SPC-3 Proposal for New Persistent Reservation Type

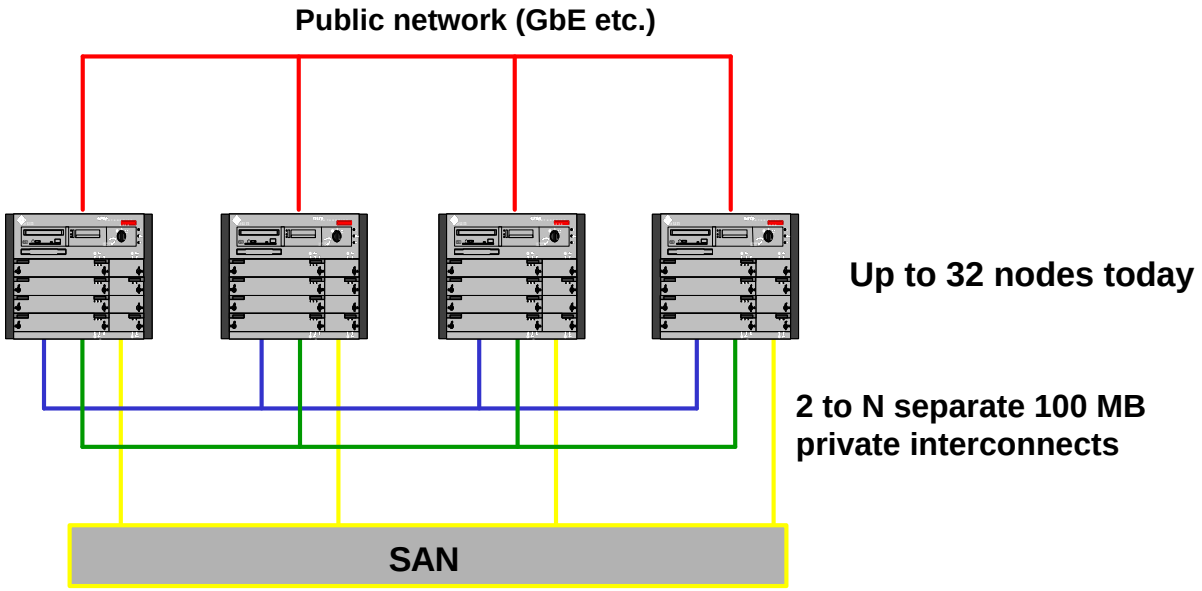
Roger Cummings

September 12, 2001

Background

- Ongoing work resulting from the rejected SPC-2 PR Comment
- VERITAS still believes that current registrant-only persistent reservations definitions has significant issues in its cluster architecture
  - Have tried to implement a “Preempt” scheme since last meeting…..
    - But results in a bunch of race conditions that are difficult to handle…..
    - And even more difficult to PROVE are handled correctly

VERITAS Cluster Architecture



VERITAS Cluster Architecture

- Up to 32 nodes shipping today, but architecture designed for 100s
- NT, Solaris, HP-UX support
- All nodes have same software, implement the same functionality
  - No quorum node, group membership manager
  - These functions implemented in purely distributed manner
    - Uses private interconnects, special low-latency protocol stack which enforces cross-cluster consistency & operation atomicity
    - Minimum of 3 separate heartbeat mechanisms
    - Handles the multiple failure case very cleanly

VERITAS Cluster Architecture

- Based on a resource object definition
  - Resource types may be storage, files, applications, IP addresses, NIC etc.
  - Resources may be collected into service groups (e.g. specific volume, file system, web pages & database, app, NICs, IP address for a web site)
  - Both failover & parallel groups defined
    - Failover target can be selected in several distributed ways (predefined list, round-robin, load balance etc.)
    - Parallel groups define number of nodes required to run service - equivalent to a quorum for a specific service
- All nodes register for specific storage & share Write Exclusive Registrants-Only reservation

Problem

- Private interconnect protocol operates essentially in one direction:
  - Resource is idled on Node (or Node Set) A
  - Distributed mechanism identifies next location
  - Resource transferred to Node B, made active @ Node B
- Protocol has difficulty with bi-directional process such as:
  - Tell Node B that resource will be transferred
  - Node B performs Preempt, when complete informs A
  - Resource is transferred from Node A, which deregisters
  - Resource made active on Node B
  - But what happens if Node B fails during transfer?

Need

- Reservation protocol which does not need to be invoked during resource transfers
  - Or which can be implemented with the “one direction” protocol, has no race conditions & leaves no data integrity exposure, even where multiple nodes fail concurrently
- Define a new Persistent Reservation type which maintains a Registrants Only reservation until the last Initiator deregisters
  - And ANY registered Initiator can Release
- Defined in detail in SPC-3 proposal in 01-204r1
  - Will be asking for approval @ November meeting

Summary

- New persistent reservation type would be appropriate for all types of distributed peer-to-peer clusters
- VERITAS has prototyped proposed definition
  - An array vendor implemented support for new code in less than 500 lines of code
  - Completed testing work indicates that code is race free and stable