IT168.com ChinaUnix.net ITPUB

SACC

2020 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2020

架构融合 云化共建

live 2020年10月22日 - 24日网络直播

http://www.itpub.net/

云上会议

高性能存储系统XFS的架构实践

周超勇/字节跳动

http://www.itpub.net/

Agenda

- 背景
- 场景举例
- 带目录存储
- 架构与设计
- 监控
- 开源

http://www.itpub.net/

背景

- XFS是一个用户空间的、面向小文件的、带目录的、高性能存储系统

  定义：在XFS中，小文件是指不超过64MB大小的文件

- XFS来源于RFS（Random access File System）

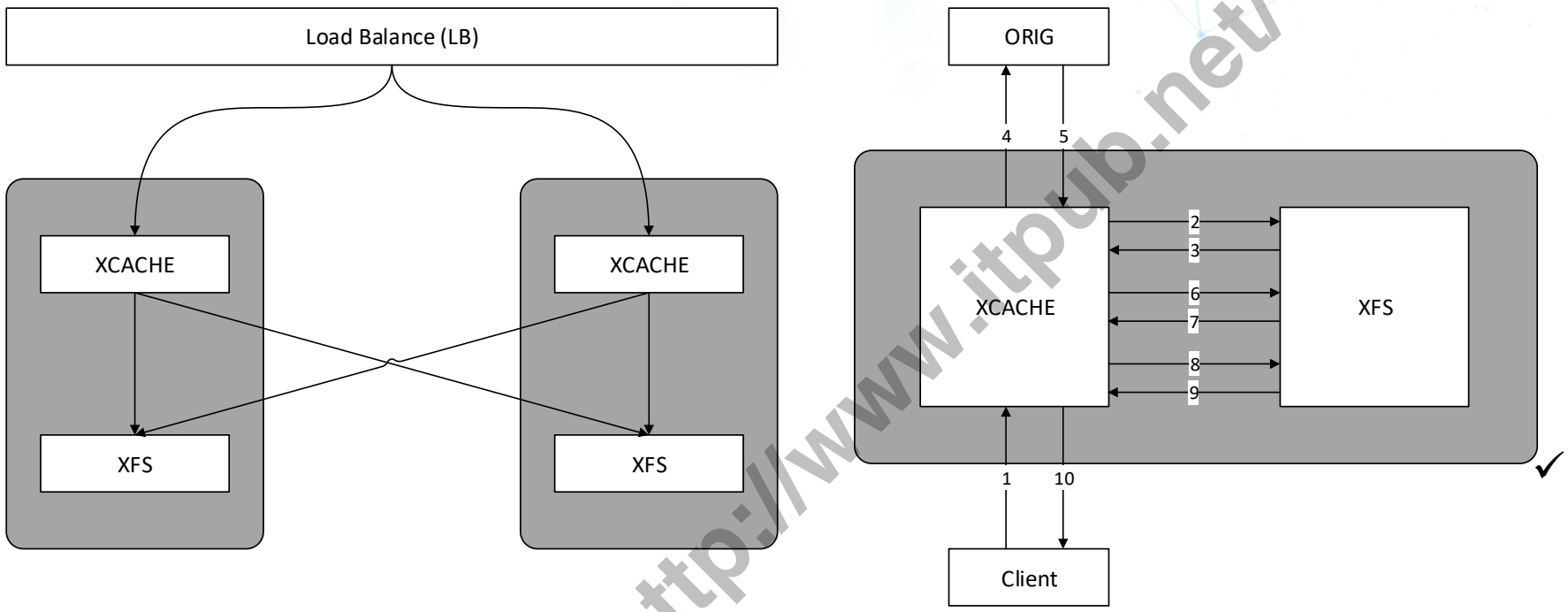
  XFS = RFS + 裸盘管理 + DMA（分级缓存技术）

- XFS和Linux xfs文件系统没有任何关系，只是重名而已

- XFS接近于通用型存储，目前为稳定状态

http://www.itpub.net

场景举例 – CDN CACHE

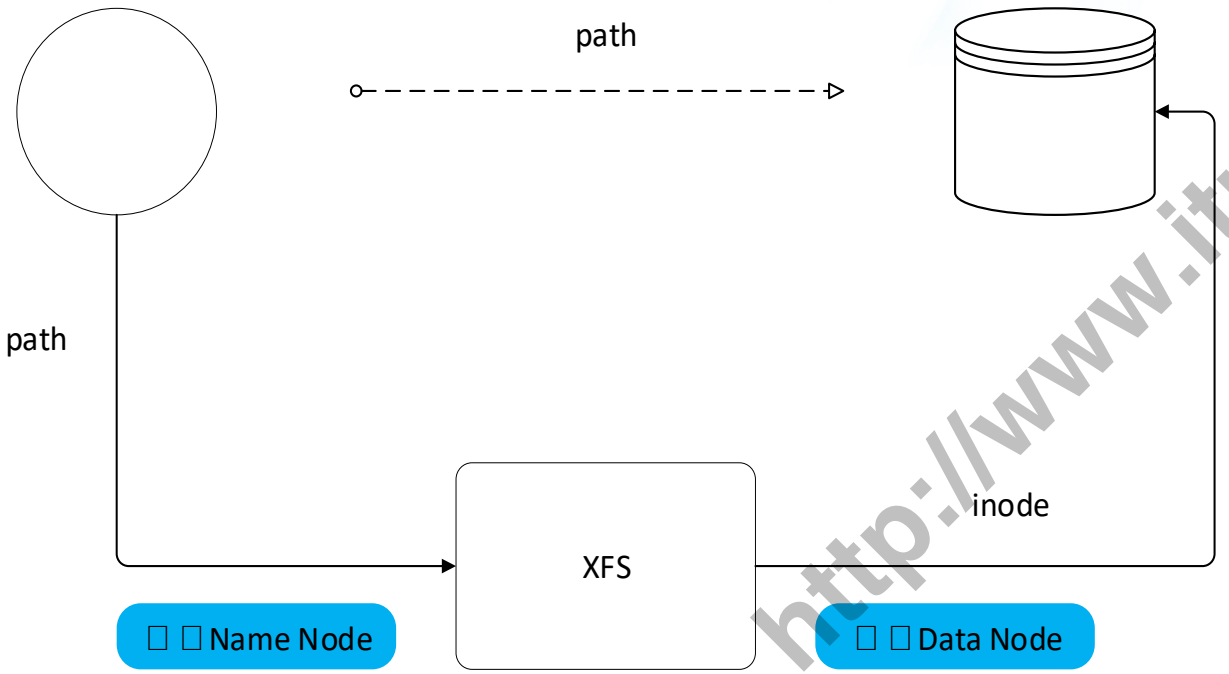


http://www.itpub.net/

- ✓ 禁掉淘汰功能，可作为随机访问的通用版存储系统
- ✓ 仅留一级目录，可作为对象存储系统

```
bgn> hsxfs 0 qlist /www.test.com/1M.dat full on tcid 10.10.67.18 at console
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] [SUCC]
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] 0 # /www.test.com/1M.dat/0
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] 1 # /www.test.com/1M.dat/1
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] 2 # /www.test.com/1M.dat/2
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] 3 # /www.test.com/1M.dat/3
[2020-08-12 20:31:09.021][tid 1172738][co 0x7f04fef05d88] 4 # /www.test.com/1M.dat/4
```

