NetApp

Go further, faster™

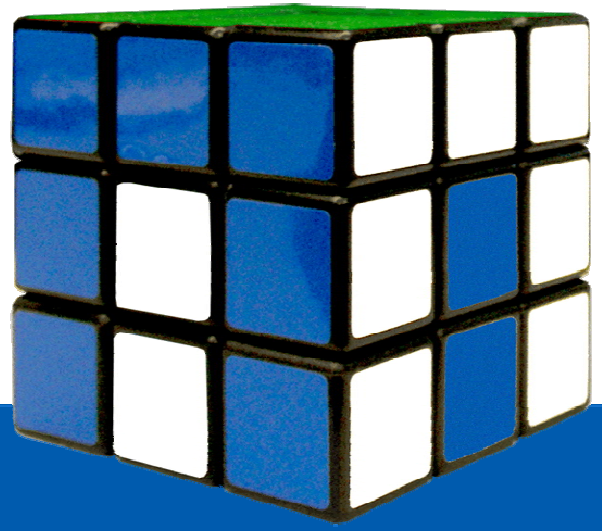
Clustered ONTAP pNFS Server (WIP)

Pranoop Erasani
pranoop@netapp.com
Connectathon – Feb 24, 2009

Agenda

- Clustered ONTAP Architecture
- Striped WAFL
- pNFS and Striped WAFL
- Performance
- What Next?
- Q&A

Clustered ONTAP Architecture

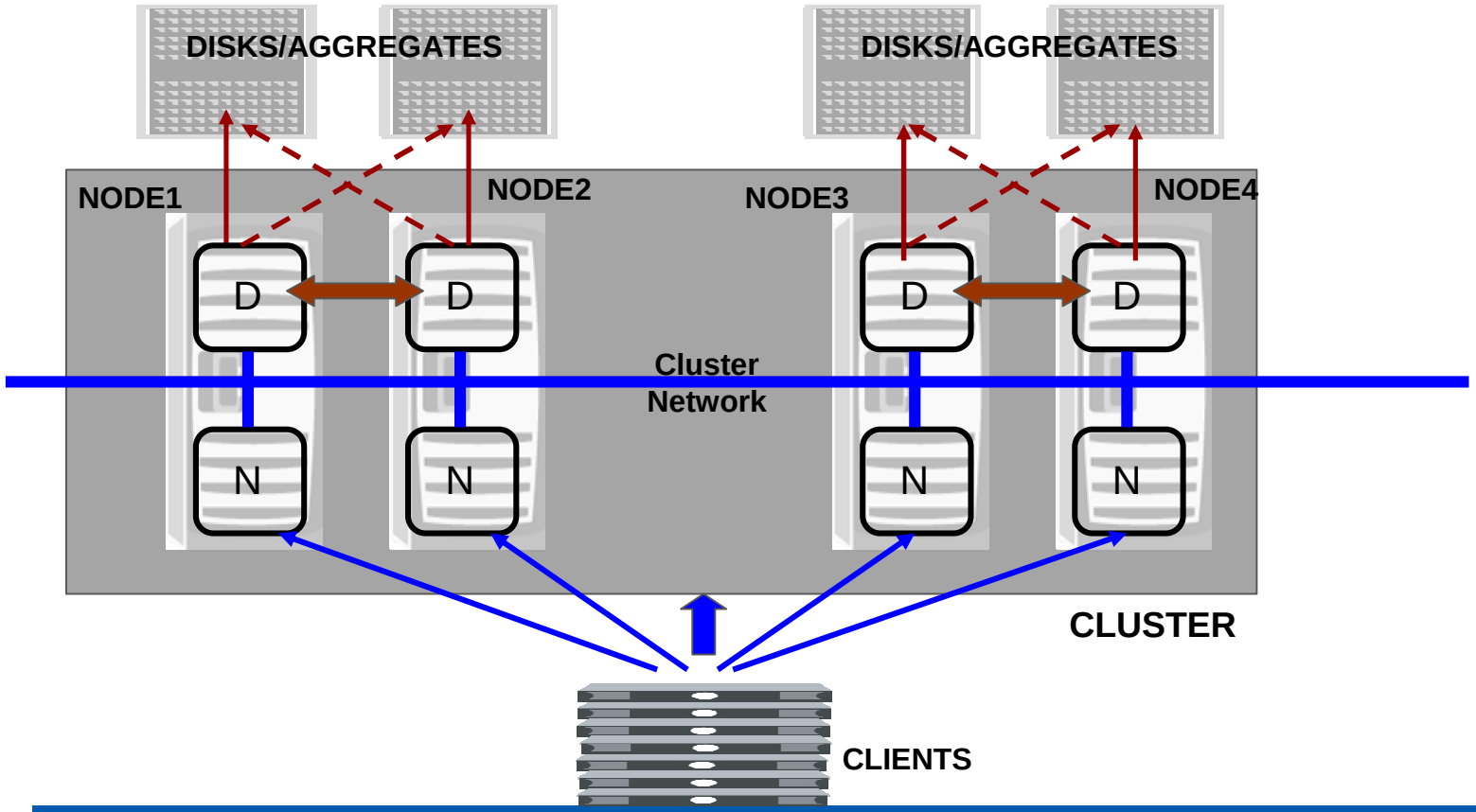


Clustered ONTAP

- NetApp's Next-generation ONTAP
- Basically, clustered system of HA pairs
- Building blocks
  - N-blade
  - D-blade
  - VLDB
  - LifMgr (or VifMgr)
  - SpinNP
  - Others (Management)….
- Primarily built for Global namespace
  - Junctions (a new filesystem object) stitch the namespace
  - Each vserver has it's own namespace stitched from volumes in the cluster
  - Each vserver has a root volume and rest of them are brought into namespace via junctions

