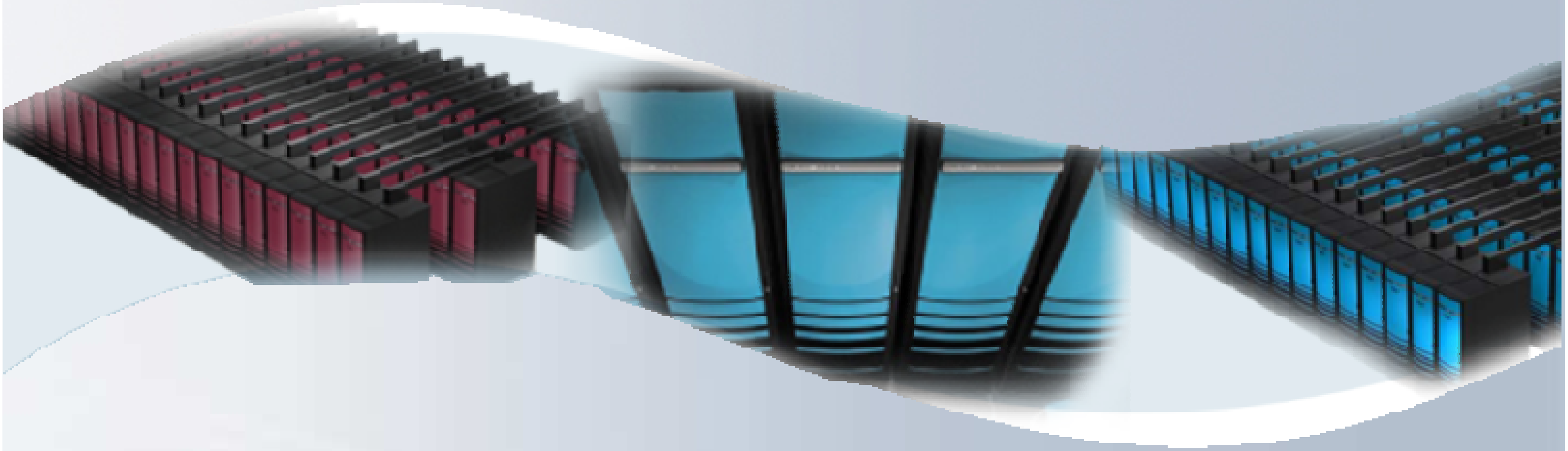
Cray Data Virtualization Service

a Method for Heterogeneous File System Connectivity



David Wallace
and
Stephen Sugiyama

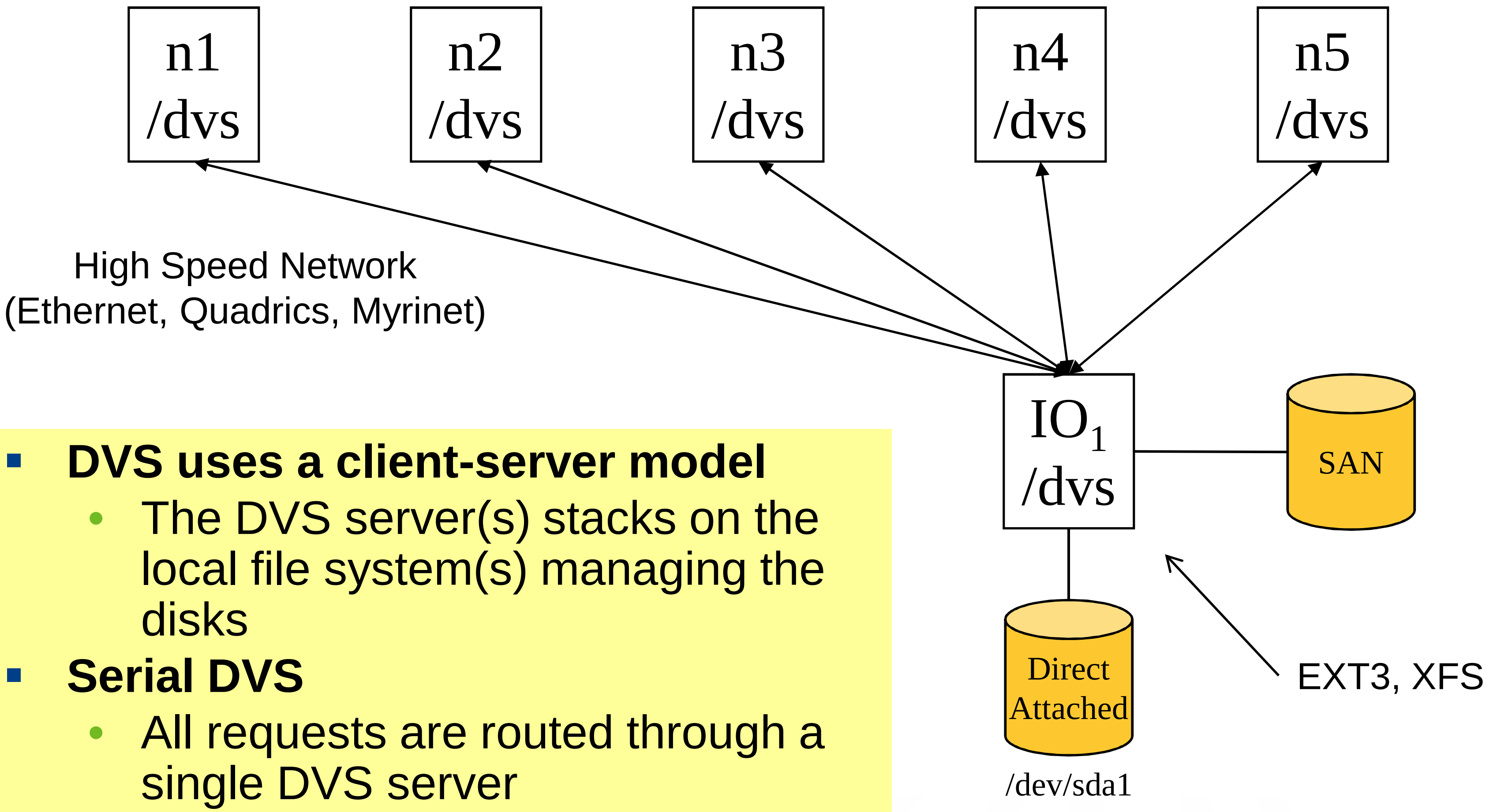
Cray Data Virtualization Service

- A little background and history
- Cray DVS Support in CLE 2.1
- What if…

DVS: Background and history

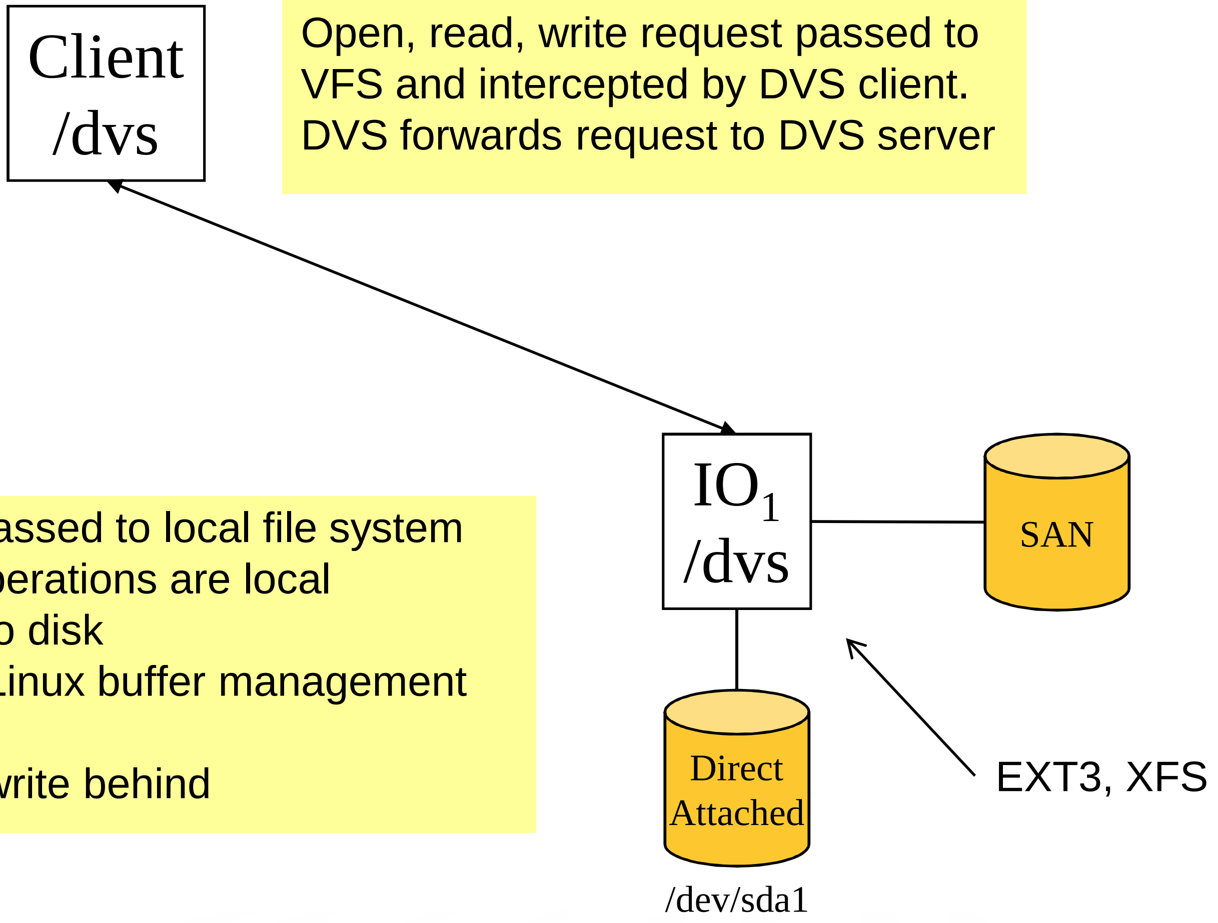
- Concept derived from Cray T3E system call forwarding
  - Focused on I/O forwarding aspects
- Initial work focused on clusters
- Some design objectives
  - Provide a low(er) cost alternative to having HBAs on all nodes in a cluster
  - Utilize bandwidth and capacity of a cluster's high speed network
  - Provide global access to file systems resident on I/O nodes
  - Provide high performance, parallel file system access
  - Target I/O patterns of High Performance Computing applications
    - Very large block sizes
    - Sequential access
    - Low data re-use

Serial DVS: Multiple Clients to a Single Server



- **DVS uses a client-server model**
  - The DVS server(s) stacks on the local file system(s) managing the disks
- **Serial DVS**
  - All requests are routed through a single DVS server
  - Provides similar functionality as NFS