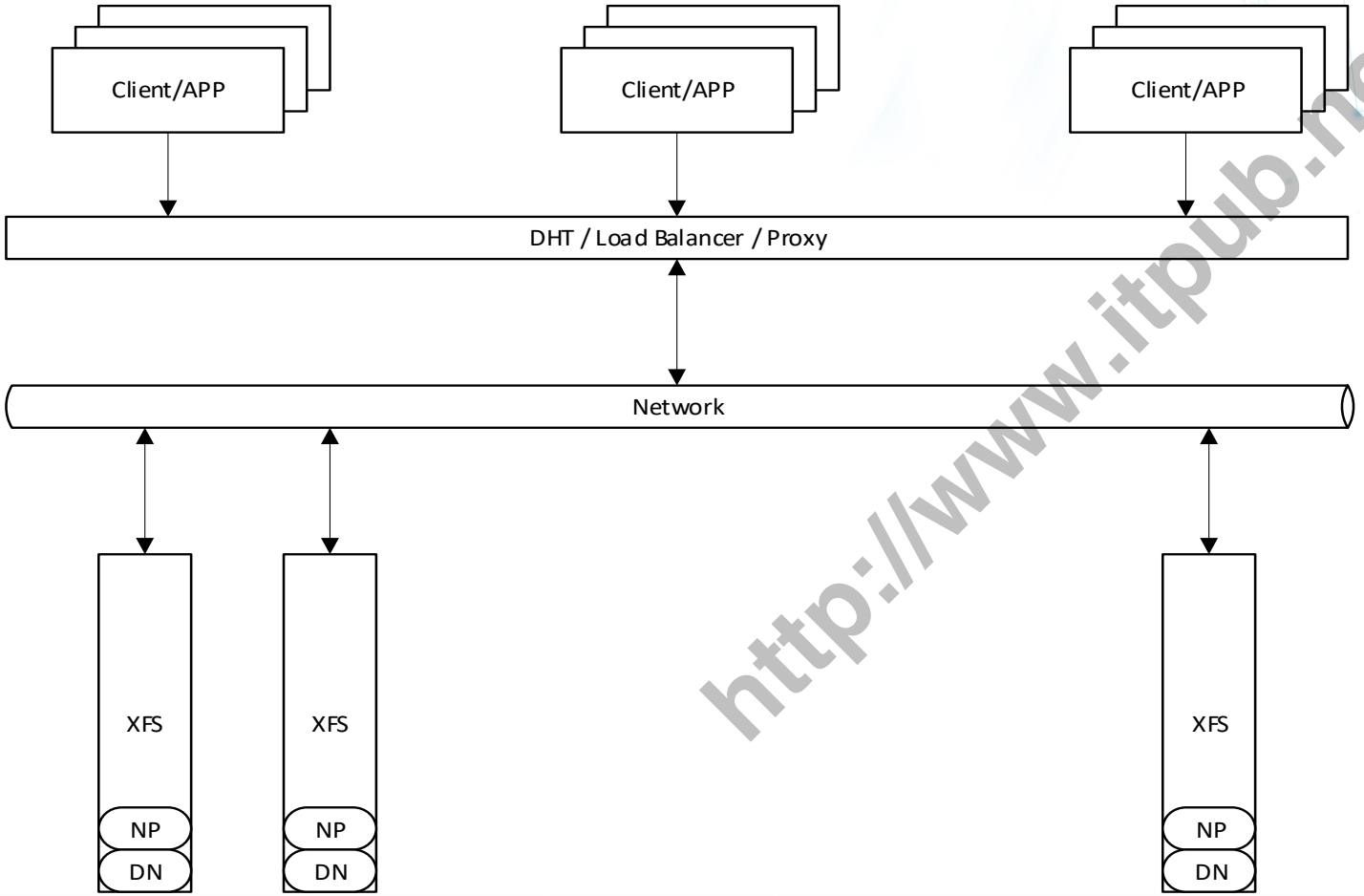
带目录存储



- 存储本质：Name Node + Data Node
- 存储设计：完成两次映射
  - ✓ （path, Name Node） -> inode
  - ✓ （inode, Data Node） -> file content
- XFS设计：带目录的实现为存储场景带来更多的可能
- 带目录存储：最初是为了CDN CACHE刷新功能，实现真实秒刷