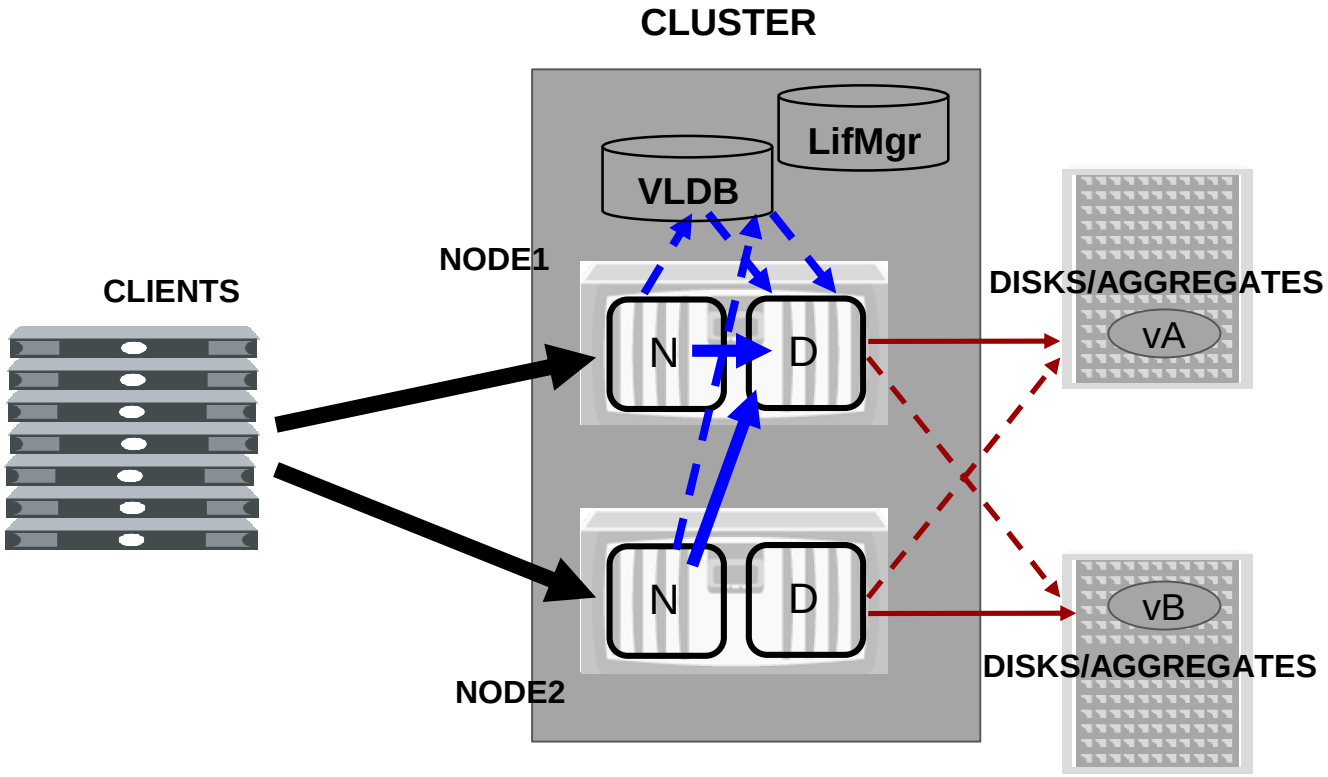
Clustered ONTAP outline

- Cluster of HA pairs - HA pairs help in storage failover
- N-blade: Client-facing, owns networking, protocol stack
- D-blade: Owns disks, aggregates (disk groups) and thus volumes
- High Speed interconnect for cluster traffic



