Solid State Technology – Where Does it Fit for Customer Applications?

Rob Peglar
Vice President, Technology
Xiotech

A Blast from the Past – Happy 25th!
August 12th, 1981

IBM Model 5150 Specifications

Processor	Intel 8088
Speed	4.77 MHz
RAM	16KB
Storage	Cassette Tape, optionally 5.25" 160KB floppy drives
Expansion	5 expansion slots
Bus	Industry Standard Architecture (ISA)
Video	Initially CGA (320x200x16 color, 640x200x2 color) or monochrome (80x25 text only))
I/O	Parallel, Serial
OS	Microsoft Basic 1 (ROM)
Killer App	VisiCalc

Fast Forward – to 2006

- Today, we have CPUs which are 1,000x
- Today, we have RAM which is 1,000,000x
- Today, we have storage which is 3,000,000x
- So what's the problem?

The Problem WAS – AND IS – I/O

- In a perfect world, I/O would not be necessary
  - 1st level store would hold everything, forever
- Access Density – IOPS/GB
  - Getting WORSE over time
- Example:
  - IBM Model 5150 – 625 KB/s, 8.33 ms, 3,600 RPM
    - IOPS/GB = 20 / .001 = 20,000
  - Today – 85,000 KB/s, 2.9 ms, 15,000 RPM
    - IOPS/GB = 200 / 300 = 0.667

The Advent of Solid State Disk

- Technology Choices – Boiled Down to Two
- NAND Flash
  - Slow, cheap, dense, non-volatile
  - JFFS2
  - ONFI – Open NAND Flash Interface - finally
- DRAM
  - Very fast, dense, not cheap, volatile
  - No internal file system
  - Is it cache, or is it disk?
- NOR Flash – odd man out

Business Criteria

- Flash as Magnetic Disk Replacement
  - Write cycles, cost/GB, media lifetime, TCO
- DRAM as Cache Replacement
  - Cost/GB, TCO, expandability/flexibility
- Application Workloads
  - Transactional (random) versus Batch (sequential)
  - Database versus Files
  - Structured versus Unstructured

Application Requirements

- Applications don’t want disks
  - They want space (more is better)
- Applications don’t want IOPS
  - They want time (less is better)
- Applications do I/O because they have to
  - But they don’t really want to
- The problem is not applications, it’s application programmers

Real-World Application Workloads

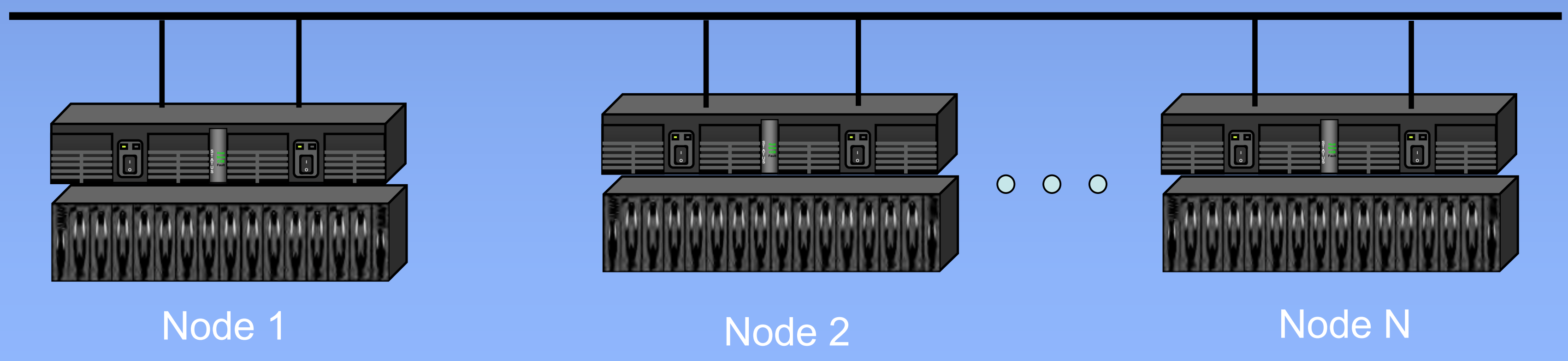
- Unstructured data
  - Unstructured data is a poor fit for SSD
  - Exception – small, non-growing tagged files
  - OS images – boot-from-flash, page-to-DRAM
- Structured data
  - Structured data is an excellent fit for SSD
  - Exception – large, growing table spaces
- Databases have key elements
  - Commit files – logs, redo, undo

How should I Design my SAN for SSD?