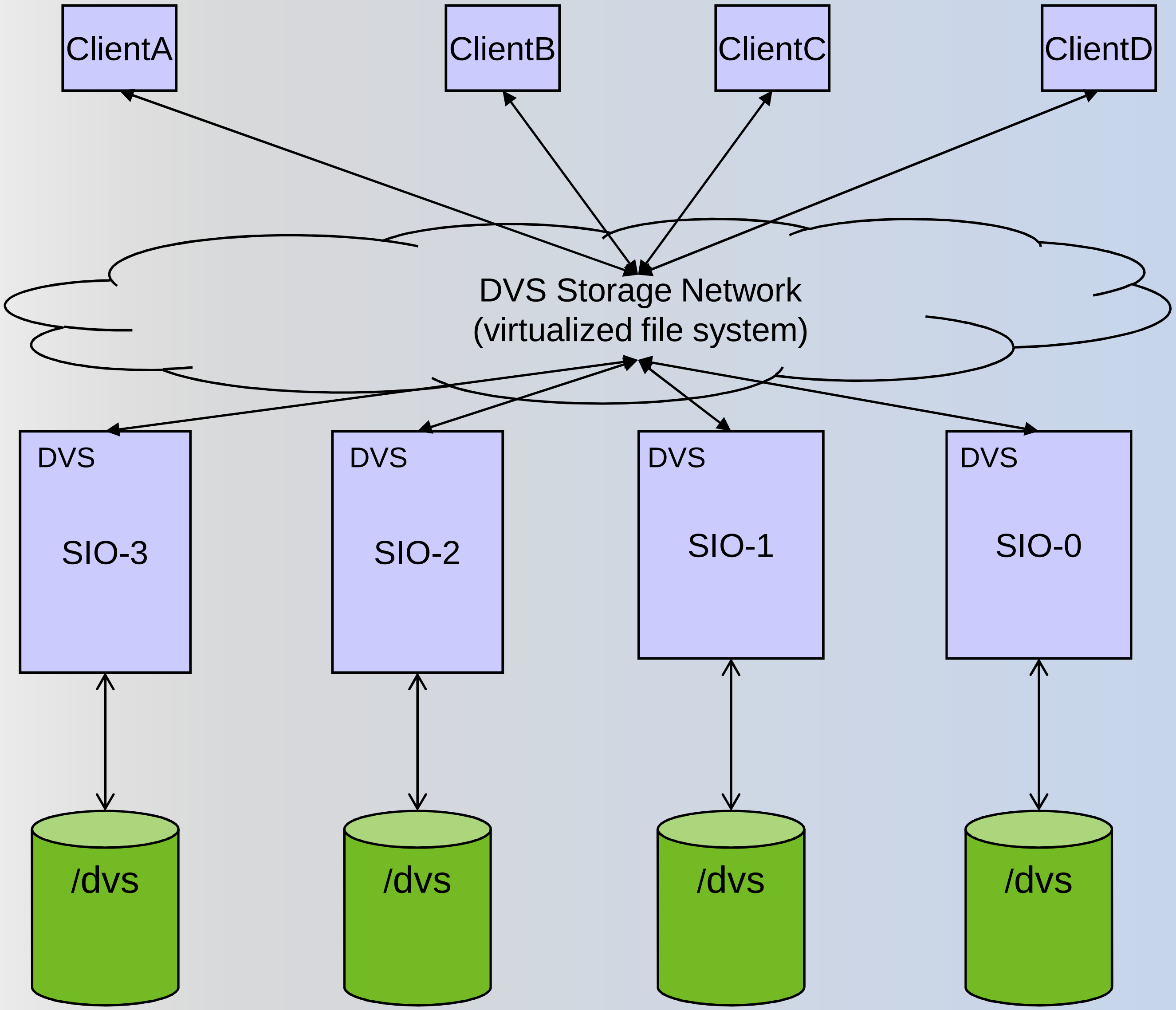
Single DVS Server to a Single I/O Device



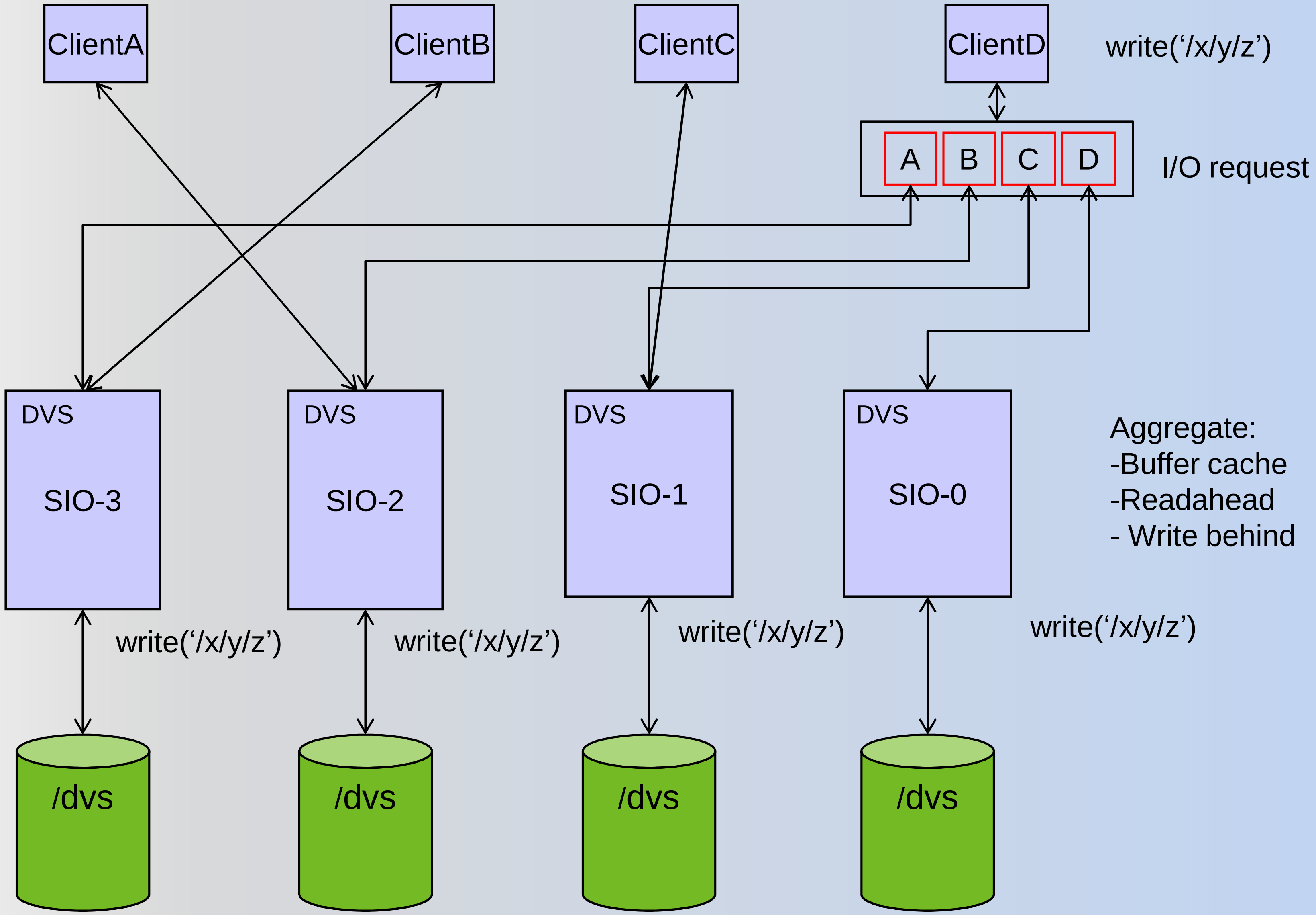
Open, read, write request passed to VFS and intercepted by DVS client. DVS forwards request to DVS server