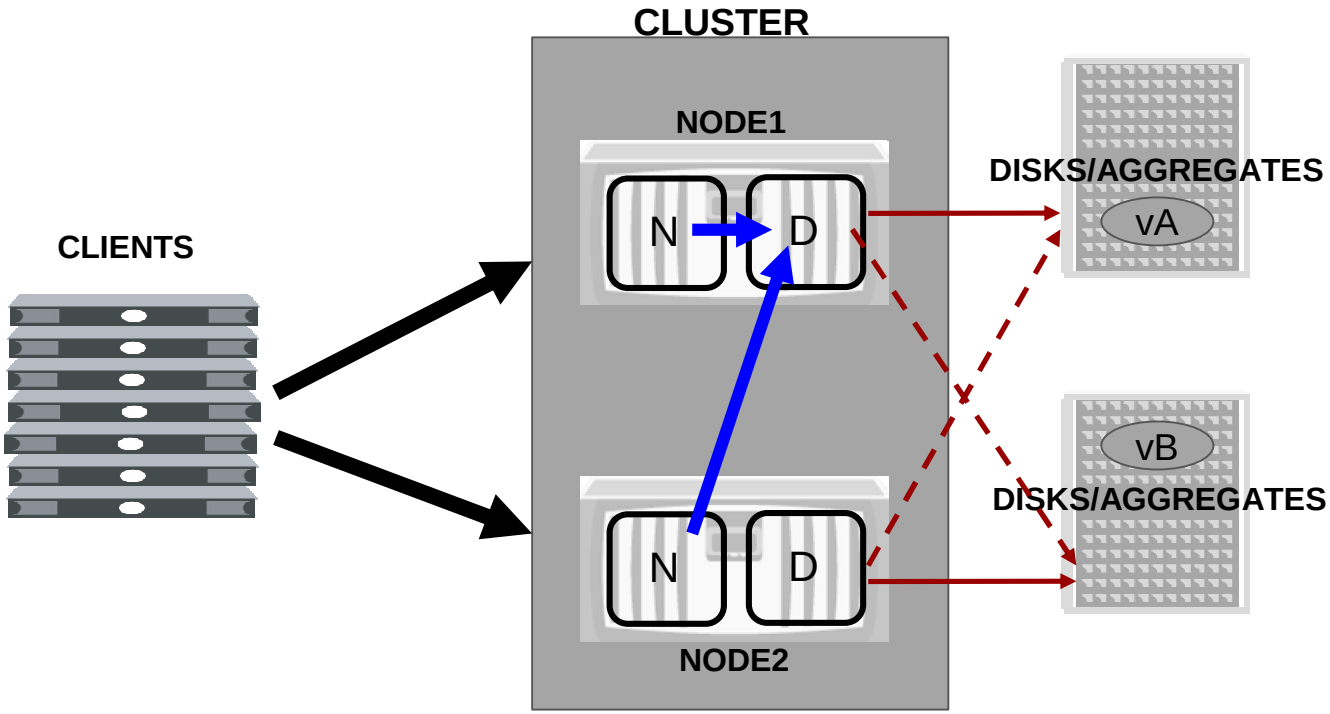
Scale-out cluster

- **N-blade**: Client-facing, owns networking, protocol stack
- **D-blade**: Owns disks, aggregates (disk groups) and volumes
- **LifMgr**: LIF Manager, manages networking related information
- **VLDB**: Volume Location Database, manages name space information
- **SpinNP**: Protocol for communication between Blades

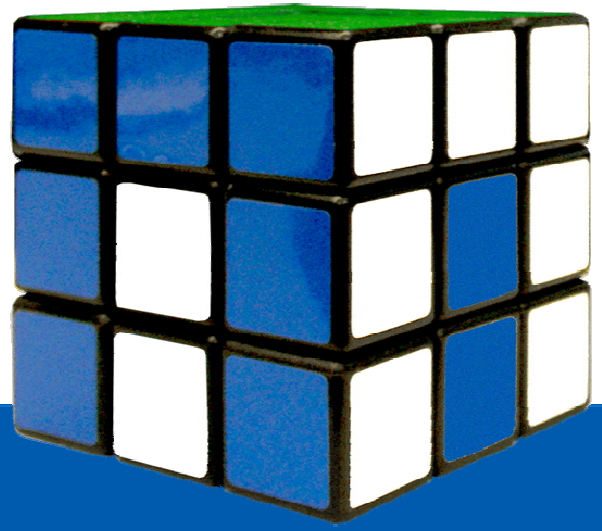


Flexible Volumes

- A D-blade owns aggregates (i.e.., RAID protected disk groups)
- Flexible volume sits within an aggregate
- Thus, at any moment, a flexible volume (vA or vB) sits on a D-blade
- BTW, flexible volumes can be moved within the cluster
- Owning D-blade serves **SpinNP** protocol requests from all N-blades
- All protocol requests (NFSv[2,3,4,4.1], NLM, CIFS) converted to **SpinNP** requests