http://www.itpub.net/

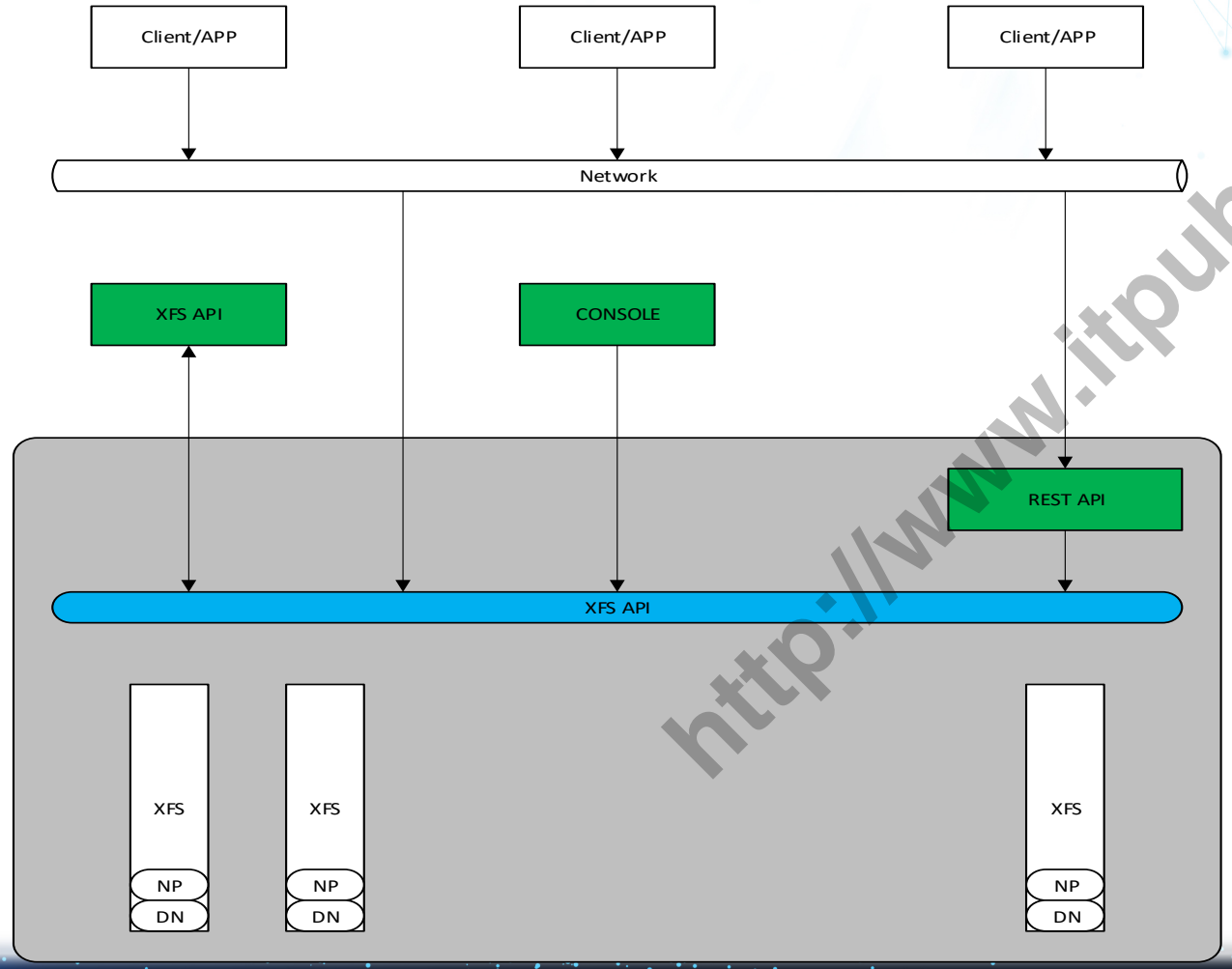
架构与设计



http://www.itpub.net/

- ✓ 单盘单进程管理
- ✓ 节省CPU资源
- ✓ 以盘为单位，幂等性，横向扩展
- ✓ 单盘上下线
- ✓ 前端负载均衡和切片

接口



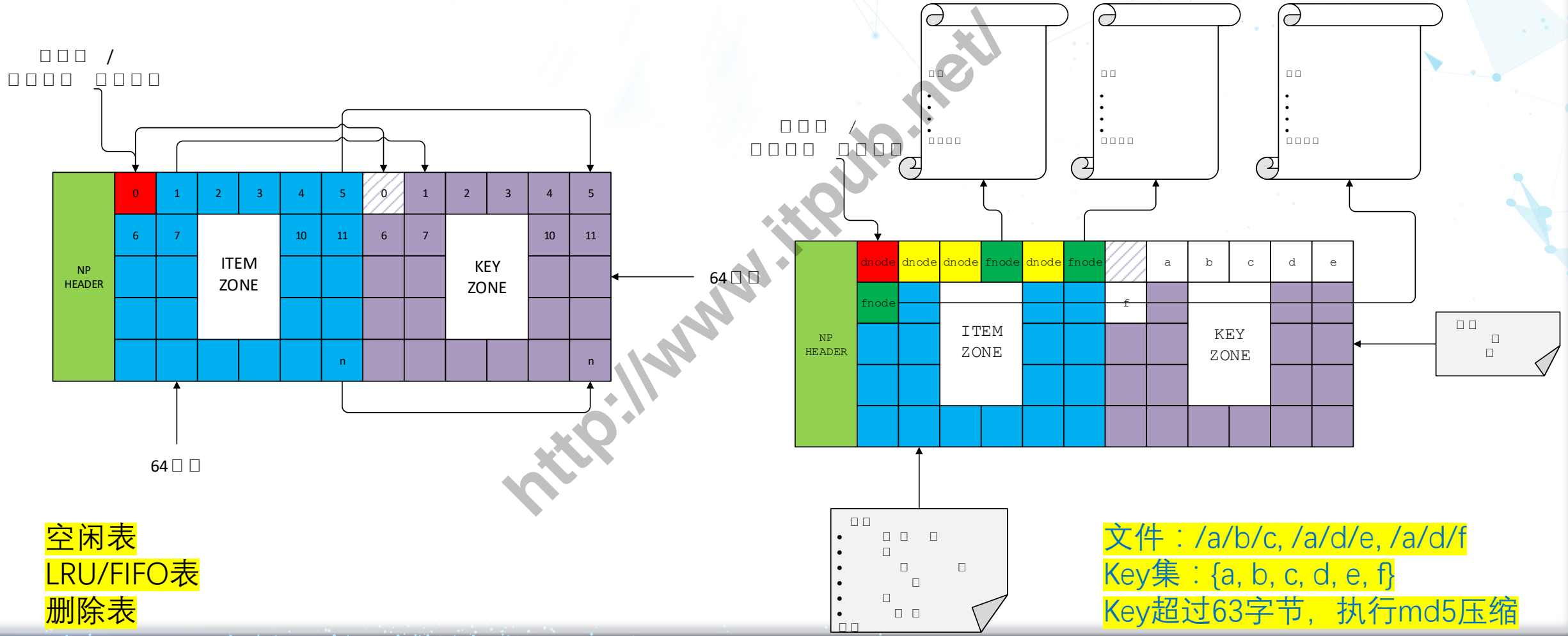
- ✓ 117个XFS API，私有协议
- ✓ CONSOLE在线运维工具，私有协议， XFS API的子集
- ✓ REST API，HTTP协议，支持长连接，XFS API的子集
- ✓ 聚焦：读、写、删

