A Highly Parallel Production File System
by the PSC Advanced Systems Group
Design Motivation

- To optimize science utilization of the machine
  - Maximize “compute time / wall time” ratio
  - Maximize aggregate “TB / sec / $”
    - Without compromising R.A.S.
    - Scalable up to large I/O clusters
- To ensure fault tolerance
  - Application recovery following HW/SW failures
    - Checkpoint I/O
- In exchange for:
  - Temporarily suspending some POSIX semantics
    - while retaining identical calling sequences (transparency)
  - Delayed availability for reading
Design Features

- Client-side parity calculation
  - Leverages compute node CPU over I/O node CPU
  - In a transparent intercept library

- Sequential I/O
  - Aggregate many small RPCs into fewer large RPCs
  - Disk block allocator gets “next available”
    - Both effective “latency hiding” strategies

- Asynchronous fragment reconstruction
  - Reassembly MD is carried with data blocks
Software Components

- **Client Library (on compute node)**
  - Transparent intercept
  - Parity calculation
- **LNET Router (on SIO node)**
  - Route messages from HSN to external IB NW
- **ZEST daemon (on Zest I/O node)**
  - Communications Threads
    - LNET to memory
  - IO Thread
    - All I/O to/from each “internal” disk
  - Syncer Thread
    - Staging data out to external PFS
  - Parity Thread
    - Regenerate lost data buffers
“Scalable Unit” Hardware

- No single point of failure
Benchmarking Tests

- **Standard Benchmarking Applications**
  - *No* source modifications required (transparent intercept)
  - IOR – used for the following scaling plots
  - SPIOBENCH
  - Bonnie++
  - FIO – PSC-developed, highly parallel and configurable suite

- **Zest Component Tests (sustained)**
  - I/O Threads – 1.1 GB/sec (91% of 1.2 GB/sec raw spindle)
  - RPC Threads – 1.1 GB/sec (27% of 4 GB/sec over IB)
Benchmarking Goals

1. Measure client scaling
2. Measure max write bandwidth
3. Compare Zest to Lustre

All of the following measurements:
- IOR client application
- 1 single server (identical hardware)
Zest Performance (Linux / SDP)

Note:
- Good client scaling
- 750 MB/sec
Zest Performance (XT3 / IPoIB)

Note:
- Excellent client scaling to 1K
- Highly consistent per-write performance
- IPoIB limited
- 150 MB/sec
Lustre Performance (RAID0 / o2ib)

Note:
- Just for illustration (max BW)
- 16 OSTs
- No redundancy
- 430 MB/sec

![Graph showing Lustre Performance](image)

- Max
- Avg
- Min

© Pittsburgh Supercomputing Center
Lustre Performance (RAID5 / o2ib)

Note:
- What you get w/o HW RAID
- 1 OST: Linux software RAID
- 100 MB/sec
Storage System Write Efficiency

“Efficiency” = Aggregate Application BW / Aggregate Raw Spindle Speed

Because per-drive cost is a substantial portion of a storage subsystem

<table>
<thead>
<tr>
<th>% of Raw Spindle Bandwidth</th>
<th>SW RAID5</th>
<th>HW RAID5</th>
<th>SW RAID0</th>
<th>ZEST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8%</td>
<td>26%</td>
<td>X</td>
<td>63%</td>
</tr>
</tbody>
</table>

Zest: Highest Server Efficiency
Storage System Performance/Cost

Aggregate Application BW / Cost of Storage Subsystem

MB/sec/$10K

600

400

200

0

SW RAID5

HW RAID5

SW RAID0

ZEST

501

287

153

67

Zest: Highest TB/sec/$ (no per-server overhead)
The Zest Dashboard

- Server “Efficiency” (aggregate)
  - >90% peak
  - 80% sustained

- All Disk BW
  - Sustained near max

- Ingest Buffers
  - Remained empty throughout benchmarks
zestiondctl

A shell tool that provides **interactive reporting and control** of Zest primitives, including:

- **Controls**: logging, syncing, disk online/offline
- **Performance**: disk I/O, thread activity, communication rates
- **Resources**: memory structures, inodes, multilock lists, requests, parity groups

NAME

zestiondctl - zestion daemon runtime control

SYNOPSIS


DESCRIPTION

The zestiondctl utility examines and manipulates zestiond(8) behavior.
Zest Configuration (Linux / SDP)
Zest Configuration (XT3 / IBolB)
Lustre Configuration (RAID0 / o2ib)
Lustre Configuration (RAID5 / o2ib)