Flexible Network Attached Storage using Remote DMA

Jørgen Sværke Hansen  
<jorgen.hansen@inrialpes.fr>  
*SIRAC Project*
*INRIA Rhône-Alpes*

Outline

- Motivation
- The Proboscis Prototype for Scalable Coherent Interface Networks
- Measurements
- Conclusions
**Motivation**

I/O capacity of clusters using centralized storage servers scales poorly with number of nodes

---

**Storage I/O Alternatives**

- Distributed/replicated file servers
- Storage area networks:
  - Special purpose networks (InfiniBand may change that)
  - Shared, dumb disks
- Network attached storage devices:
  - Hard drives with processor and network card
- PC attached storage devices:
  - COTS components
  - Customizable
Thinly Spread PC Attached Disks

We propose:
• Thinly spread PC attached disks
where:
• Each node in the cluster shares locally attached
disk(s) with the other nodes in the cluster

Result:
• Both I/O bandwidth and processing capacity
  scales with number of nodes, i.e., nodes double as
  processing and SAN nodes.

Thinly Spread PC Attached Disks: Issues

• Reliability worse than for SANs but may be
  increased through:
  – Redundancy
  – Special support for failures of short duration
• Node overhead is reduced by:
  – Zero-copy networking
  – Surplus storage I/O capacity
• Administration:
  – Can get ugly
The Proboscis Framework

Framework for remote disk access construction and administration:

- Modular
- Extensible

Makes I/O data paths explicit to facilitate:

- Reconfiguration at runtime
- Disk access scheduling
- Monitoring

Proboscis Prototype Implementation

- Kernel level implementation for Linux 2.4
- Uses Scalable Coherent Interface (SCI)
- Currently supports simple remote disk access
- Used for examining disk node overhead of three different buffer transfer strategies
Proboscis Prototype: Buffer Transfers for RDMA networks

Scalable Coherent Interface: Address Mapping
Direct Access Approach

- Transfer data directly between disk and client buffers

Direct Access Approach (2)

- Implementation:
  - Map remote buffers into I/O address space
  - Give disk controller I/O space addresses
  - Unmap remote buffers on I/O completion

- Pros:
  - No memory or memory bus load

- Cons:
  - Mappings are a scarce resource
  - Error handling during writes
Temporary Buffer Approach

- Temporary buffers on disk node

![Diagram of client and disk node with temporary buffers and I/O buses](image)

Temporary Buffer Approach (2)

- Implementation:
  - Read to disk node local buffer and copy to client on completion
  - Two strategies for copying to disk node on write:
    - Client node pushes data to disk node
    - Disk node pulls data from client

- Pros:
  - Easier failure handling

- Cons:
  - Loads memory and memory bus
Performance Measurements

Answer the questions:

• Can remote disks compete with local ones?
  – local versus remote performance using:
    • Bonnie benchmark on ext2 file system
    • Raw device access using dd

• What does it cost for a node to host a disk?
  – Measure application slowdown on disk nodes caused by raw device access

Test Node Configuration

• Base Configuration:
  – 933 MHz Pentium III
  – 1 GB memory
  – 64 bi, 66 MHz PCI bus

• Disk Configuration (Maxtor Atlas 10K II):
  – 40 MB/s maximum sustained transfer rate
  – 4.7 millisecond average seek time

• Network Configuration (Dolphin SCI D330 Adapter):
  – Small message latency: 28 microseconds
  – Maximum remote bandwidth:
    • Read: 44 MB/s (DMA), 5 MB/s (PIO)
    • Write: 240 MB/s (DMA), 204 MB/s (PIO)
Device Performance (dd)

- Direct access write disappoints
- Otherwise, remote equals local performance

Application Slowdown

- Simple benchmark:
  - Loop with integer calculations using processor registers
- Scientific Calculation – bt from NAS:
  - Heavy memory usage (298MB)
  - Floating point calculation
- Performed concurrently with device copy operation (dd read and write)
Application Slowdown Results

- Push strategy best for writes (7% - 16%)
- Direct access best for reads (13% - 23%)

Conclusion

- Remote disk performance is comparable to local disks (except for direct access write)
- Slowdown of complex application not too bad under maximum load:
  - Write 16% (push strategy)
  - Read 23% (direct access strategy)
  with disk performance close to unloaded case
- Cluster nodes can double as processing and I/O nodes (but with load restriction facilities)
SCI Clusters: Advantages

- High performance (2-5 microseconds latency, bandwidth up to 84-240 MB/s)
- Little operating system overhead
- Hosting processor is not interrupted on remote load or store
- Reliability in hardware
- Flexible

SCI Clusters: Hardware

- Network interface mapped by processor
- Operating system sets up memory mapping
- Communication through user-level load/store or DMA
- No cache consistency
Ext2 R/W Performance (bonnie)

- Remote equals local disk seek performance