- SSD introduces a new complexity into SAN
  - Or does it?
- SSD should be treated <u>exactly</u> like magnetic
- External SSD == bad
  - Captive to server, doesn't scale
- SAN-based SSD == good
  - Not captive to server, scales
  - Add more SSD drives as demand grows, online
  - Virtualized & Clustered

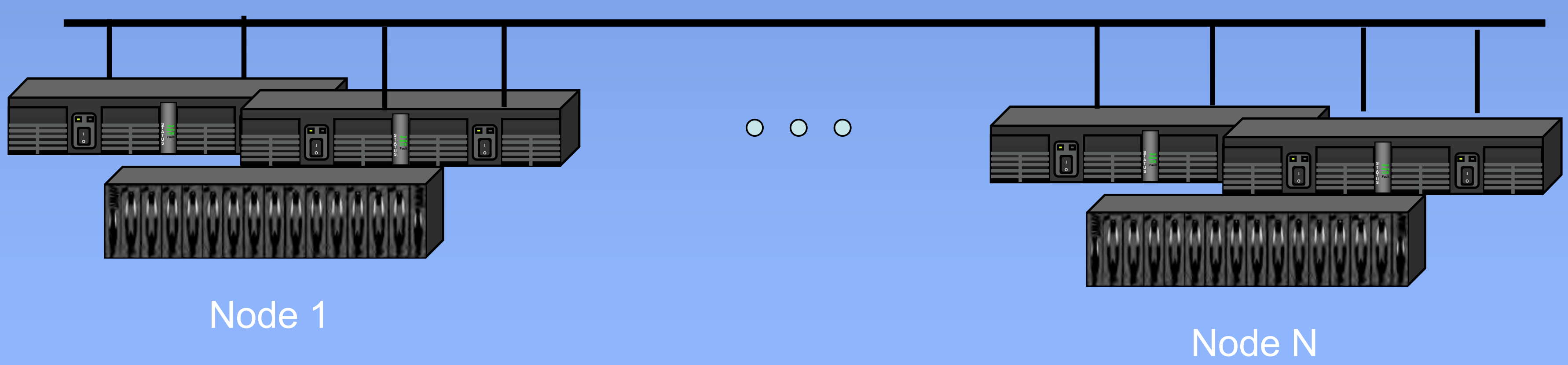
Three Types of Clustered Storage

- Type 1 - Single-access captive storage
  - Controller node integrated with captive disks
- Clustering is via multiple instances of nodes
  - Flaw - Inter-node latency to reach disks
  - Flaw – Sparing across nodes – SPOF controller



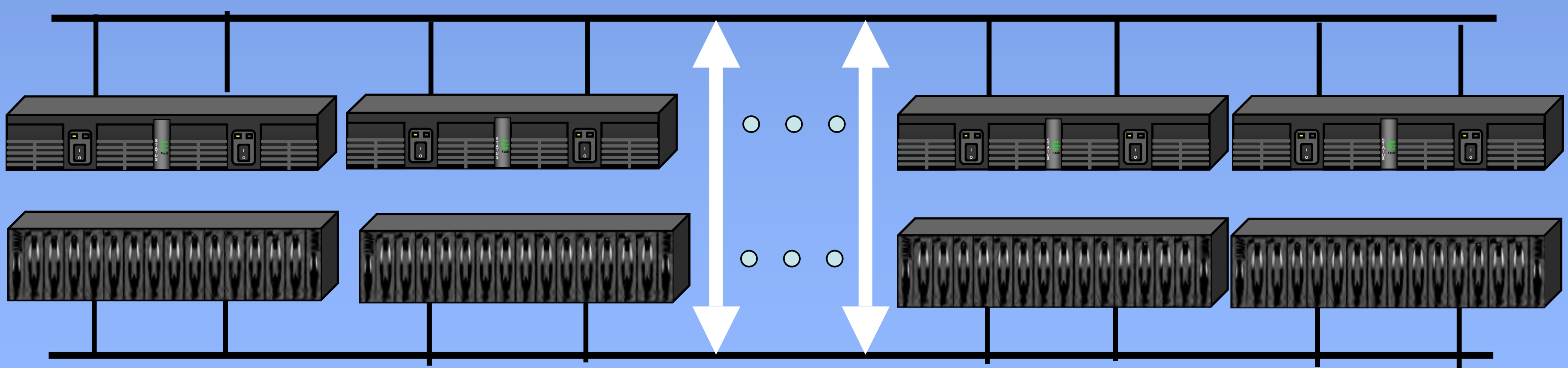
Three Types of Clustered Storage

- Type 2 - Dual-access captive storage
  - Pairs of controller nodes integrated with captive disks
- Clustering is via multiple instances of dual nodes
  - Flaw - Inter-node latency to reach disks
  - Flaw – Sparing across nodes



Three Types of Clustered Storage

- Type 3 - Multi-access non-captive storage
  - N controller nodes networked with N storage nodes
- Clustering is optimal – controllers & storage
  - Any-to-any communication to disks
  - Sparing at the storage node level



THANK YOU

Q&A