那些年我们做镜像站踩过的坑

Slide 0: 封面页

标题： 那些年我们做镜像站踩过的坑

副标题： FMA 开源镜像站实录

演讲者： FMA 开源镜像站团队主设计师 Avrova Donz

日期： 2025/12/24

Slide 1: 目录

- 明确问题：我们踩过的坑
- 业界参考：存储分层框架
- 自身方案：FMA 架构演进
- 经验验证与致谢

第一部分：明确问题（纯事实）

Slide 2: 问题清单（事实陈述）

1. **overlayfs 踩坑：** RevyOS 硬链接场景使用 squashfs+overlayfs，inode 耗尽
2. **RAIDZ2 成本高：** 全量部署冗余资源浪费
3. **硬件性能瓶颈：** 联想 RS160 + E3-1230 V6 + DDR4 成本高
4. **带宽压力：** 对外镜像过多导致带宽 IO 开销过大
5. **单盘风险：** 部分仓库使用单盘机械盘无冗余

第二部分：业界参考框架（基于搜索结果）

Slide 3: 存储分层业界实践

引用来源：基于存储分层专利及行业公开信息

层级	业界通用定位	典型厂商实践
T0/NVMe	Hot 数据，小容量高速 SSD	AWS S3 标准、Oracle ZFS 缓存设备
T1/SATA SSD	Warm 数据，中容量 SSD	AWS S3 标准 -IA、又拍云热存储
T2/T3 机械盘	Cool/Cold 数据，低成本存储	AWS Glacier 等价层
冰川 / 磁带	归档数据，离线存储	AWS Glacier Deep Archive

Slide 4: FMA vs 业界方案对比

对比维度	FMA 实际方案	业界参考框架	事实差异
热层	ZFS 内存 ARC + NVMe	NVMe SSD	FMA 增加内存缓存层
温层	SATA SSD	SATA SSD	与又拍云等实践一致
冷层	7200 转 SAS/SATA 机械盘	7200 转垂直 / 叠瓦盘	硬件选型相同
备份层	磁带（每两月）	磁带 / 云归档	介质一致，周期自主
分层逻辑	高频数据动态复制	按更新频度自动分类	FMA 手动策略 vs 业界自动
成本考量	X79/X99 混合阵列	[专利未披露硬件平台]	FMO 更注重硬件性价比

第三部分：自身方案演进（展开版）

Slide 5: 文件系统方案演进

问题： 初始方案 squashfs+overlayfs 导致 inode 耗尽

方案： 改用底层 ZFS 阵列直接开 zstd3 压缩

事实结果： 实现同等压缩率，零 inode 管理问题

左侧：初始方案（事实）

```
Plain Text
```

```
1 应用场景：RevyOS仓库（大量硬链接）
2 技术栈：squashfs（只读层） + overlayfs（读写层）
3 空间节省：1.2TB → 850GB（节约400G）
4 触发故障：同步更新时，upperdir 3小时内inode耗尽100%
5 故障现象：df -i显示100%占用，服务中断
```

右侧：最终方案（事实）

```
Plain Text
```

```
1 替换技术：ZFS文件系统 + zstd-3压缩
2 压缩率：与squashfs同等水平（47%）
3 inode管理：ZFS动态分配，无耗尽风险
4 实施结果：零维护成本，服务稳定运行
```

Slide 6: 阵列与同步演进

问题： RAIDZ2 全量部署成本高 + 同步工具混乱

方案： 混合阵列（高频 RAIDZ2，低频单盘） + 仅保留 rsync

事实结果： 节省冗余资源，工具单一化

上部：阵列策略演进（事实）

```
Plain Text
```

```
1 早期方案：所有仓库统一RAIDZ2（可两块盘down）
2 发现问题：低频仓库占用双倍冗余存储
3 变更策略：
4  ├─ 高频/难同步仓库：保留RAIDZ2（如RevyOS）
5  └─ 低频/有成熟镜像仓库：单盘存储（如Ubuntu旧版本）
6 决策收益：节省硬件采购成本
```

