ISTORE: Hardware Overview and Software Challenges

http://www.cs.berkeley.edu/~yelick/294

The ISTORE Gang
James Beck, Aaron Brown, Daniel Hettena, Jon Kuroda, David Oppenheimer, David Patterson, Alex Stamos, Randi Thomas, Noah Treuhaft, Kathy Yelick

Related Project: IRAM
Microprocessor & DRAM on a chip:
- Better latency 5-10X,
- Better bandwidth 50-100X
- Improve energy 2X-4X
- Smaller board area/volume

IRAM focus is ~1-chip systems (PDAS, etc.)

A glimpse into the future?
• Add system-on-a-chip to disk: computer, memory, redundant network interfaces without significantly increasing size of disk
• ISTORE HW in 5-7 years:
  - Building block: MicroDrive integrated with IRAM
  - If low power, 10,000 nodes fit into one rack!
• Design for O(10,000) nodes

The real scalability problems: AME
• Availability
  - Systems should continue to meet quality of service goals despite hardware and software failures
• Maintainability
  - Systems should require only minimal ongoing human administration, regardless of scale or complexity: Today, cost of maintenance = 10X cost of purchase
• Evolutionary Growth
  - Systems should evolve gracefully in terms of performance, maintainability, and availability as they are grown/upgraded/expanded
• These are problems at today’s scales, and will only get worse as systems grow

Principles for achieving AME (1)
• No single points of failure
• Redundancy everywhere
• Introspection
  - Reactive techniques to detect and adapt to failures, workload variations, and system evolution
  - Proactive techniques to anticipate and avert problems before they happen
Principles for achieving AME (2)

- Performance robustness is more important than peak performance
  - performance under real-world, imperfect scenarios, including failures
- Performance can be sacrificed for improvements in AME
  - resources should be dedicated to AME
  - compare: biological systems spend > 50% of resources on maintenance
  - scale system to meet performance needs

The Applications: Big Data

- Two key application domains:
  - storage: public, private, and institutional data
  - search: building static indexes, dynamic discovery
- Applications demand large storage, some computation

Today's Agenda

- ISTORE History & Vision
- ISTORE-1 Hardware
- Proposed ISTORE-2 Hardware
- Software Techniques
- Administrivia

HW (1): SON

- Cluster of Storage Oriented Nodes (SON)
- Distribute processing with storage
  - Most storage servers limited by speed of CPUs
  - Amortize sheet metal, power, cooling, network for disk to add processor, memory, and a real network?
  - Embedded processors 2/3 perf, 1/10 cost, power?
  - Switches also growing with Moore's Law
- Advantages of cluster organization
  - Truly scalable architecture
  - Architecture that tolerates partial failure
  - Automatic hardware redundancy

HW(2+3): Instruments+DP

- Heavily instrumented hardware
  - sensors for temperature, vibration, humidity, power, intrusion
  - helps detect environmental problems before they can affect system integrity
- Independent diagnostic processor on each node
  - provides remote control of power, remote console access to the node, selection of node boot code
  - collects, stores, processes environmental data for abnormalities
  - non-volatile "flight recorder" functionality
  - all diagnostic processors connected via independent diagnostic network

HW(4): Subset Isolation

- On-demand network partitioning/isolation
  - Internet applications must remain available despite failures of components, therefore can isolate a subset for preventative maintenance
  - Allows testing, repair of online system
  - Managed by diagnostic processor and network switch via diagnostic network
HW(5): Fault Injection

- Built-in fault injection capabilities
  - Power control to individual node components
  - Can inject glitches into I/O and memory busses
  - Managed by diagnostic processor
  - Used for hardware introspection
  - Controlled testing of error-recovery mechanisms
  - Important for AME benchmarking

ISTORE-1 hardware platform

- 80-node x86-based cluster, 1.4TB storage
  - Cluster nodes are plug-and-play, intelligent, network-attached storage “bricks”
  - A single field-replaceable unit to simplify maintenance
  - Each node is a full x86 PC w/256MB DRAM, 18GB disk
  - More CPU than NAS; fewer disks/node than cluster

ISTORE Chassis

- 80 nodes, 8 per tray
- 2 levels of switches
- 20 100 Mbit/s
- 2 1 Gbit/s
- DUAL UART
- Environment monitoring: UPS, fans, heat and vibration sensors...

