

Truly Scalable NFS

Raju C. Bopardikar
CTO



What problems need to be solved ?

- The need to pool compute resources (compute grid / cluster)
- Need to independently choose physical raw storage or reuse existing raw storage
- Need to be able to scale non-disruptively
(Growth is certain but unpredictable)
- Need to stop over-provisioning resources



Solution has to provide

- Uniform access to all data from all compute nodes
- Access to data has to scale as compute nodes scale
- Performance has to scale as compute nodes scale
- No Single point of failure (Reliability)
- Survive multiple failures (Resiliency)



Why NFS ?

- Simplicity of deployment
- NFS does not require any special filesystem or driver support or integration on the application server
- Inherently OS, Platform and Application agnostic

Architectural issues impeding NFS scalability

- Data is physically bound to a single node/filer (connectivity/access and system bottleneck)
- Data has to be moved across a back-end cluster interconnect (Interconnect bottleneck)
- Scaling requires [disruptive] migration and redistribution of data
- Full performance can only be realized if data and access to data is artificially distributed uniformly

What is the benefit ?

Why should you care ?

- Enabled to create and leverage a true Compute resource pool (Grid / Cluster) [much better ROI]
- Tremendous reduction in complexity of management and operations of clusters and applications [Lower Opex Costs]
- Reliable (HA) File serving costs are significantly reduced as a function of $N+m$ as opposed to $2N$ or greater [Lower Capex Costs]
- Reconfigure and subdivide cluster to suit application needs (fully utilize all resources)
- Upgrade / reboot / change personalities of compute nodes on the fly
- Enabled to match processing to the natural workflow of applications
- Reduce overall end-to-end processing cycle time

• ----- DO MORE REAL WORK !!! -----

What can be achieved ?

