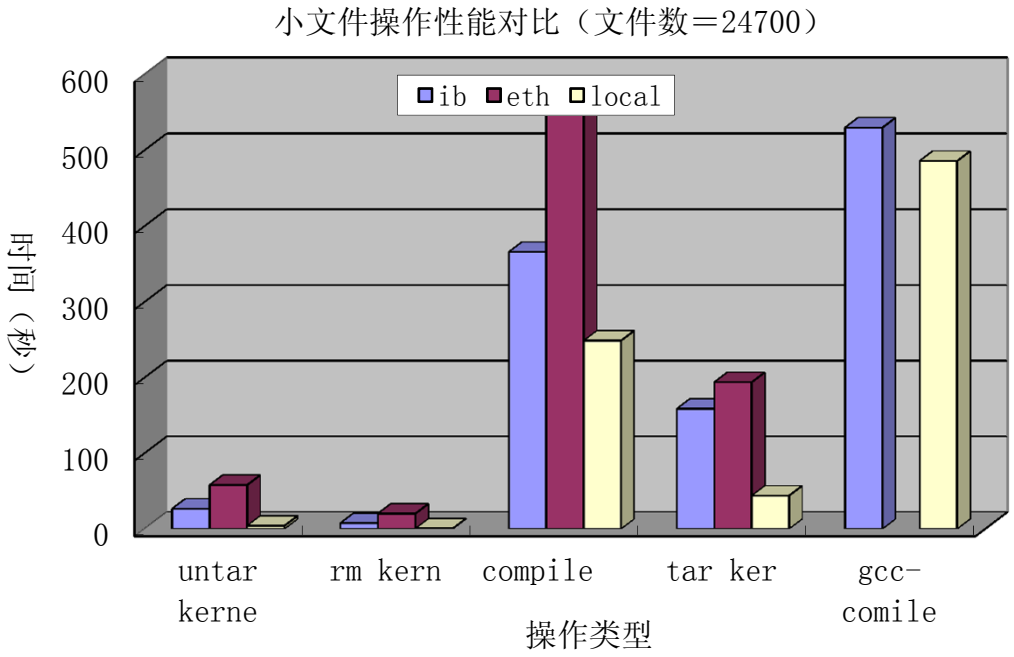
性能-MDS

- 并发文件操作
  - **create >10000op/s**，可接受
  - **readdir**，终端用户会抱怨

- 整个文件系统中的最脆弱点
  - **MDS**忙时，**mds threads**有时处于**D**状态，会导致系统僵死

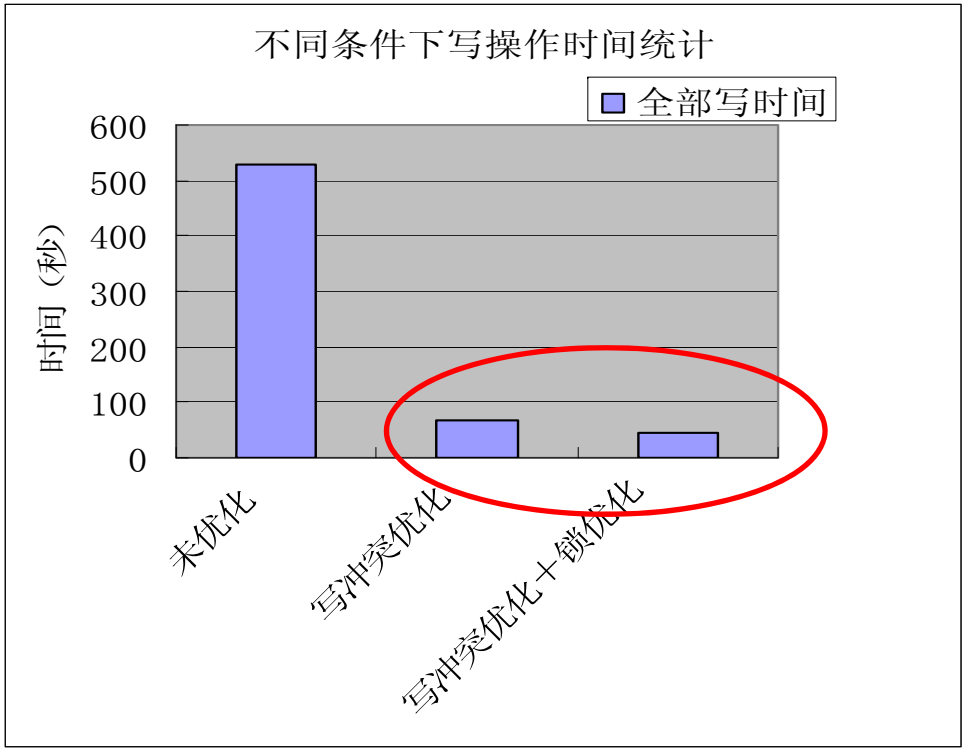
性能-MDS

- 元数据访问延迟
  - SSD，有局限性
  - 元数据性能对网络延迟比较敏感



性能-小粒度访问

- Lock
  - 消除每次IO操作的lock请求，30%的好处

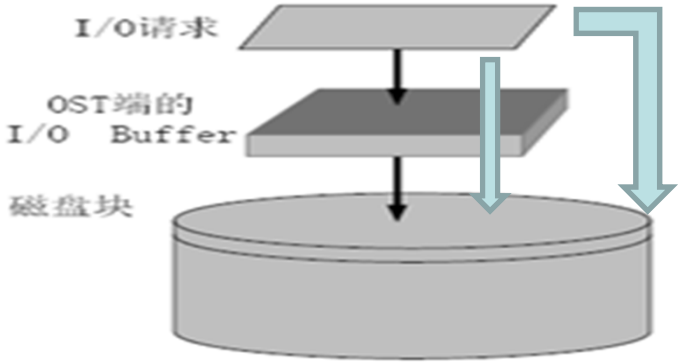


性能-负载均衡

- 空间均衡
  - **Round-Robin**，空间加权
  - 常常需要人工干预，例如：使用**client**端工具
- 性能均衡
  - 文件按**Round-Robin** 方式创建
  - 对并行作业性能影响大，**benchmark**带宽下降
    - **OST**自身性能不均衡
    - 应用程序间相互干扰
    - **MDS**还不够聪明？

性能-OSS端cache

- OSS端的cache有什么作用
  - 对于大文件好处不明显，实际应用情况下**read**命中率低，建议关掉**OSS**的**cache**
  - 对于小文件会有好处，如何利用？



管理-服务端管理和监控

- MDS和OSS采用单映像管理模式
  - 网络引导，自动配置和模块部署
  - 单点管理
    - 成员管理，文件系统配置和格式化
    - 参数调整，**OST**分区和屏蔽，数据迁移，扩展容量，**HA**服务管理
- OSS服务监控
  - **OSS**状态
  - **OST**和盘阵联动，屏蔽**unhealthy**的**OST**
  - 和**IPMI**管理联动，**reboot**恢复

```
MDS0: YHFS-MDT0000 774  running(healthy)
OSS0:  YHFS-OST0000 775   running(healthy)   YHFS-OST0001 775 running(healthy)
OSS1:  YHFS-OST0002 775   running(healthy)   YHFS-OST0003 775 running(healthy)