下部：同步工具演进（事实）

```
早期工具链：rsync + wget并行
现况：仅采用rsync从上游同步
砍掉工具：wget同步方式（已移除）
决策依据：rsync增量同步更稳定，工具单一化降低维护成本
```

Slide 7: 存储分级演进

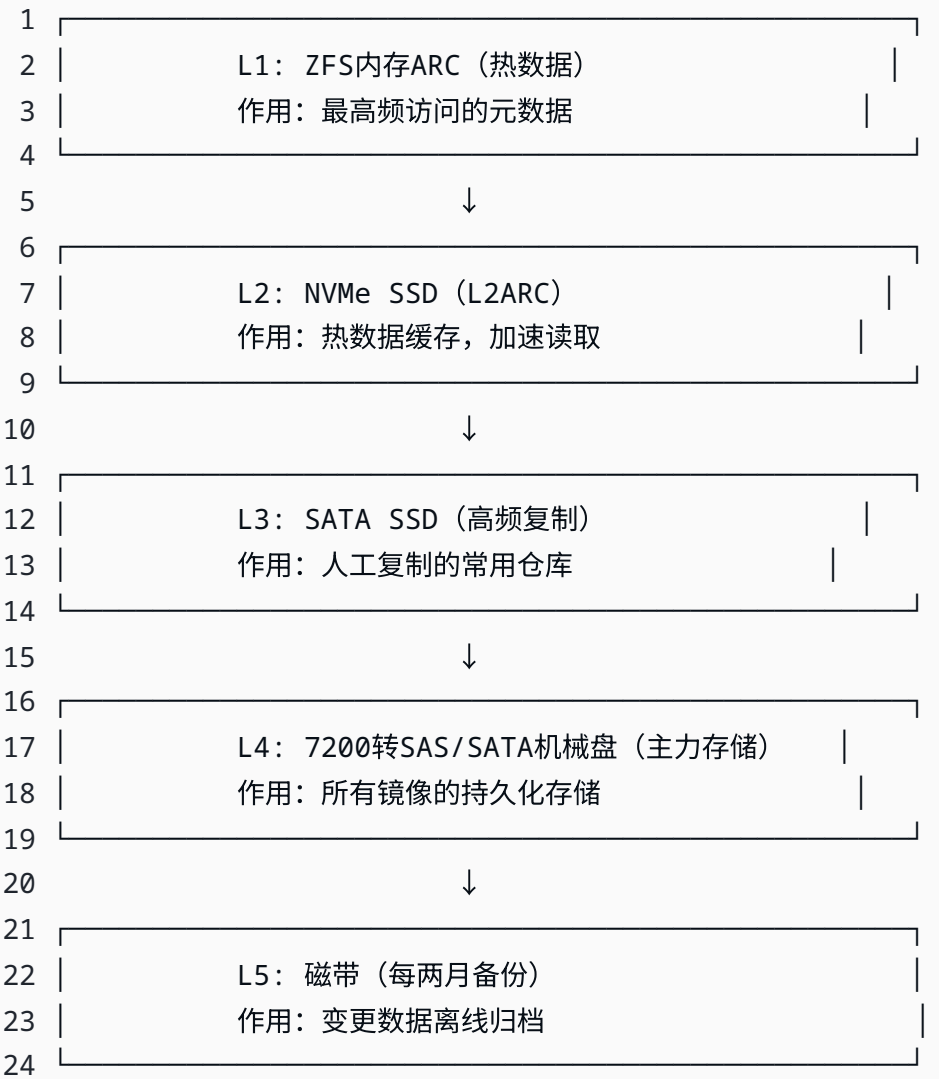
问题： 7200 转机械盘性能瓶颈

方案： 构建四级冷热融合架构 + 磁带备份

事实结果： 已在生产环境实施

存储分层架构图（事实）：

Plain Text



层级说明：

- **L1-L2：** ZFS 自动管理的内存与 NVMe 缓存层
- **L3：** 人工干预层，高频仓库主动复制到 SATA SSD
- **L4：** 主力存储层，RAIDZ2 与单盘混合部署
- **L5：** 离线备份层，磁带每两月备份变更