- On server, request passed to local file system
- Meta data, locking operations are local
- Data is read/written to disk
  - Uses standard Linux buffer management
  - Local cache
  - I/O readahead/write behind

DVS: Multiple I/O node support



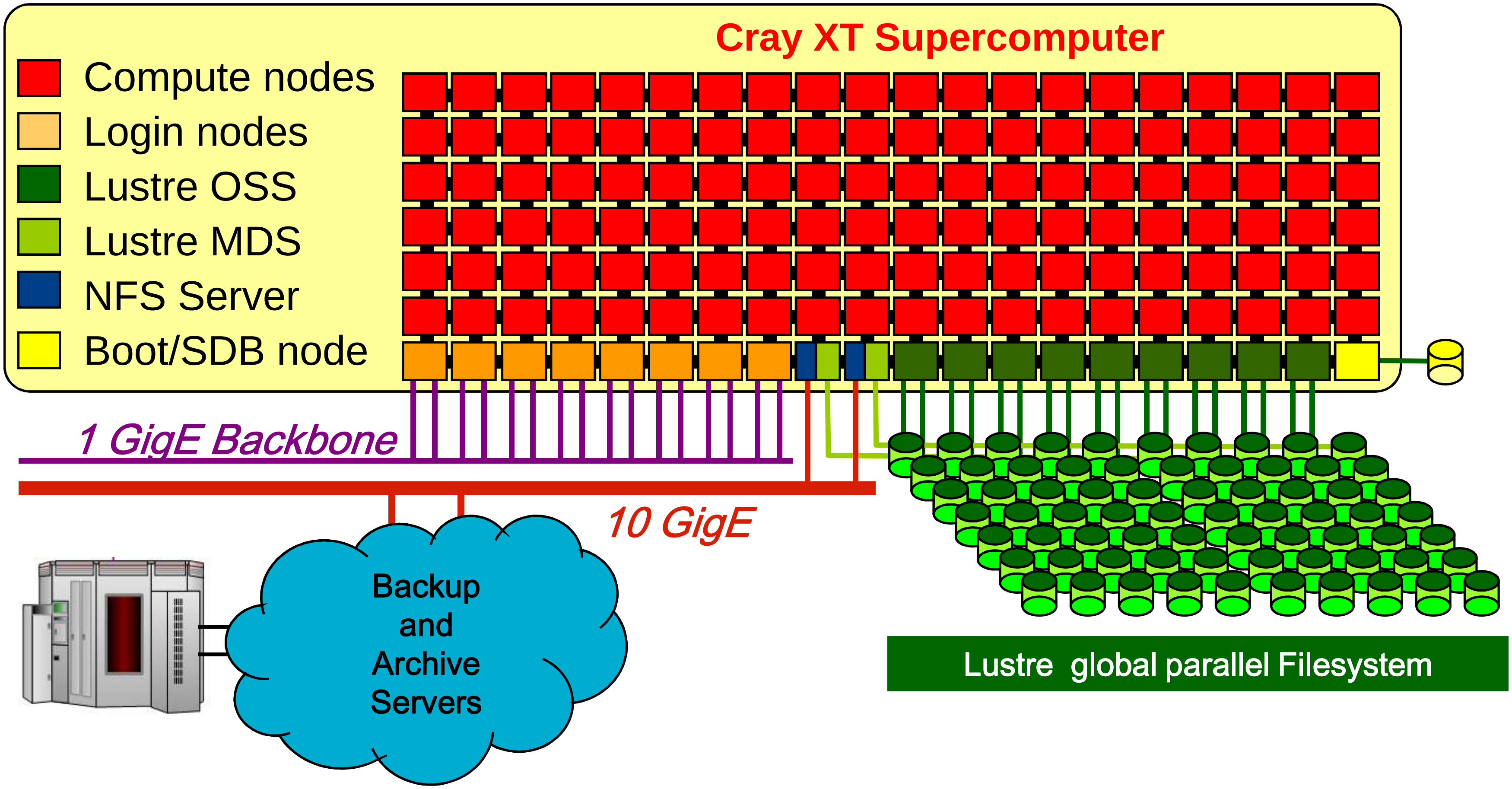
DVS: Parallel file system



Aggregate:
-Buffer cache
-Readahead
- Write behind

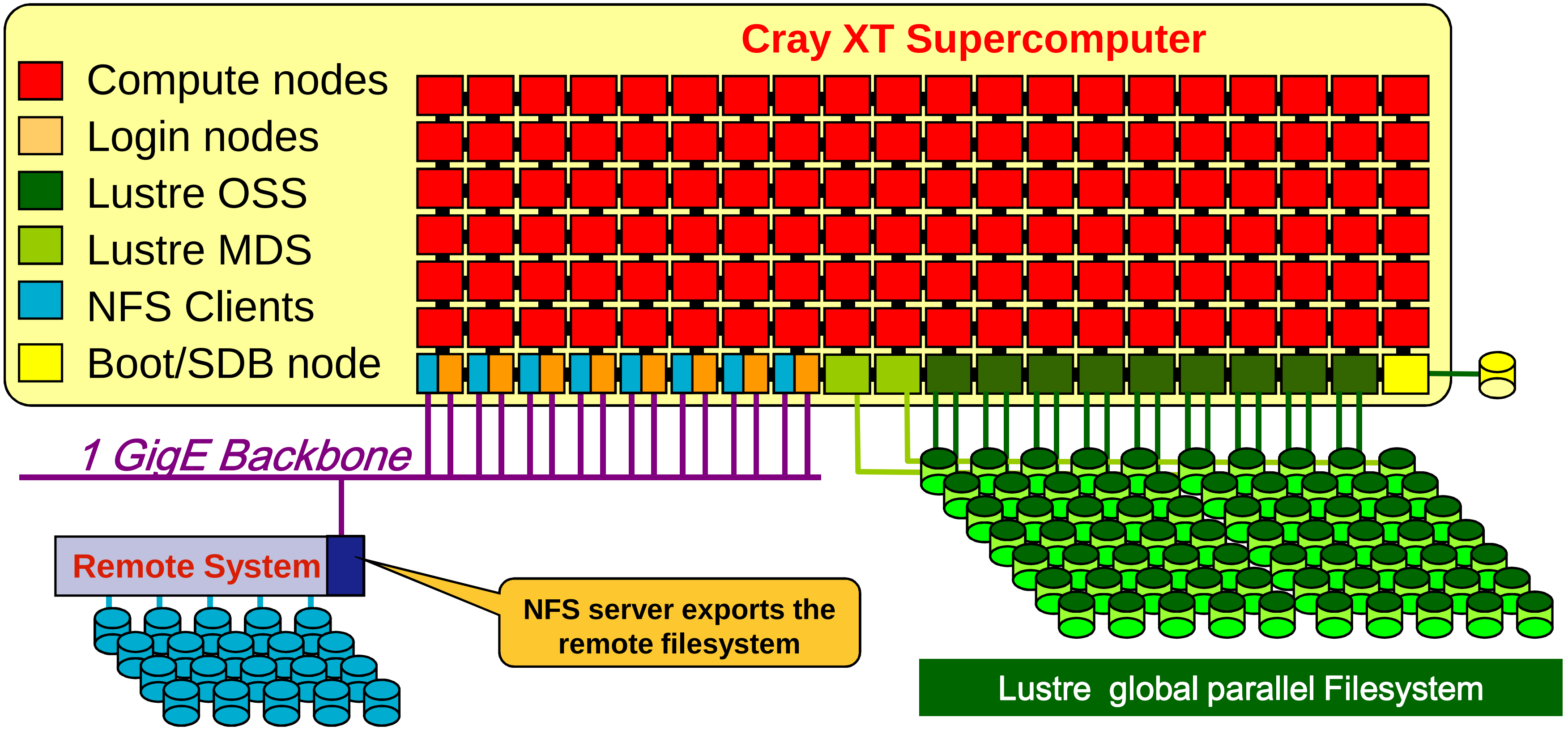
SO, WHERE ARE WE TODAY?

Cray XT Scalable Lustre I/O: Direct Attached



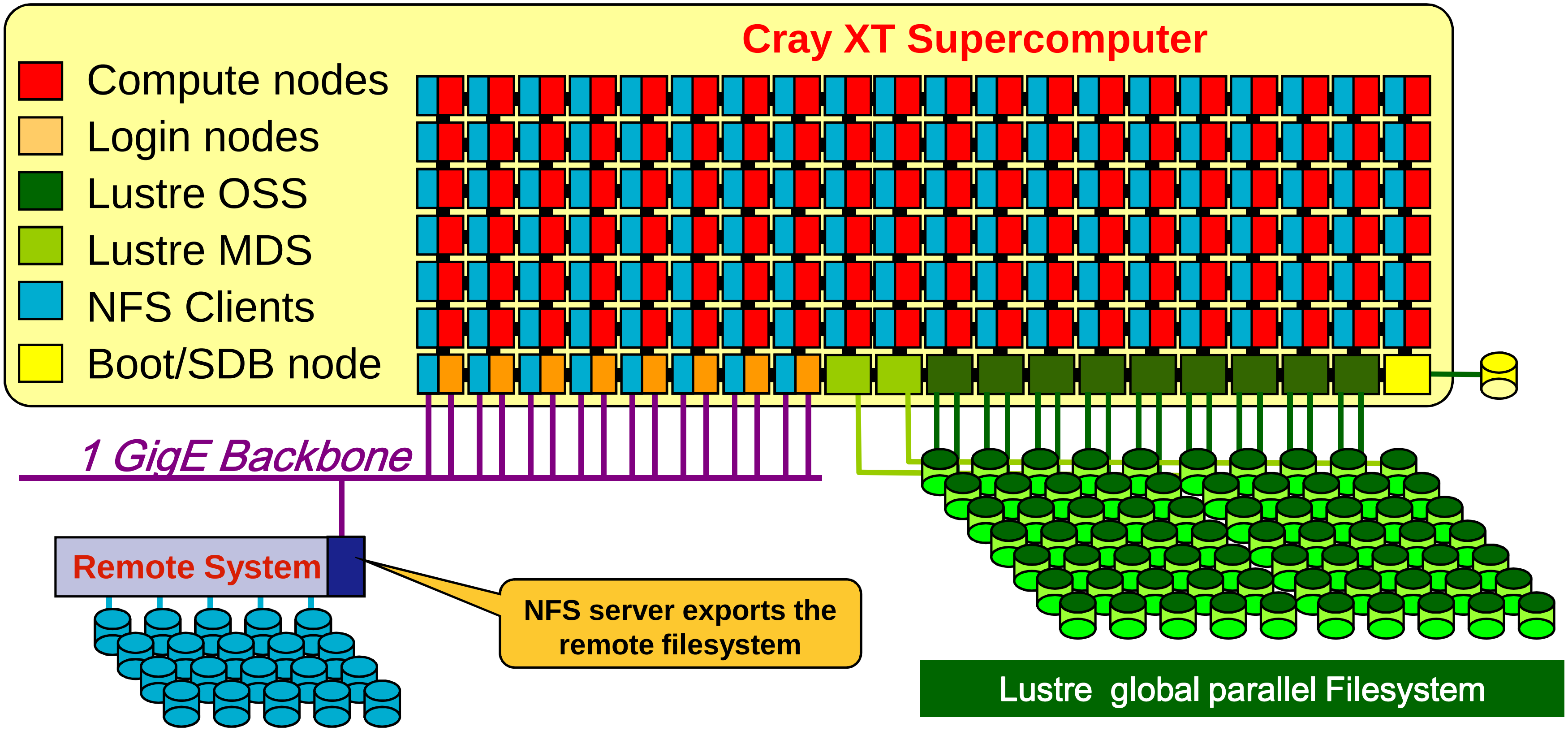
- Each compute node runs a Lustre client
- The NFS server would allow to export the Lustre filesystem to other systems in the network