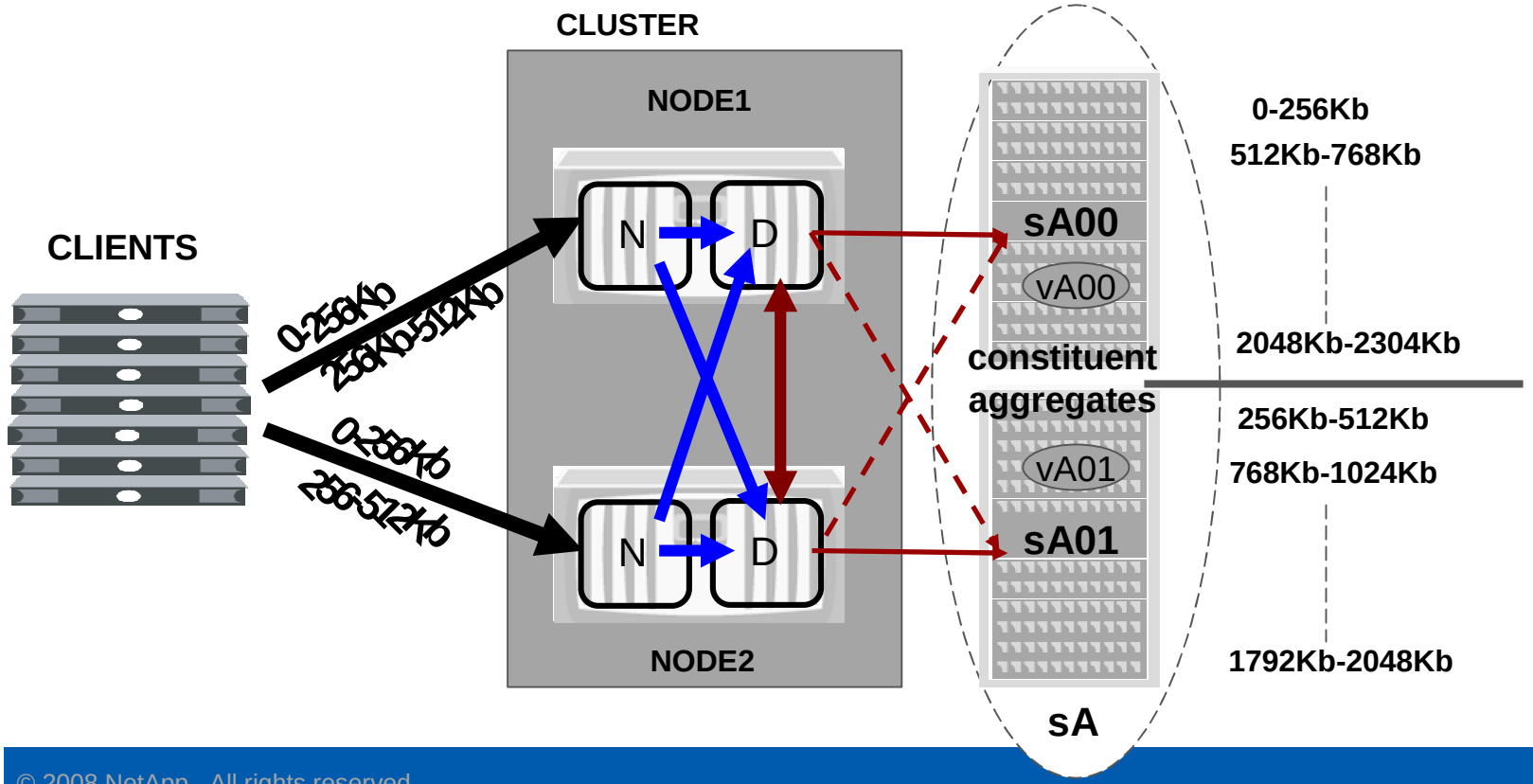


Striped WAFL



Striped Volumes

- A striped volume (vA) has constituent volumes (vA00, vA01)
- Striped aggregates (sA) – aggregates that hold striped volumes (sA00, sA01)
- Thus, Each D-blade could own a constituent volume (multiple too)
- Data gets distributed/striped (e.g., 2 stripes, 256Kb) across constituent volumes
- N-blade routes request based on striping configuration present in VLDB



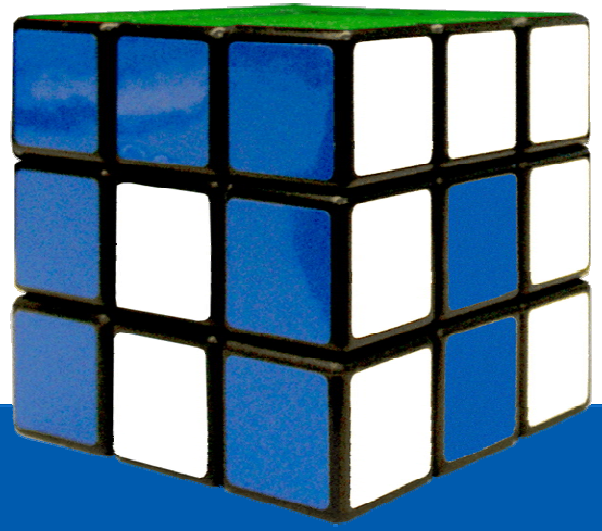
Striped Volumes (Continued)

- Supports Data, Metadata striping
  - Data gets striped onto constituent volumes
  - Metadata owner varies with each file
  - Each constituent volume owns metadata of some files
- Supports Directory striping
  - Directory contents striped across constituent volumes
- It repeats every 4096 stripes (proprietary algorithm)
- Example configuration (two constituent volumes) is a special case
  - As you may have seen in the previous slide
  - i.e.., we don't put two adjacent stripes on the same constituent volume
  - Appears as if its round robin

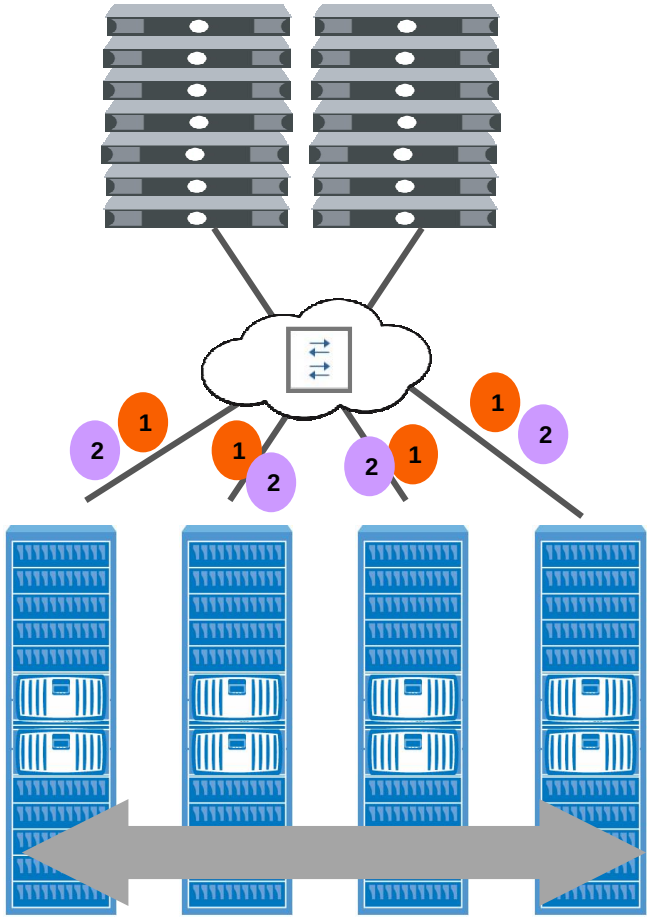
Striped Volumes Terminology

- Stripe Count
  - Number of constituent volumes (2 to 254)
  - Striped volumes logically ID'd from 0 to N - 1
- Stripe Width (pNFS stripe unit)
  - Maximum number of data bytes written and read to and from a specific member-volume (128K to 1G)
- Striping Table
  - Table that establishes striping pattern of a striped volume
  - Size fixed at 4096 entries – irrespective of stripe count
  - Basically, Pattern repeats after 4096 stripes
  - Elements in the table will be logical ID's of stripes
  - Proprietary algorithm – Not to be disclosed