ISTORE Brick Block Diagram

- Mobile Pentium II Module
- DRAM 256 MB
- PCI
- Ethernets 4x100 Mbit/s
- Sensors for heat and vibration
- Control over power to individual nodes

ISTORE Brick Block Diagram

- Mobile Pentium II Module
- DRAM 256 MB
- PCI
- Ethernets 4x100 Mbit/s
- Sensors for heat and vibration
- Control over power to individual nodes

ISTORE-1 System Layout

- UPSs: “used”
- PE5200s: PowerEngines
- 100Mb switches
- PE1000s: PowerEngines
- 1Gb switches
- “Used” switched

ISTORE-1 Status

- All boards fabricated
- Boots OS
- Diagnostic Processor Interface SW complete
- Minor boards: backplane, patch panel, power supply in design/layout/fab
- Finish 64-80 node system: October

Onward to ISTORE-2?

- IStore-1: Present UCB Project
- IStore-2: Possible Joint Research
  - Prototype between UCB and IBM
  - IBM: Winfried Wilcke, Dick Booth...
  - ~2000 nodes
  - Split between UCB, IBM and others
  - Hardware similar to ISTORE-1
  - Focus on real applications and management software
HW(6): RAIN

- Switches for ISTORE-1 substantial fraction of space, power, cost, and just 80 nodes!
- Redundant Array of Inexpensive Disks (RAID):
  - replace large, expensive disks by many small, inexpensive disks, saving volume, power, cost
- Redundant Array of Inexpensive Network switches (RAIN) for ISTORE-2:
  - replace large, expensive switches by many small, inexpensive switches, saving volume, power, cost?
  - E.g., replace 2 16-port 1-Gbit switches by fat tree of 8 8-port switches, or 24 4-port switches?

Redundant Networking

- ISTORE-1 has redundant networks
  - Each node has 4 100Mb connections
  - 2 Gb switches at the top
  - Can-bus network for DPs
- Using the 4 ethernets
  - Stripping for performance (on ISTORE-0)
  - Adaptation for single switch or interface failures

Underlying Beliefs...

- Commodity components are quickly winning the server wars
  - Gigabit Ethernet will win everything
  - x86 Processors
  - Linux OS will prosper
- Large servers (100-10k nodes) will be quite common - and most are storage centric
- What matters most:
  - Ease of management, density of nodes and seamless geographical interconnect

IStore-2 Footprint

- 16 Storage (19") Racks
  - 64 Storage bricks/rack
  - 8 type 1 storage bricks/drawer
  - 8 storage drawers/rack
  - Ethernet switches in rack
- 8 Global Ethernet Switch (19") Racks
- Requires 600 sq ft lab!!
  - Soda II or better packaging and integration (2.5 disks...) necessary

Today's Agenda

- ISTORE History & Vision
- ISTORE-1 Hardware
- Proposed ISTORE-2 Hardware
  - Software Techniques
  - Administrivia

Software Techniques

- Reactive introspection
- Proactive introspection
- Semantic redundancy
- Self-Scrubbing data structures
- Language for system administration
- Load adaptation for faults
- Benchmarking
- Your ideas go here!
SW1: Reactive Introspection
• Automatic fault detection
• "Mining" any available system data
  - Disk logs (aka Tertiary Disk experience)
  - Disk "firings" are predictable
  - Load, memory usage, etc.
  - Network traffic
  - Programmer counter?
    - Untested code is always buggy
    - Exception code is often untested

SW2: Proactive Introspection
• Test recovery system
  - Are replicas, backups, etc. consistent?
  - Does recovery code work in current setting?
• Isolate subset of system
• Inject faults using hardware or software techniques
• Automate this process and schedule for regular runs

SW3: Semantic Redundancy
• Replication is the simplest form of redundancy
• How can other forms of redundant information help reliability?
  - Multiple paths in linked data structures
  - Coding-based replication
  - Checkpointing of computations using application-specific knowledge

SW4: Self-Scrubbing Data Structures
• Distributed data structures
  - Often implicit: based on communication patterns in application
    - Trees, hash tables, etc.
  - Complex invariants may exist across the structure
    - Maintain n copies of every data element
    - Balance size of each sub-container within p%
  - Check invariants at runtime (and repair?)