Cray XT accessing Lustre and Remote Filesystems



- Each Compute and Login node runs a Lustre client
  - Lustre is accessable by every node within the Cray XT system
- Each Login node imports the remote filesystem
  - Filesystem only accessable from Login nodes
  - For global accessability, login nodes need to copy files into Lustre

Why not use NFS throughout?



Cray XT Supercomputer

🟥 Compute nodes

🟧 Login nodes

🟩 Lustre OSS

🟩 Lustre MDS

🟦 NFS Clients

🟨 Boot/SDB node

1 GigE Backbone

Remote System

NFS server exports the remote filesystem

Lustre global parallel Filesystem

- Each Compute and Login node runs a Lustre client
  - Lustre is accessable by every node within the XT system
- Each Login **AND** Compute node imports the remote filesystem
  - The remote filesystem is accessable from all nodes
  - No copies required for global accessability

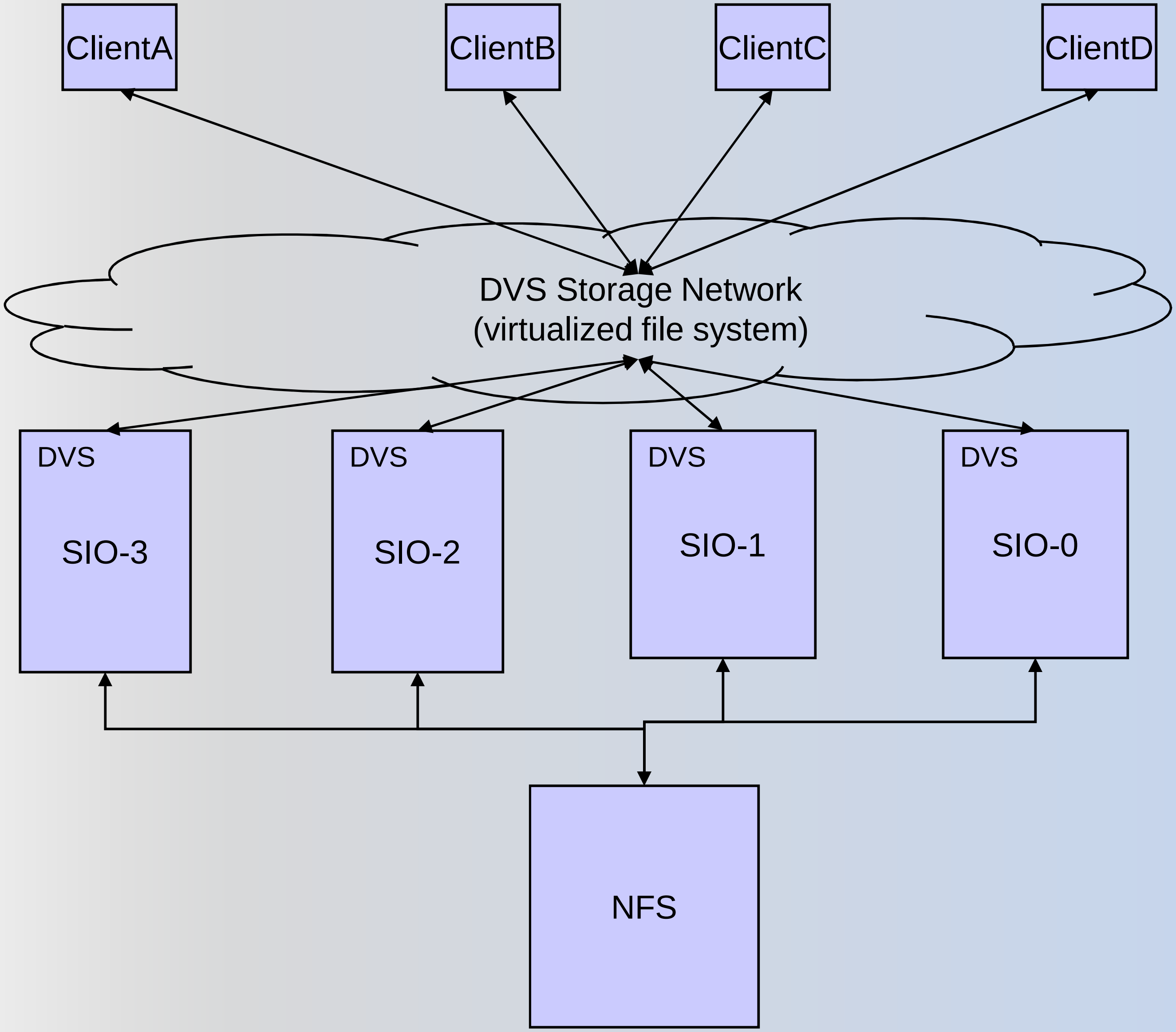
Issues with NFS

- Cray XT systems have thousands of compute nodes