http://www.itpub.net/

设计

NameNode/inode/fnode/dnode

架构融合
云化共建



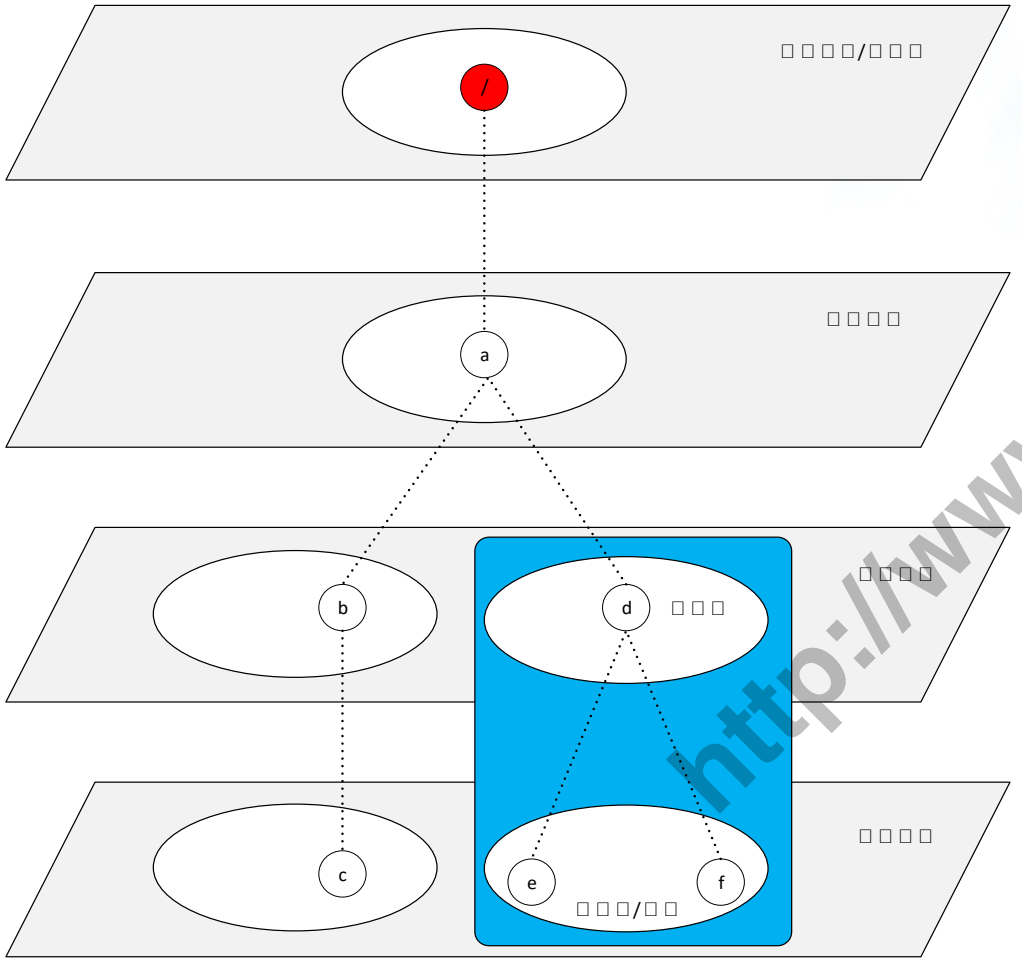
空闲表
LRU/FIFO表
删除表

文件：/a/b/c, /a/d/e, /a/d/f
Key集：{a, b, c, d, e, f}
Key超过63字节，执行md5压缩

http://www.itpub.net/

设计

Name Node/目录结构视图

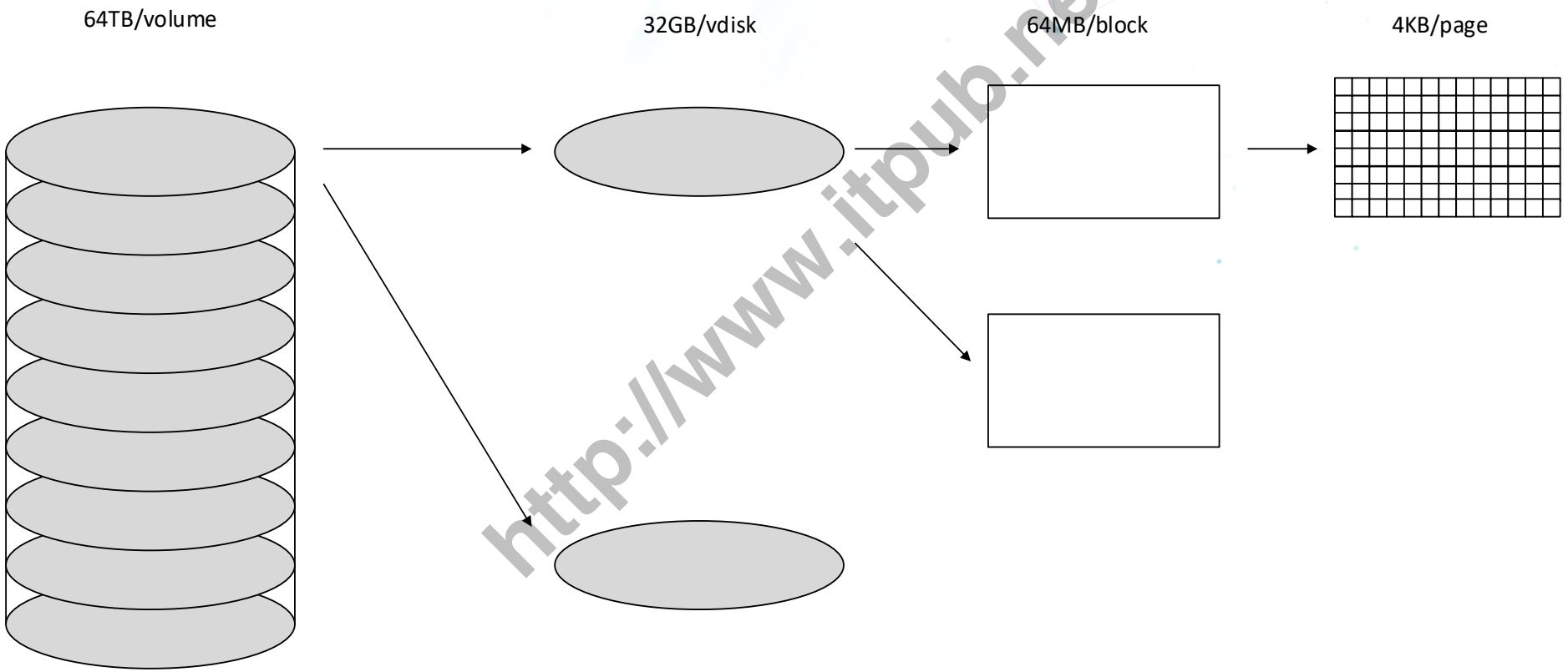


- ✓ 4棵红黑树：
  - {/, a}
  - {a, b, d}
  - {b, c}
  - {d, e, f}
- ✓ 立体、多维、分层
- ✓ 目录层级视角
- ✓ 父子目录视角
- ✓ 描述的是inode的组织形式