SW5: Safer System Management
• ~40% of systems failures are due to administrative errors
• The most common tools used by administrators include:
  - Perl, csh, awk, sed
  - Lack all of the features of good language design
    - Little or no redundancy, e.g., untyped
    - Syntax based on minimizing keystrokes, error prone
  - Little or no support for procedural or data abstraction
  - No static checking of program correctness, e.g., compiler
  - Systems like Cfengine provide limited support
• Apply language techniques to age-old problem?

SW6: Load Adaptation for Faults
• ISTORE looks homogeneous now, but
  - Didn't get enough disks (only 70)
  - New model, IDE vs. SCSI?
  - Large systems change over time
    - Grow based on need
    - Decay (components replaced) based on faults
  - Even physically homogenous systems may have performance heterogeneity
    - Process or network load, sharing machine, ...
  - Hardware faults + redundancy can lead to performance faults
SW7: I/O Performance Heterogeneity

- Delivered disk performance is subject to many factors: specs, layout, load
- Static I/O load balancing turns local slowdown into global slowdown
- Two solutions
  - Virtual Streams (River system)
  - Graduated Declustering implementation and performance

Context: Bulk I/O

- Applications in decision support, data mining, simulation (oil reservoir) use
  - Batch-style, bulk-synchronous computations
  - Large reads (e.g., sorting) and writes (e.g., checkpoints)

Virtual Streams
Dynamic load balancing for I/O Replicas of data serve as second sources
- Maintain a notion of each process's progress
- Arbitrate use of disks to ensure equal progress
- The right behavior, but what mechanism?

Performance Heterogeneity
- System performance limited by the weakest link
- Virtual Streams: dynamically off-load I/O work from slower disks to faster ones

Read Performance: Multiple Slow Disks

- Ideal
- Virtual Streams
- Static

Record-breaking performance is not the common case

- NOW-Sort experience: heterogeneity is the norm
  - disks: inner vs. outer track (50%), fragmentation
  - processors: load (1.5-5x) and heat
  - E.g., perturb just 1 of 8 nodes and...
HW/SW: Benchmarking

- Benchmarking
  - One reason for 100X processor performance was ability to measure (vs. debate) which is better
    - e.g., Which most important to improve: clock rate, clocks per instruction, or instructions executed?
  - Need AME benchmarks
    - what gets measured gets done
    - benchmarks shape a field
    - quantification brings rigor

Availability benchmarks

- Goal: quantify variation in QoS metrics as events occur that affect system availability
- Leverage existing performance benchmarks
  - to generate fair workloads
  - to measure & trace quality of service metrics
- Use fault injection to compromise system
  - hardware faults (disk, memory, network, power)
  - software faults (corrupt input, driver error returns)
  - maintenance events (repairs, SW/HW upgrades)
- Examine single-fault and multi-fault workloads
  - the availability analogues of performance micro- and macro-benchmarks

Methodology for reporting

- Results are most accessible graphically
  - plot change in QoS metrics over time
  - compare to "normal" behavior?
- 99% confidence intervals calculated from no-fault runs

Example single-fault result

- Compares Linux and Solaris reconstruction
  - Linux: minimal performance impact but longer window of vulnerability to second fault
  - Solaris: large perf. impact but restores redundancy fast

Summary

- ISTORE: hardware for availability, reliability, maintainability
- Needs software
  - To use hardware ft features
  - For problems not solvable by hw
- Other ideas?

Administrivia

- Homework 1 due next Tuesday (9/5)
  - Anatomy of a failure
- Read Grapevine and Porcupine papers for 9/5
  - Porcupine link fixed
  - Grapevine paper copies available
- Read wireless (Baker) paper for 9/7
  - Short discussion next Thursday 9/7 (4:30-5:00 only)