  - A single NFS server typically cannot manage more than 50 clients
  - Could cascade NFS servers
    - There only can be a single primary NFS server
    - The other servers would run as clients on the incoming side and servers on the outbound side
    - Cascaded servers would have to run in user space
  - Complicates system administration
- Performance impacts
  - NFS clients run as daemons, thus introducing OS jitter on the compute nodes
  - The primary NFS server will be overwhelmed
  - TCP/IP protocol less efficient than native protocol within the Cray SeaStar network

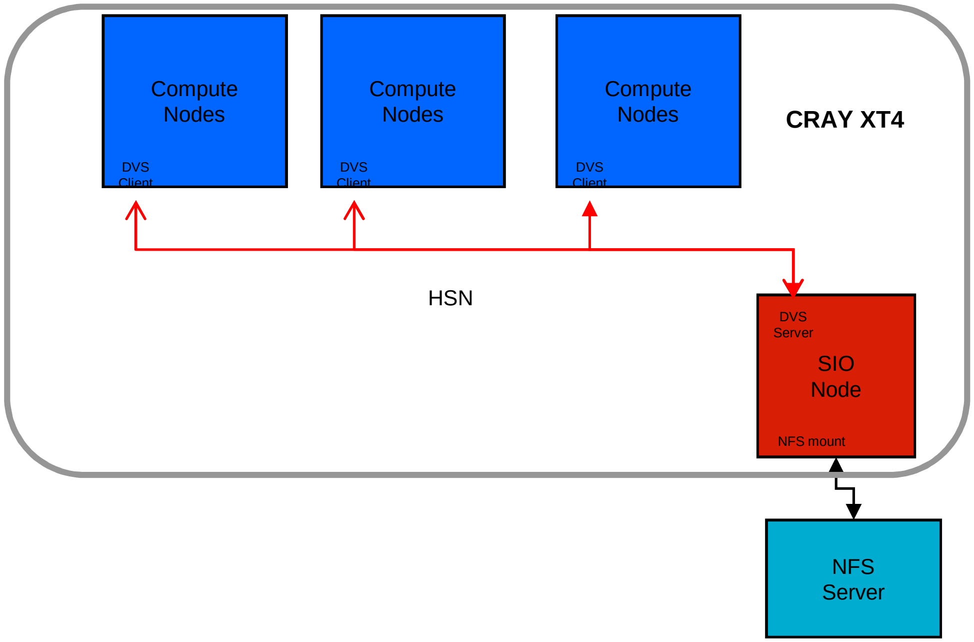
Immediate Need

- Access to NFS mounted file systems (/home on login nodes)
  - For customers migrating from CVN, equivalent functionality to YOD I/O

DVS: Support for NFS in CLE 2.1



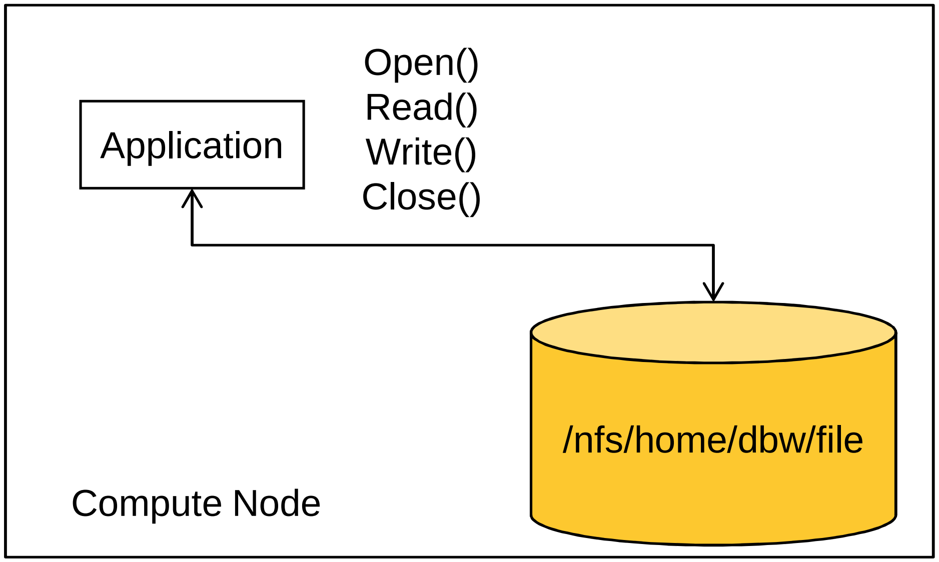
Cray DVS – from the admin point of view



- Admin mounts file systems in fstab per usual

```
mount -t dvs /nfs/user/file4
```