http://www.itpub.net/

设计
Data Node分四层

架构融合
云化共建

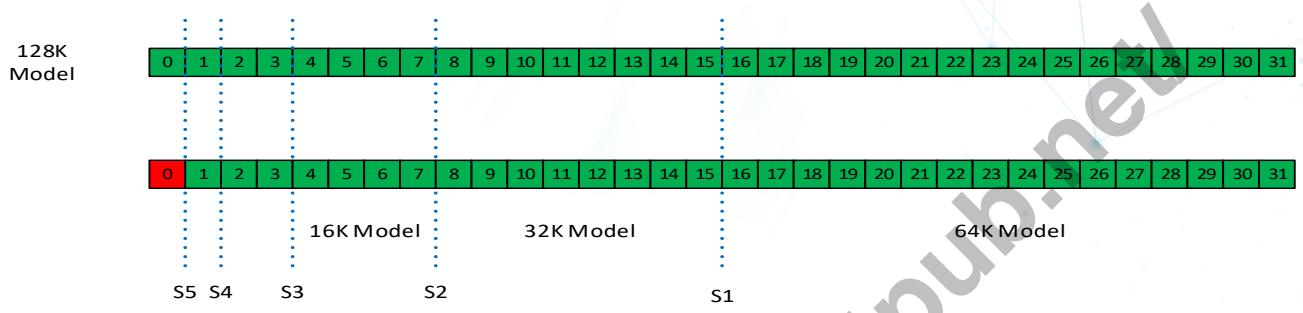


http://www.itpub.net/

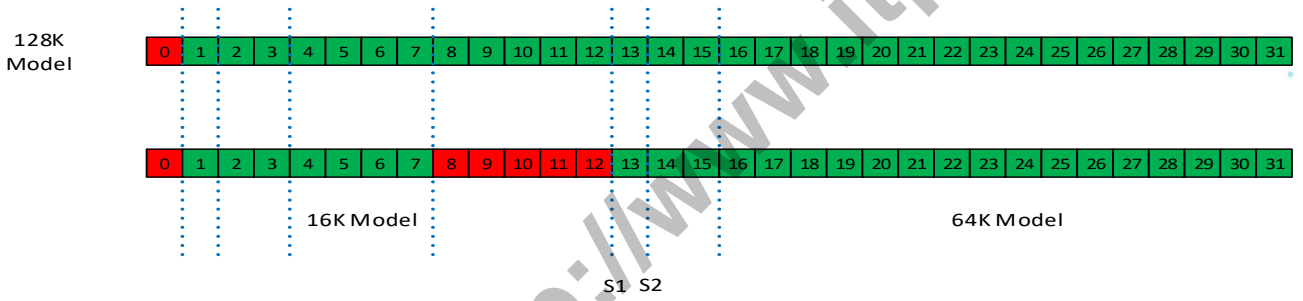
设计
Data Node 分裂与合并

架构融合
云化共建

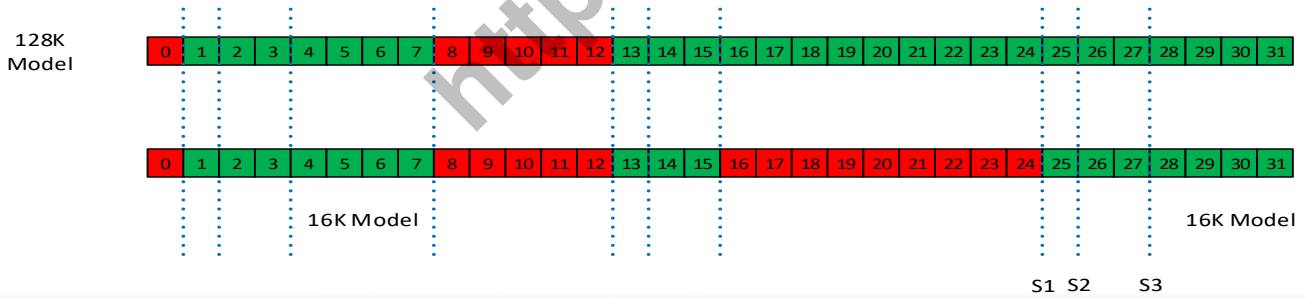
分配1个页



再分配5个页



再分配9个页



- 类似伙伴系统算法
- 特点：
  - ✓ 始终严格对齐
  - ✓ 全程位操作
  - ✓ 分裂与合并互逆
- 分裂算法：

一分为二取其左，反复迭代

- 合并算法：

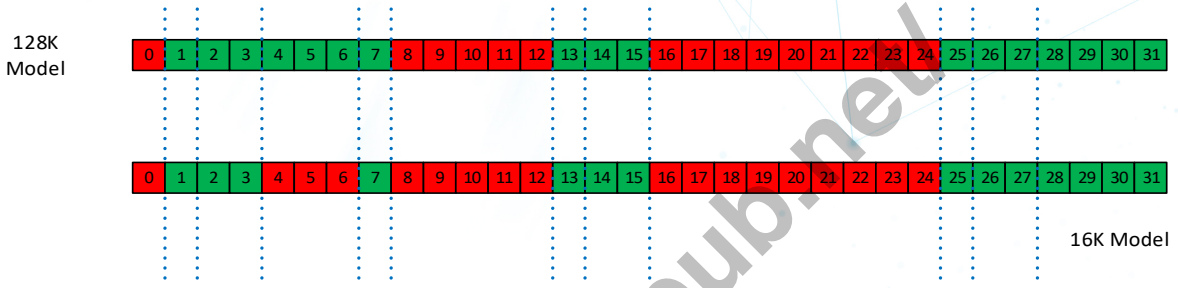
看奇偶，看空闲，左右合并，反复迭代

http://www.itpub.net/

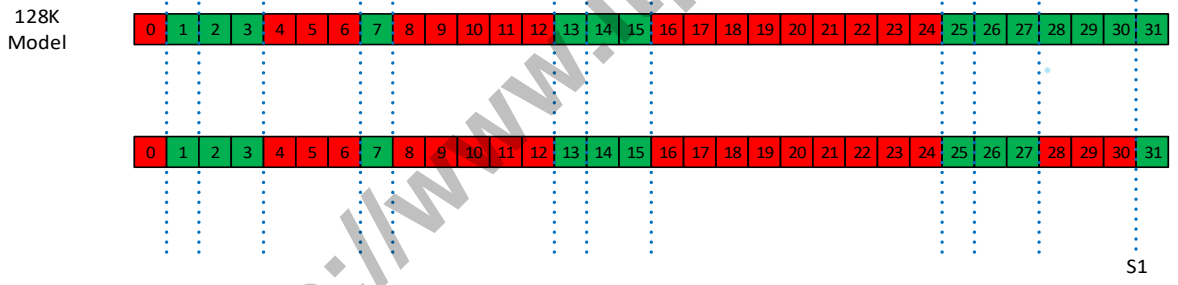
设计
Data Node 分裂与合并

架构融合
云化共建

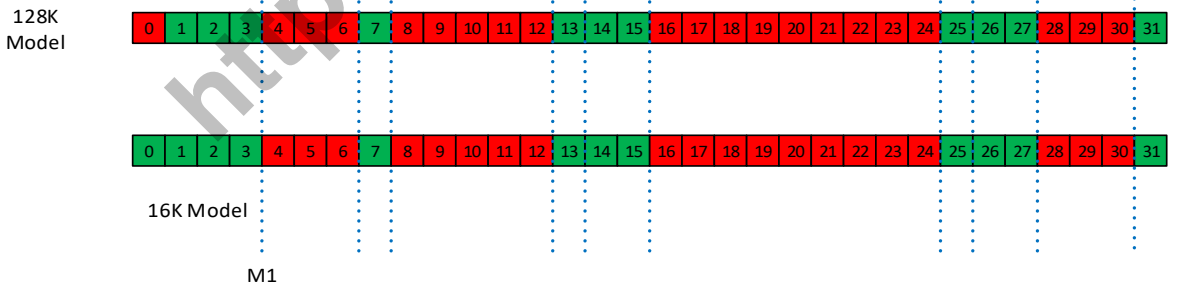
再分配3个页



再分配3个页



释放第0页



碎片化是否严重？

这个问题太难了，建议找博士研究去。

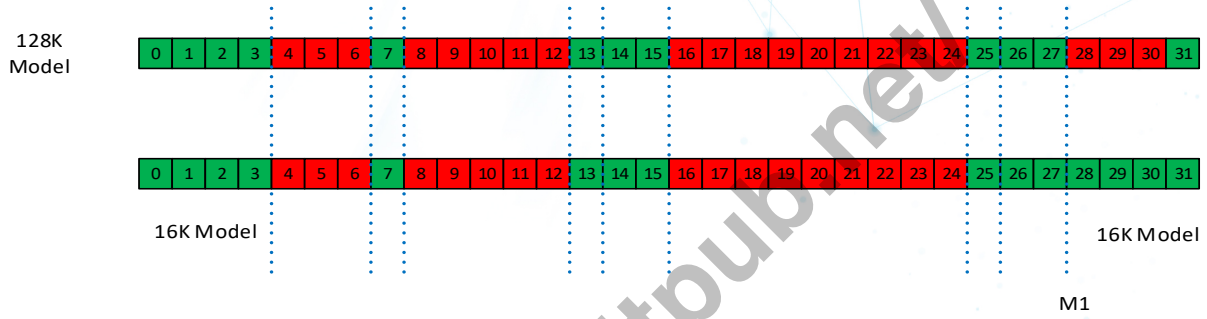
在CACHE场景下，恰好不用考虑，因为前端分片，后端文件只有若干固定的文件尺寸。