pNFS and Clustered ONTAP

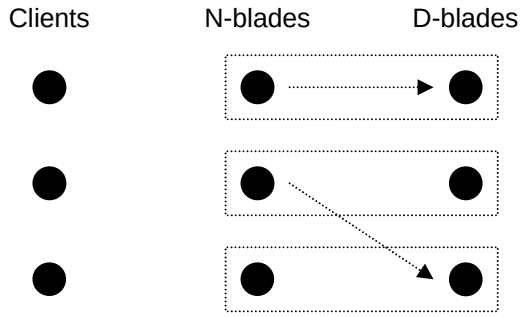


Clustered ONTAP and pNFS



- Leverage cluster backend
- In the context of global name space
- Striping with WAFL Striped volumes
- Avoid single-blade data bottleneck
- Solve the N-blade latency problem with striped volumes
- What about Flexible Volumes?

Performance through Scaling (No pNFS)



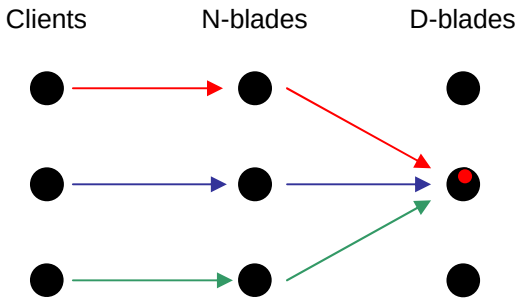
no Striping, no pNFS

Striping without pNFS

pNFS without Striping

Striping with pNFS

Performance through Scaling (No pNFS)



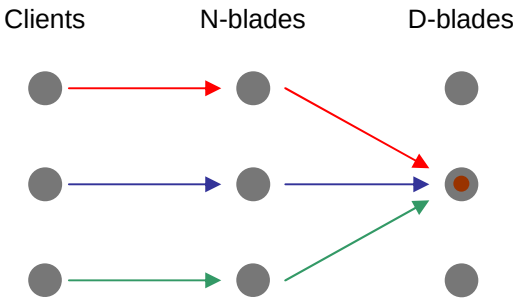
no Striping, no pNFS

Striping, no pNFS

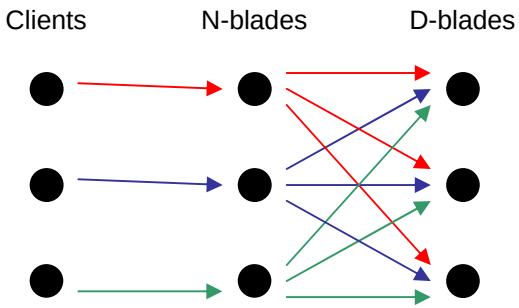
no striping, with pNFS

Striping, with pNFS

Performance through Scaling (No pNFS)



no Striping, no pNFS



Striping, no pNFS

no striping, with pNFS

Striping, with pNFS

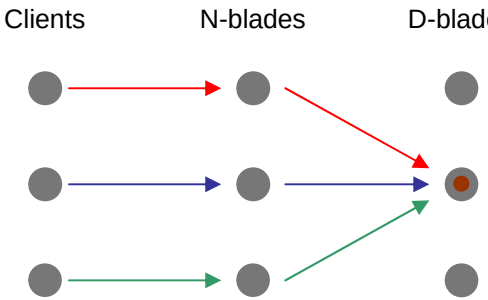
pNFS in Clustered ONTAP

- Striping helps in D-blade utilization
  - Striping alone doesn't help resolve N-blade bottleneck
  - Striping + pNFS will solve that problem
- pNFS in Clustered ONTAP
  - Parallel I/O via Multiple LIFs
- Any LIF could be pNFS metadata server
  - Usually the one the client mounts
- Remaining LIF's are data servers
  - Depending on the configuration
- Support sparse layouts
  - Routing within the cluster backend is based on logical file offset

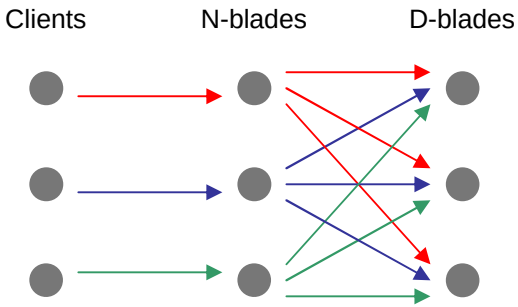
pNFS with Flexible Volumes

- Supported - Remember we are a cluster
- All volumes in the cluster are accessible via all N-blades
- Parallel IO via Multiple LIFs on different N-blades will work
- But, D-blade hosting the volume becomes bottleneck
- Thus, the real value is in supporting striped volumes with pNFS

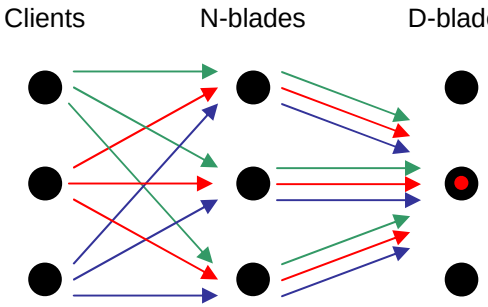
Performance through Scaling (pNFS)