```

管理-client监控

- 早期在计算结点上运行监控模块
  - 大规模系统必须控制监控开销
- 现在采用主动轮询方式
  - 监控**ost**的客户端连接数
  - 客户端并发**Check**，客户端数**7000+**，几十秒
- dmesg是主要的故障诊断信息来源
  - 不易理解

可靠性和可用性

- 可接受的稳定性
  - 对于**HPC**领域，**Lustre**足够稳定
- 可用性和可靠性均依赖硬件
  - **HA**方式和硬件**RAID**
- 网络故障基本上可自动恢复
  - 有时需要客户端**reboot**
- Ldiskfs error
  - 多数情况**e2fsck**可恢复

数据备份

- 国内Lustre的用户如何备份数据？
  - 备份的速度慢，几十**MB/s**
  - 往往没有备份窗口
  - 选择性的备份少量数据
- 缺少有效的lustre备份工具
  - 需要支持并发的文件备份工具
  - 需要提高**find**的速度
  - 我们曾经尝试，**e2scan+parallel tar**

针对Lustre的定制开发

- GLND
  - 用于自主互连网络的**LND**，基于**RDMA**的协议
  - **1.4**，**1.6**，**1.8**，大范围部署使用
- 增强容错功能的条带设计
  - **RAID1 ,0+1**，针对重要数据的**availability**
  - **1.4 only**，小范围部署使用
- Multi-rail IB，1.6，研究
- Lock优化，1.4，研究
- 小文件性能优化，1.6，研究

我们的需求

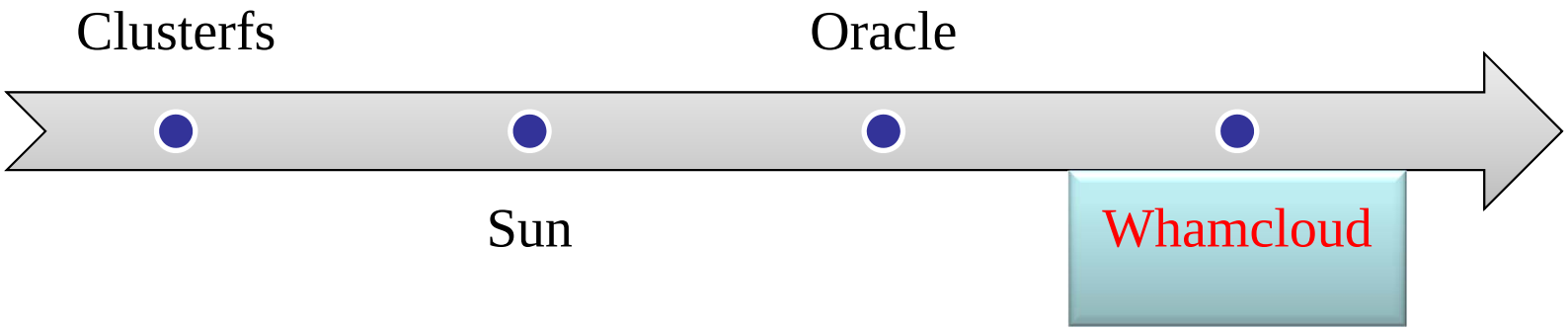
Scalability

- Scalable communication
- MDS performance improvement
- Consistency options

Reliability

- File replication
- Robust backend FS

前景展望



Lustre会被更广泛的关注、部署应用	• 更多更大规模的系统会使用lustre
我们会持续关注Lustre	• 更好的集成和优化使用，扬长避短
我们的主要目标	• scalability，reliability

The End

请多指正！