http://www.itpub.net/

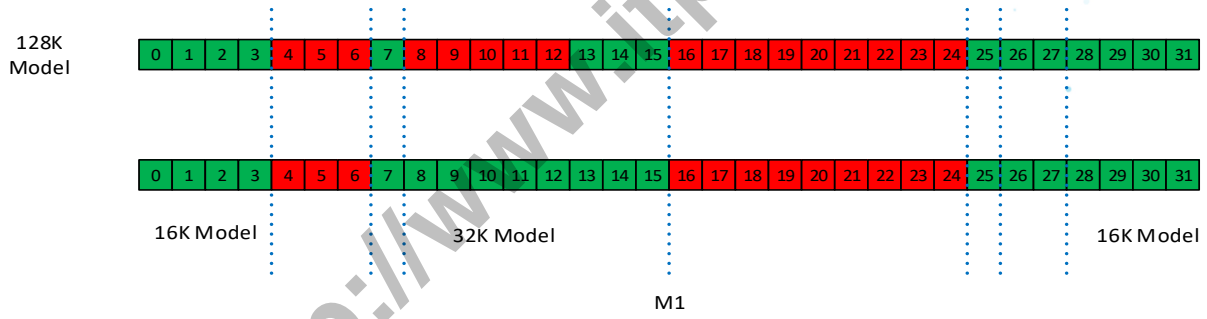
设计
Data Node 分裂与合并

架构融合
云化共建

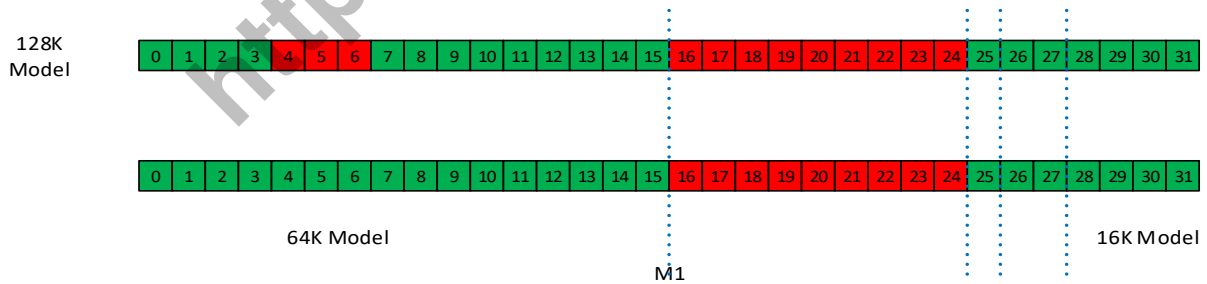
再释放第28~30页



再释放第8~12页

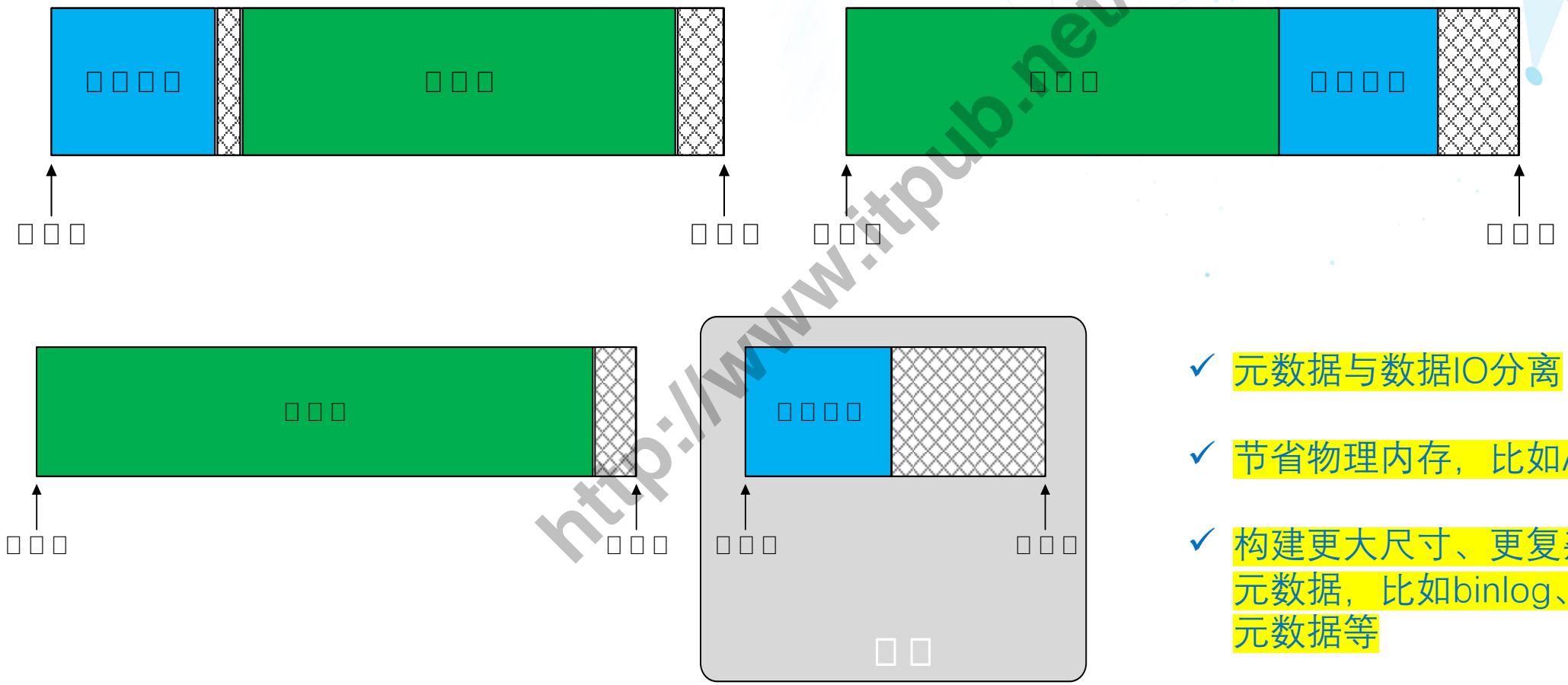


再释放第4~6页



http://www.itpub.net/

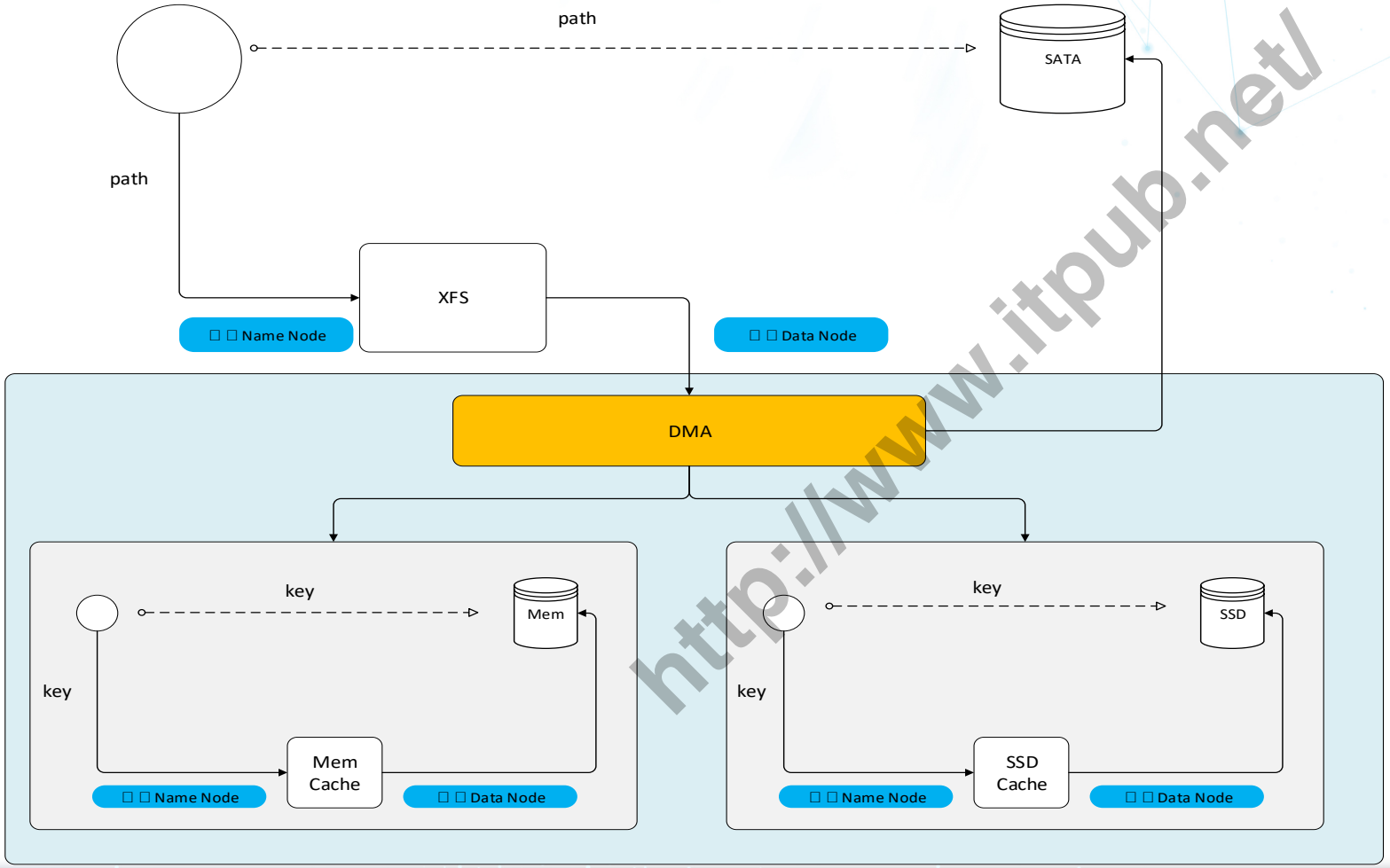
设计
元数据布局



- ✓ 元数据与数据IO分离
- ✓ 节省物理内存，比如AEP盘
- ✓ 构建更大尺寸、更复杂的元数据，比如binlog、主备元数据等

http://www.itpub.net/

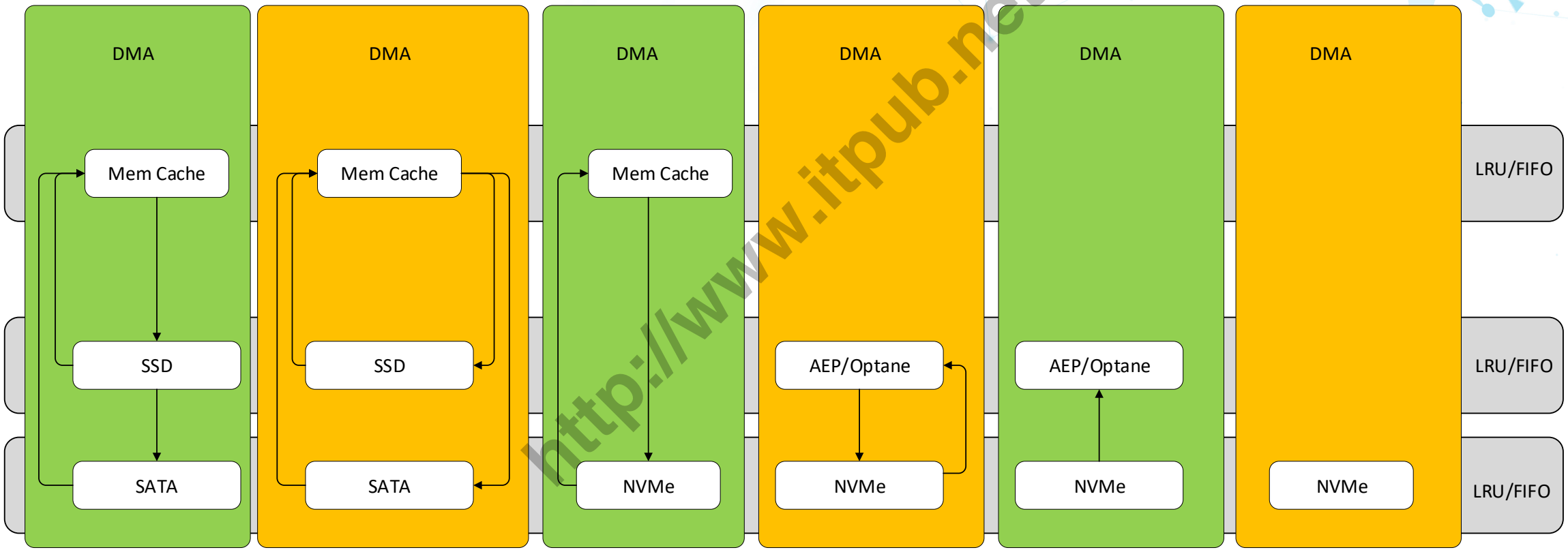
设计
分级缓存DMA



http://www.itpub.net/

- ✓ DMA将三级缓存封装，对外暴露为一块SATA盘
- ✓ 每一级缓存都是一个简化版XFS

设计
分级缓存DMA



http://www.itpub.net/

性能与应用 – 监控

架构融合 云化共建

```
# /etc/init.d/xfs stat 1 | jq
```


```
{
  "coroutine": {
    "co_idle_num": 15,
    "co_busy_num": 1,
    "co_post_num": 0,
    "co_total_num": 16
  },
  "access_stat": {
    "ac_read_counter": 41,
    "ac_read_np_succ_counter": 33
    "ac_read_np_fail_counter": 8,
    "ac_read_dn_succ_counter": 33
    "ac_read_dn_fail_counter": 0,
    "ac_read_nbytes": 7341782,
    "ac_read_cost_msec": 204,
    "ac_write_counter": 10,
    "ac_write_np_succ_counter": 1
    "ac_write_np_fail_counter": 0
    "ac_write_dn_succ_counter": 1
    "ac_write_dn_fail_counter": 0
    "ac_write_nbytes": 2097852,
    "ac_write_cost_msec": 29,
    "ac_delete_counter": 0,
    "ac_delete_nbytes": 1048926,
    "ac_update_counter": 10,
    "ac_update_succ_counter": 10,
    "ac_update_fail_counter": 0,
    "ac_update_nbytes": 2097852,
    "ac_update_cost_msec": 30,
    "ac_renew_counter": 0,
    "ac_renew_succ_counter": 0,
    "ac_renew_fail_counter": 0,
    "ac_renew_nbytes": 0,
    "ac_renew_cost_msec": 0,
    "ac_retire_counter": 1,
    "ac_retire_complete": 5,
    "ac_recycle_counter": 1292344,
    "ac_recycle_complete": 7,
    "ac_recycle_nbytes": 1048926
  },
  "xfs_model": {
    "cxfs_lru_model_switch_desc":
    "cxfs_lru_model_switch": 0,