no Striping, no pNFS



Striping, no pNFS



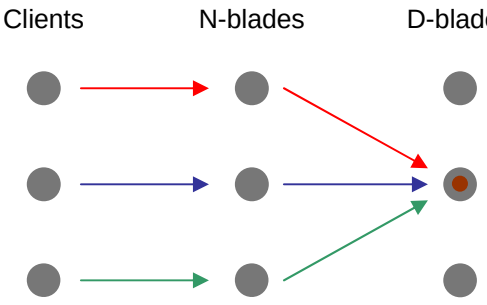
no striping, with pNFS

Striping with pNFS

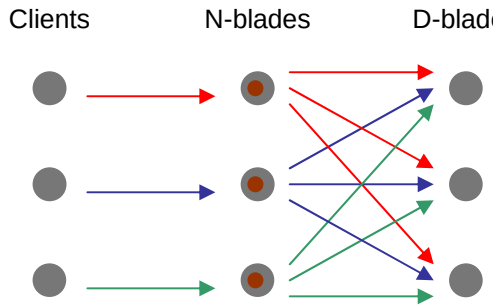
pNFS in Clustered ONTAP

- Importance given to configuration that will give maximum performance
  - Find a LIF on the same node as constituent volume
  - i.e.., Pair a LIF and a Stripe
  - Export the striping geometry to the client
- Best performance – Data server LIF sits on same cluster node as stripe
  - Best performance is not always the requirement
- COT pNFS implementation makes that happen automatically

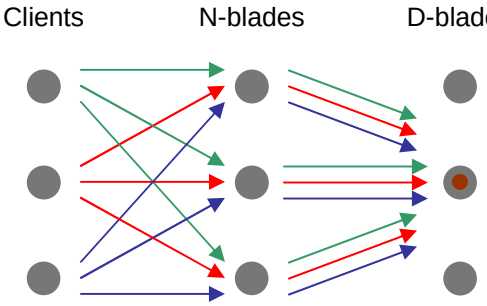
Performance through Scaling (pNFS)



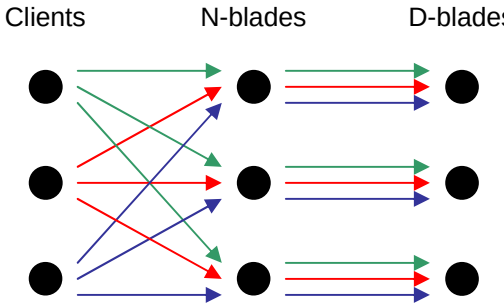
no Striping, no pNFS



Striping, no pNFS



pNFS without Striping

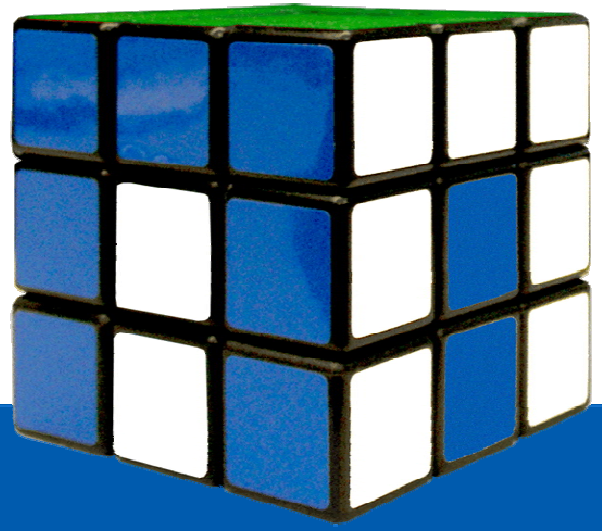


Striping with pNFS

COT pNFS mechanics

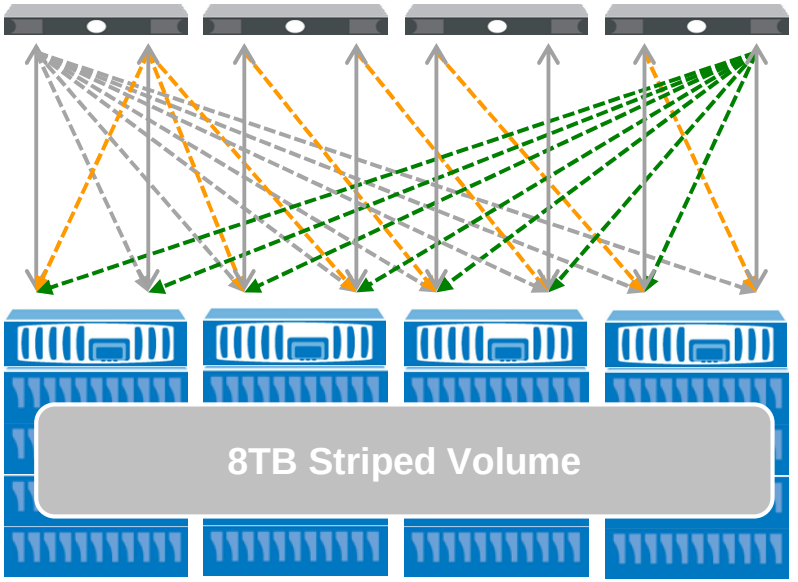
- pNFS stripe(indices) count = striping table size
  - i.e. always 4096 stripe indices
  - Yes, our GETDEVICEINFO response is large
- No. of pNFS device addresses = stripe count
  - Of course, does not consider multipathing
- pNFS first stripe index
  - Varies for each file
  - An index into the striping table of striped volume
  - Thus, different files start on different constituent volumes

