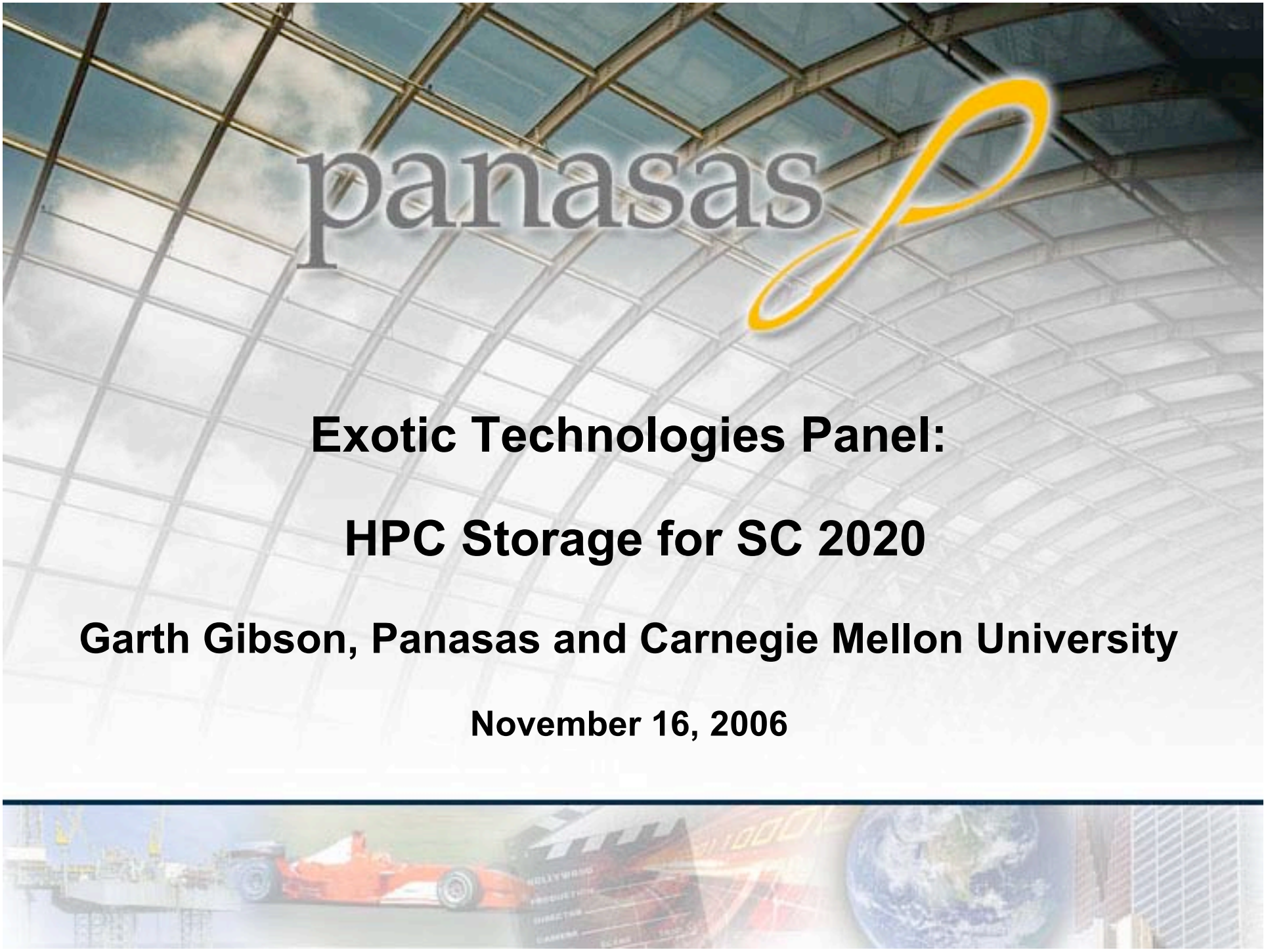

Exotic Technologies Panel:

HPC Storage for SC 2020

Garth Gibson, Panasas and Carnegie Mellon University

November 16, 2006

2020 Simple Projections

- Disks -- 40%/yr density -- 20%/yr data rate -- 5%/yr seek rate
  - 3.5” disks at 60 TB, 2.5” disks at 10 TB; 1” disks at 500 GB
    - BW will be 300 - 600 MB/s peak; Seeks will 1/3 today, a few msecs
  - SC 2020 machines will use 2.5” disks with 20 GB+ of memory each
    - We will be talking about switching to 1” disks, but the cost will seem prohibitive
- Networks
  - 1 Tbit/sec will be common and we will be talking about going to 10 Tbit/sec
  - Current differences between ether, IB etc are long gone, so call most of it ethernet, and the leading edge version will have another name
- SC 2020 peak systems -- top500 trends will continue
  - 5 ExaFLOPS, 2 ExaBytes memory, 100 ExaBytes storage, 5 PB/s network