From Users Point-of-View



File system appears to be local

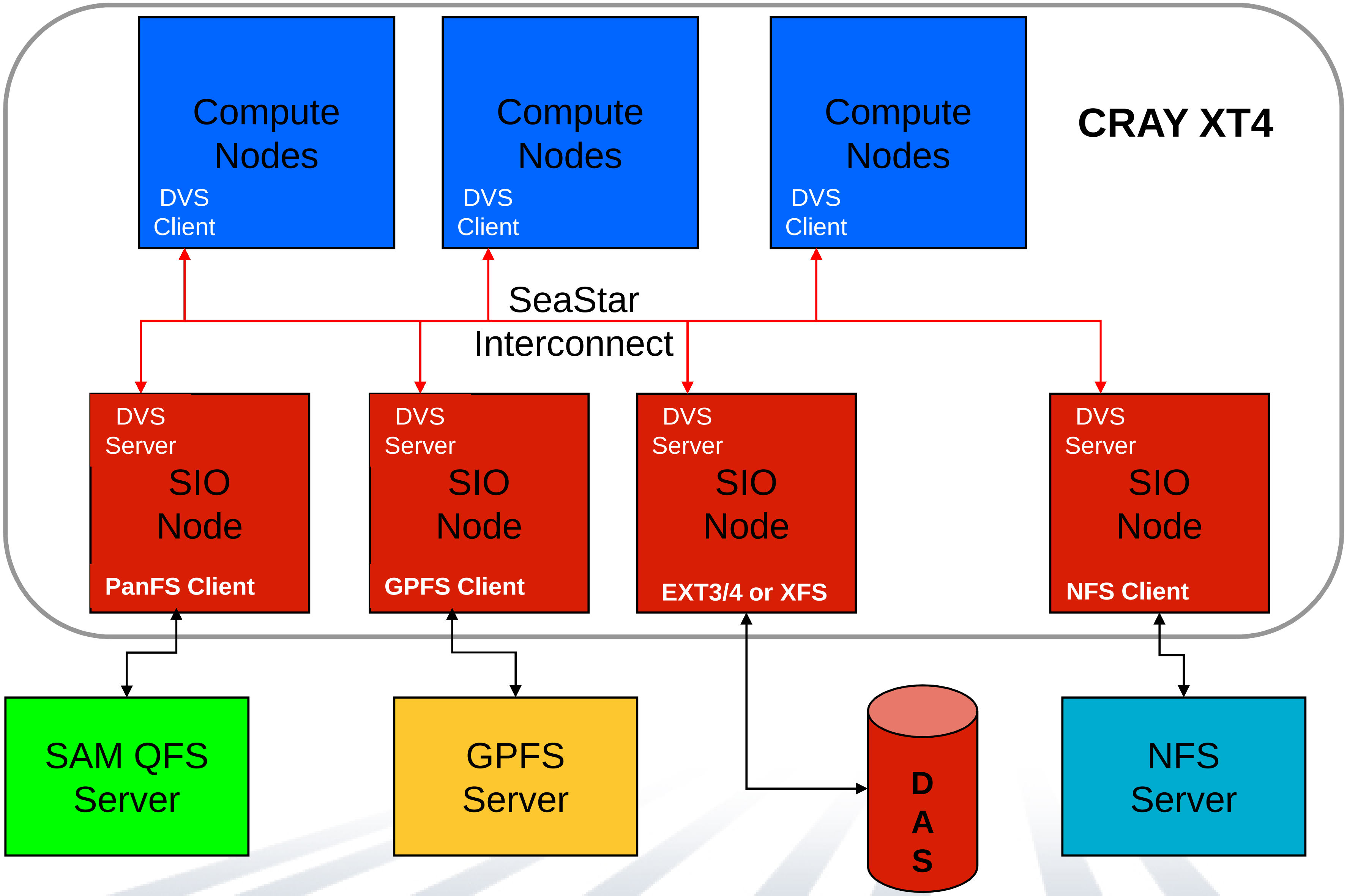
So what if …

- Users have data on one or more file systems (e.g. EXT3, XFS, GPFS, Panasas, NFS) on servers in the data center which they want access from the compute nodes on the Cray XT system WITHOUT having to copy files?

Then …

- You install Cray DVS (**D**ata **V**irtualization **S**ervice) server software on each of the existing file servers and you install DVS client software on each of the compute nodes and this allows the admin to “mount –t dvs” the file systems

DVS Concept in a Cray XT Environment



Cray DVS - Summary

- No file copying required!
- Simplicity of DVS client allows for larger scale than NFS or cluster file systems
- DVS can amplify the scale of compute nodes serviced to O(10000)
  - Can project a file system beyond limit of underlying clustered file system
    - GPFS on Linux is limited to 512 clients
- The seek, latency and transfer time (physical disk I/O) for every I/O node is overlapped (mostly parallel)
- Every I/O node does read-ahead and write aggregation (in parallel)
- The effective page cache size is the aggregate size of all of the I/O nodes page caches
- Allows the interconnects to be utilized more fully:
  - multiple I/O nodes can drive a single app node interconnect at it's maximum speed
  - multiple app nodes can drive all of the I/O node interconnects at their maximum speed
- Takes advantage of RDMA for those interconnects that support it (Cray SeaStar, Quadrics, Myricom)

Cray DVS – Initial Customer Usage

- ORNL
  - Began field trial in December 2007
  - Installed in production on ~7200 XT3 cores
  - Replacement for Catamount YOD-I/O functionality
  - Mounting NFS mounted /home file systems on XT compute nodes
- CSC – Finland
  - Working with Cray to test DVS with GPFS
  - Installed on TDS and undergoing development and testing
- CSCS
  - Begin early access testing 2Q2008

Acknowledgements

- David Henseler, Cray Inc
- Kitrick Sheets, KBS Software
- Jim Harrell, Cray Inc
- Chris Johns, Cassatt Corp
- The DVS Development Team
  - Stephen Sugiyama
  - Tim Cullen
  - Brad Stevens
  - And others

CRAY XT5

Thank You

PetaScales