    "cxfs_fifo_model_switch_desc":
    "cxfs_fifo_model_switch": 1,
    "cxfs_camd_overhead_switch_des
    "cxfs_camd_overhead_switch": 0
    "cxfs_camd_discard_ratio": 10,
    "c_memalign_counter": 6
  },
  "namespace": {
    "np_model": 6,
    "np_max_num": 1,
    "np_size": 536870912,
    "np_start_offset": 8650752,
    "np_end_offset": 545521664,
    "np_total_size_nbytes": 536870
    "np_item_max_num": 4194302,
    "np_item_used_num": 8
  },
  "datanode": {
    "dn_pgd_disk_choice_desc": "32
    "dn_pgd_disk_choice": 34359738
    "dn_pgb_page_choice_desc": "32
    "dn_pgb_page_choice": 32768,
    "dn_bad_page_choice_desc": "25
    "dn_bad_page_choice": 262144,
    "dn_offset": 545521664,
    "dn_fsize_nbytes": 57671680,
    "dn_disk_num": 9,
    "dn_disk_max_num": 9,
    "dn_page_max_num": 9437184,
    "dn_page_used_num": 33,
    "dn_page_used_ratio": 3.49680582682291
    "dn_used_size_nbytes": 1048926,
    "dn_assign_bitmap_desc": "000011111111
    "dn_assign_bitmap": 4095
  },
  "camd_stat": {
    "camd_disk_dispatch_hit": 2,
    "camd_disk_dispatch_miss": 43,
    "camd_rd_page_is_aligned_counter": 35,
    "camd_rd_page_not_aligned_counter": 0,
    "camd_wr_page_is_aligned_counter": 8,
    "camd_wr_page_not_aligned_counter": 0,
    "camd_rd_node_is_aligned_counter": 30,
    "camd_rd_node_not_aligned_counter": 5,
    "camd_wr_node_is_aligned_counter": 8,
    "camd_wr_node_not_aligned_counter": 0,
    "camd_mem_reused_counter": 28,
    "camd_mem_zcopy_counter": 28,
    "camd_mem_fcopy_counter": 15
  },
  "cdc_stat": {
    "cdc_pgd_disk_choice_desc": "8G-disk",
    "cdc_pgd_disk_choice": 8589934592,
    "cdc_pgb_page_choice_desc": "256K-page
    "cdc_pgb_page_choice": 262144,
    "cdc_dn_node_choice": 4294967296,
    "cdc_lru_model_switch_desc": "on",