Performance

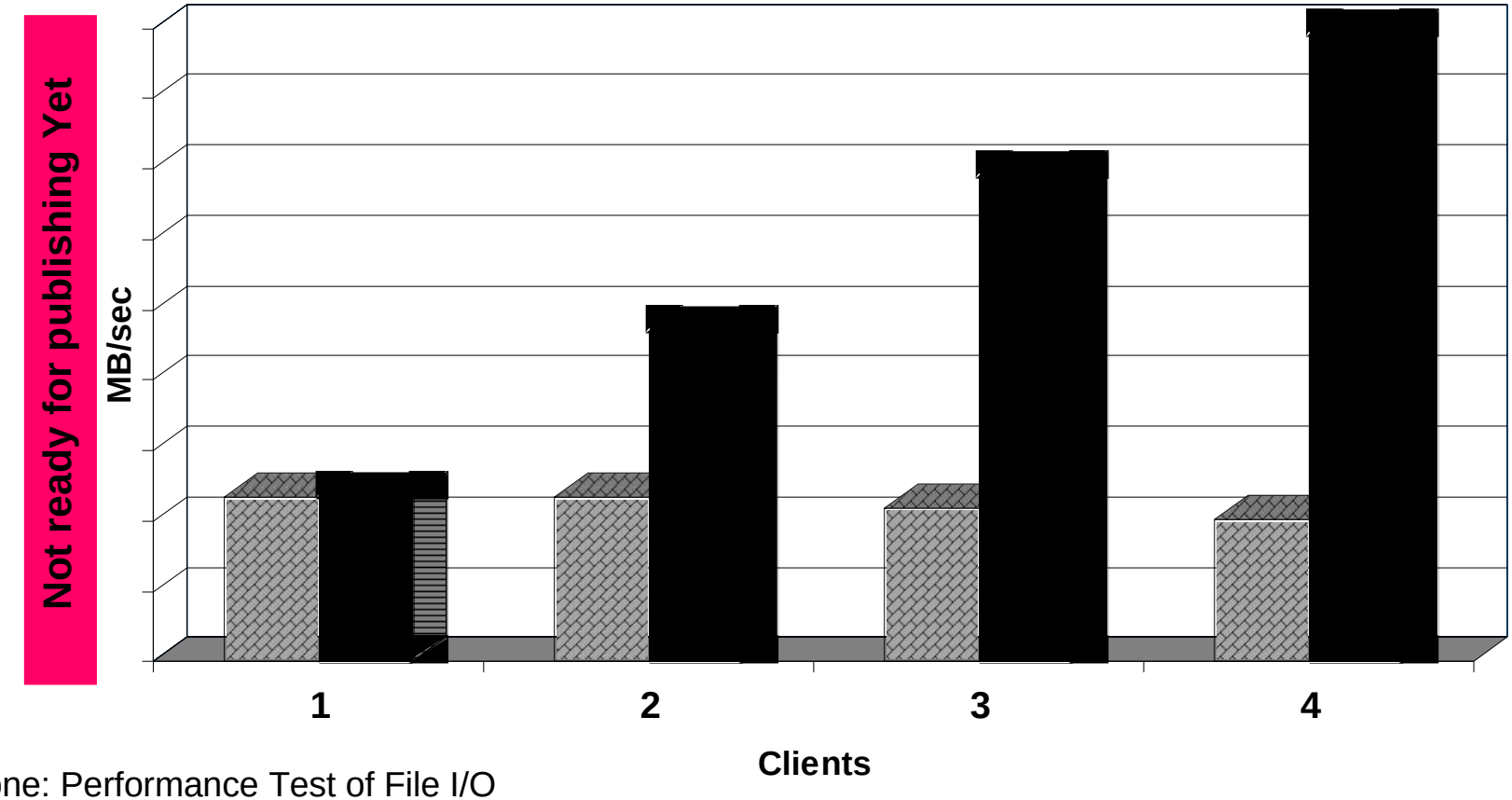


COT pNFS Test Configuration

- 4-Node FAS3070 Cluster
- 4 Linux Clients – 1GbE
- 4xDS-14 Shelves per node with15k Fiber Drives
- Data ONTAP 8: Cluster Mode
- An 8 TB Acro striped volume
  - Each member volume on a different node of the cluster
- Client automatically talks pNFS to the server on a striped volume
- 9 LIFs, 3 on one node, 2 each on other nodes
  - 1 LIF acts as Metadata Server
  - 8 LIFs (2 on each) act as Data Servers
- Two separate tests
  - Use dd to fill the volume
    - "dd bs=32k if/dev/zero of=/mountpoint/filename
  - Iozone



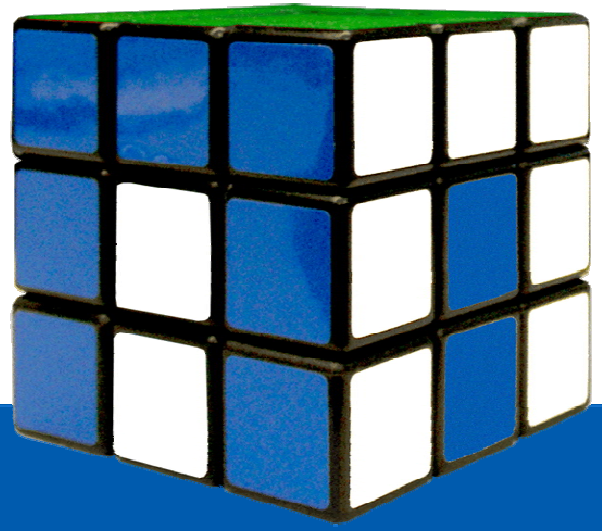
pNFS Iozone Linear Scaling Results



Iozone: Performance Test of File I/O
Version $Revision: 3.311 $
Compiled for 32 bit mode.
Build: Linux 2.6.26 w/pNFS support
Each process writes a 8388608 Kbyte file
in 32 Kbyte records