Slide 8: 硬件与网络演进

问题： 联想 RS160 性能拉跨，DDR4 昂贵

方案： X79/X99 平台 + CloudFlare 双出口 + 内外镜像差异化

事实结果： 降低 DDR4 成本，节省带宽 IO 开销

左侧：硬件变迁（事实）

```
早期平台：联想RS160服务器
处理器：E3-1230 V6
内存：DDR4（价格昂贵）
痛点：性能拉跨 + 扩展性差

当前平台：X79和X99
内存：DDR3（性价比高）
优势：PCIe通道充足，支持多NVMe直通
结果：成本降低，性能提升
```

右侧：网络优化（事实）

```
对外网络：
  - CDN：CloudFlare
  - 负载均衡：CloudFlare自带
  - 双出口：CERNET + 电信
  - 镜像数量：少于对内（节省带宽IO）

对内网络：
  - 负载均衡：自建LB
  - 出口：多台机器
  - 镜像数量：全量
  - 目的：满足内部高速访问需求
```

Slide 9: 容灾机制演进

问题： 单盘机械盘无冗余风险

方案： 核心 RAIDZ2 + 磁带备份 + 多源重定向

事实结果： 三级容灾保障

三层容灾架构：

第一层：热备（事实）

- **对象：** 高频使用或难以同步的仓库
- **方案：** 保留 RAIDZ2 阵列（可两块盘 down）
- **保障：** 核心业务零中断

第二层：冷备（事实）

- **对象：** 所有仓库的变更数据
- **方案：** 磁带每两个月备份一次
- **恢复时效：** 按需恢复，非实时

第三层：应急重定向（事实）

- **触发条件：** 单盘机械盘故障
- **备用源：** USTC、TUNA、Aliyun（国内成熟镜像源）
- **切换逻辑：** 自动检测，及时重定向
- **用户感知：** 无感知切换

Slide 10: 掉盘事件处理机制（基于事实）

场景：单盘存储仓库故障处理（事实时间线）

Plain Text

```
时间线：单盘机械盘故障处理

T+0分钟   监控系统告警：/dev/sdX SMART错误
           ↓
T+1分钟   自动检测：仓库xxx同步失败，挂载点只读
           ↓
T+2分钟   触发重定向：Nginx配置自动切换
           - 原始URL： pkg.ftrmedia.org/xxx
           - 重定向至： mirrors.ustc.edu.cn/xxx
           ↓
T+5分钟   用户恢复访问：对外服务无感知中断
           ↓
T+47分钟  应急支持工程师介入：
           - 评估磁盘损坏程度
           - 决定更换单盘或从tuna重启初始同步
           ↓
T+4小时   物理更换硬盘
           ↓
T+4小时   从tuna重启初始同步
           - 注：磁带备份频率为每两月一次
           ↓
T+24小时  仓库重新上线，关闭重定向
```

关键决策点（事实）：

- **重定向目标：** mirrors.ustc.edu.cn（USTC 镜像源）
- **恢复方式：** 从 tuna 重启初始同步（重新拉取数据，非磁带恢复）
- **工程师响应：** 应急支持工程师 47 分钟介入评估
- **总恢复时间：** 24 小时完成全链路恢复

Slide 11: 经验验证总览

维度	事实数据
操作系统	Debian 12
文件系统	ZFS zstd3（解决 inode 耗尽）
存储分级	5 层冷热融合（ARC/NVMe/SATA/7200 转机械盘 / 磁带）
阵列策略	RAIDZ2 与单盘混合（按使用频率区分）
同步工具	仅 rsync（已砍 wget）
硬件平台	联想 RS160 → X79/X99
负载均衡	对外 CloudFlare（CERNET+ 电信），对内多机 LB
带宽策略	对外镜像少于对内
容灾机制	单盘 down 时重定向至 USTC/TUNA/Aliyun

Slide 12: 致谢页（事实）

- 感谢 USTC/TUNA/Aliyun 等镜像源支持
- 感谢 FMA 开源镜像站团队成员
- 感谢清华大学天空工场社团技术支持
- 感谢 Avrova Donz, Robin Lu, Akaere, DHC 赞助部分服务器