    "cdc_lru_model_switch": 1,
    "cdc_fifo_model_switch_desc": "off",
    "cdc_used_ratio": 0.0001220703125,
    "cdc_hit_ratio": 0,
    "cdc_amd_read_speed_mps": 0,
    "cdc_amd_write_speed_mps": 0,
    "cdc_degrade_ratio": 0,
    "cdc_degrade_num": 0,
    "cdc_degrade_speed_mps": 24,
    "cdc_disk_dispatch_hit": 0,
    "cdc_disk_dispatch_miss": 34,
    "cdc_rd_page_is_aligned_counter": 24,
    "cdc_rd_page_not_aligned_counter": 0,
    "cdc_wr_page_is_aligned_counter": 10,
    "cdc_wr_page_not_aligned_counter": 0,
    "cdc_rd_node_is_aligned_counter": 24,
    "cdc_rd_node_not_aligned_counter": 0,
    "cdc_wr_node_is_aligned_counter": 10,
    "cdc_wr_node_not_aligned_counter": 0,
    "cdc_mem_reused_counter": 24,
    "cdc_mem_zcopy_counter": 24,
    "cdc_mem_fcopy_counter": 10
  },
  "caio_stat": {
    "sata_disk_fd": 13,
    "sata_disk_read_counter": 8,
    "sata_disk_read_nbytes": 2097152,
    "sata_disk_read_cost_msec": 0,
    "sata_disk_write_counter": 10,
    "sata_disk_write_nbytes": 2621440,
    "sata_disk_write_cost_msec": 1,
    "sata_disk_dispatch_hit": 3,
    "sata_disk_dispatch_miss": 18,
    "sata_rd_page_is_aligned_counter": 8,
    "sata_rd_page_not_aligned_counter": 0,
    "sata_wr_page_is_aligned_counter": 10,
```

性能与应用 – 监控

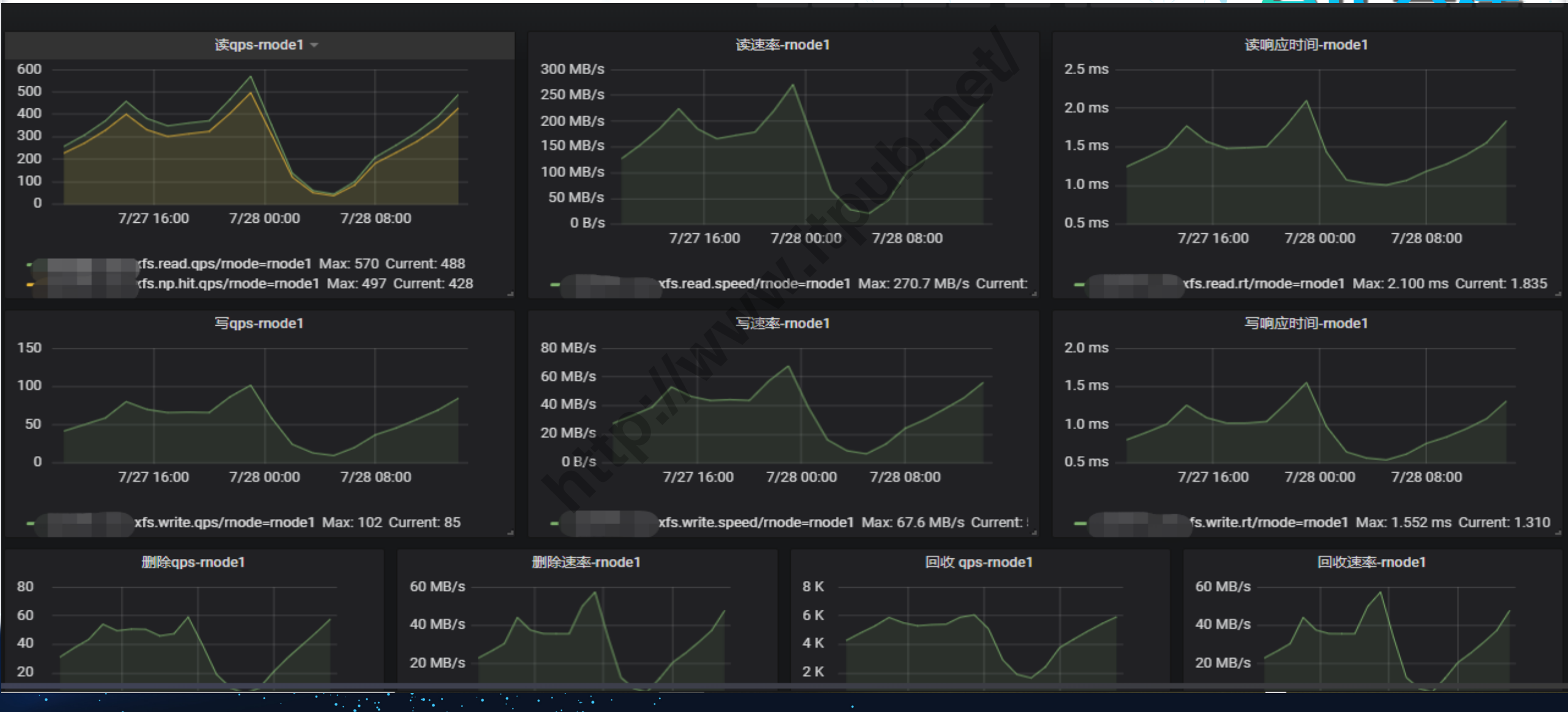
架构融合
云化共建

- 大盘概览 (12 panels)
- 基础监控 (15 panels)
- 磁盘监控 (12 panels)
- 协程监控 (4 panels)
- RAM 缓存监控 (10 panels)
- SSD缓存监控 (10 panels)
- Name Node概览 (9 panels)
- Data Node 监控 (14 panels)
- 连接监控 (1 panel)
- 错误情况 (1 panel)

http://www.itpub.net/

性能与应用 – 监控

架构融合
云化共建



http://www.itpub.net/

开源

源码： https://github.com/chaoyongzhou/XCACHE

文档：https://github.com/chaoyongzhou/Knowledge-Sharing

http://www.itpub.net/

Q&A

架构融合
云化共建

http://www.itpub.net/

Thanks

http://www.itpub.net/