File System and Storage Architecture

- Disks will be computers
  - Compute in disk for search, aggregate functions programmable by apps
    - Search accelerators will be common and well understood
  - Programming environment will be familiar, probably a virtual machine
  - Lots of memory for checkpoint bursts, data and metadata indices
  - Enough NVRAM for key metadata to not make disk seeks
  - Object semantics as a base, but much richer in terms of “method operators”
- File system will change much less
  - Direct transfer from user level to disk (to real disk, not PC called controller)
  - Significant addiction to POSIX will continue after small changes in POSIX
  - Cache management will be more important
    - Application transactions will allow program to manage cache residency better

RAID and Redundancy

- Much heavier variance in redundancy encoding
  - Massive replication of most critical and slowly changing data
  - Redundant and compressed representation of rarely used data
  - More tiers and types of memory hierarchy
- Much smaller collections in same failure domain
  - Media failures will not but at risk loss of 10X disk size
  - File systems will better understand data damaged by rare failures
    - Recovery will be at information level (files, databases, app datasets), not volume block level
    - E.g. per-file RAID as in object storage and peer-2-peer file systems
    - E.g. “chunk FS” in Linux which collect files in containers to minimize crossing links
- Recovery speed will scale with total resources
  - not one RAID card speed applied to entire disk
    - E.G. declustering of per-file RAID relationships

File Systems Architecture Con't

- Of course it will be called NFS, probably 3rd generation Parallel NFS
  - Leading edge will be faster, more parallel, more precise and not called NFS
- File system abstraction will be augmented (climb the stack)
  - File tree will still be around, but won't be the real representation of metadata
  - Many more attributes will be deduced by system, assigned by apps
  - Many more indices
    - Age dependent, App dependent, Type dependent,
    - Context dependent (dependencies of creation/information flow, coincident use)
  - File system metadata changes will have to be transactional
    - Transactional semantics offered to system users
    - Blob databases, yes, but coded fresh with scale in mind (Google GFS+BigTable)
  - Application specific interfaces, indices, access methods attached to data
    - Like schemas with keys, but with application specific semantics

Personal Perspective

- Major projects take a decade to be mature
  - RAID mid 80s to mid 90s
  - Object Storage late 90s to late 00s
  - Parallel NFS mid 00s to mid 10s
- So the question is, what is the major decade thrust after Parallel NFS?
- System will be constantly in partial failure states, recovering integrity in some areas of data space while transforming other areas at full speed
- This means aggressively increasing system complexity
- Best guess now is full formal verification of Distributed Storage protocols

Formal Verification of Storage at Scale

- Problem is complexity in high speed AND high integrity AND Scale
  - Balanced system speed to double annually, but disk speedup is 20%/yr
  - Increasing numbers of components
  - increasing asynchronous operation
  - Increasing reordering, coalescing in memory
  - No relaxation of data protection -- no checkpoint restart for storage crashes
- Model checking/formal verification
  - Follow example from circuit design
  - Enumerate all possible states and verify correct transitions in and out
  - “Manage” exponential runtime with symmetry in state space & fast computers
  - Applied effectively to circuits and wire protocols
  - Just starting to apply to file system transforms of storage

Starting in on Verification

- Early projects starting now (IBM Haifa, Stanford, Wisconsin)
  - Tiny file systems (a few MBs)
  - Transform with a few operations on file system
  - Trace all choice points in transformation
    - Capture state space for each choice (virtual machine like)
  - “Crash” code on all paths, with all orders of updating disk
  - Run static checker (FSCK) in all crashed states
    - Evaluate correctness and completeness of on disk state
- Very limited
  - Small systems, slow runtime
  - No testing of in memory state; no formal specification of full system states
  - No testing of distributed state of coherent caches in clients, clustered servers
  - No testing of adequate performance under workloads

2020 Summary

- Huge disks are full computers executing parallel search/aggregate
  - 2.5”, 10TB, 20+ GB memory, networks concentrating to 1 Tbit/sec
- More memory heirarchies, more use of memory for metadata
- More sophisticated replication, tiering, “grouping” of data sets
- Major effort to fully specify state space of transforms on abstract storage
  - Prove protocols correct
  - Prove implementations correct

Accelerating Time to Results With Clustered Storage

Garth Gibson
garth@panasas.com