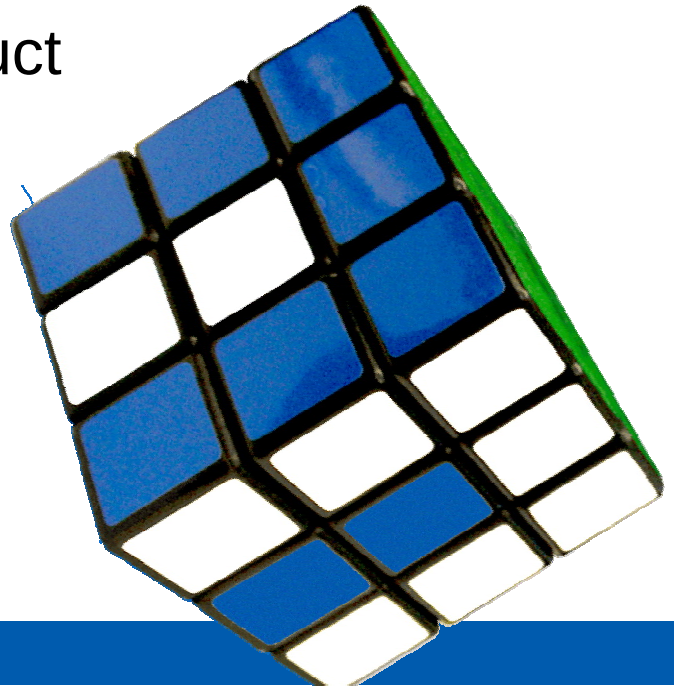
Client Throughput
Aggregate Throughput

Final Thoughts



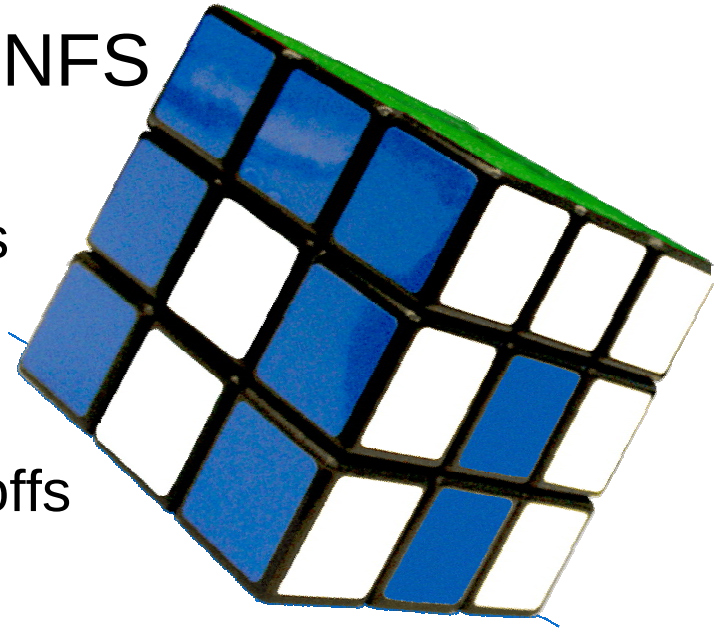
Prototype Timeline

- First Interoperability Testing
  - NFSv4.1 Sessions + pNFS with Striped Volumes
  - Bakeathon – Austin, TX – 2008, Sep 15 – 19
  - Successful with Linux, Solaris pNFS clients
- Next, SuperComputing ’08
  - Demonstrated linear scaling gains to key customers
- Will evolve into final product

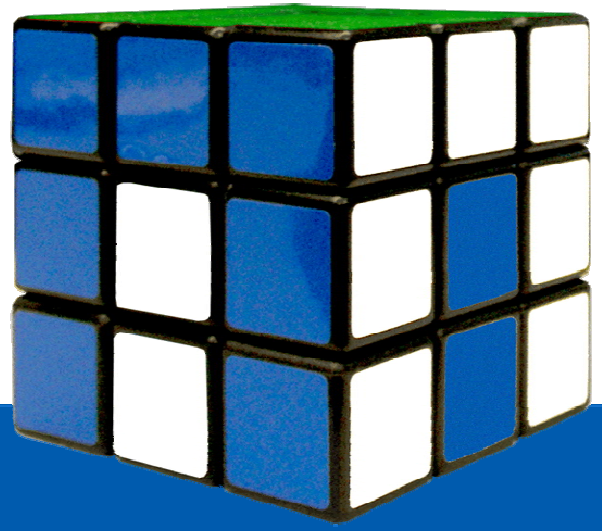


What Next?

- Actively designing in COT pNFS
  - Segmented layouts
  - Read-write vs.. read-only layouts
  - pNFS Multi-pathing
  - Restriping Behavior
  - Performance vs. reliability tradeoffs
  - UI smartness (future)
    - Policies in picking up LIFs
      - Weightage associated with LIF's
    - pNFS LIF groups
      - allows customers to configure a subset of LIF's
    - LIF-volume association
      - Pair a striped volume with LIFs
      - Sorta anti-GNS



Thank You
Q&A



Acknowledgements

- Mike Eisler
- Dave Noveck
- Michael Hein (Sr. Manager, NFS)
- Dan Muntz (Linux NFS)
- Richard Jernigan (Striped WAFL)
- And you folks…….

References

- pNFS problem statement
  - http://www.pdl.cmu.edu/pNFS/archive/gibson-pnfs-problem-statement.html
- NFSv4.1 Draft
  - http://tools.ietf.org/html/draft-ietf-nfsv4-minorversion1-29
- pNFS Tech ONTAP article
  - http://www.netapp.com/us/communities/tech-ontap/pnfs.html
- Mike Eisler's metadata striping proposal
  - http://tools.ietf.org/id/draft-eisler-nfsv4-pnfs-metastripe-01.txt
- Mike Eisler's Blog
  - http://blogs.netapp.com/eislers_nfs_blog/