CRAY

Improving I/O Bandwidth With Cray DVS Client-Side Caching

Bryce Hicks (bryceh@cray.com)

CUG 2017 CAFFEINATED COMPUTING
Redmond, Washington May 7-11, 2017

Agenda – DVS Client-Side Cache

- Introduction
- Motivation
- Interfaces
- Design
- Results
- Summary
- Q & A

Introduction

- Computational capabilities of large-scale HPC systems continue to improve

Introduction

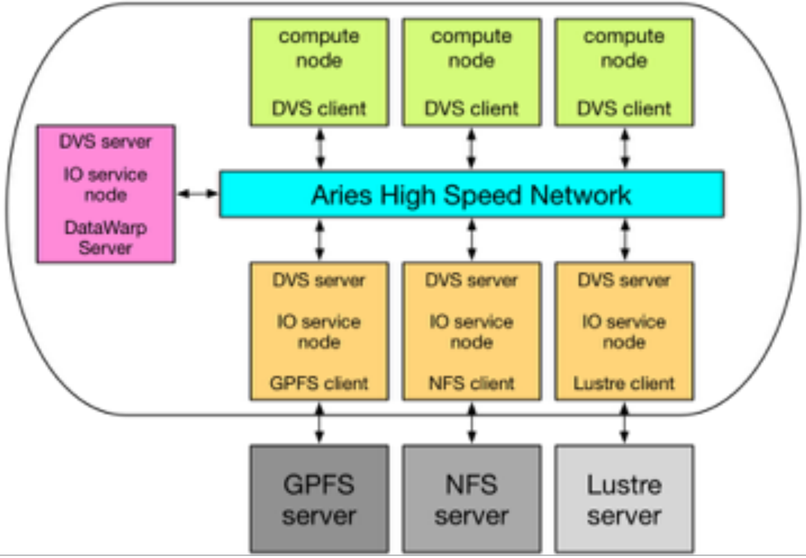
- **Filesystem performance, scale, and I/O bandwidth are not increasing at the same rate**
- **I/O is increasingly becoming the bottleneck to application performance**
- **New technologies are having to be adopted to bridge this gap**
  - Burst Buffers – Datawarp
  - I/O forwarders

Cray DVS

- **Cray Data Virtualization service**
- **I/O Forwarder**
  - Via HSN
- **Transparent Access**
  - Filesystems
  - Datawarp Accelerator
- **In-kernel – high performance**
- **Highly Scalable**
- **Tunable**

Motivation

- **DVS can drive I/O at network bandwidth**
  - Maintain I/O throughput over the network
- **Similar performance concerns of distributed parallel filesystems**
  - Network Latency
    - Increased for Subset of I/O
      - Small
      - random
      - repeating
    - Disproportionate cost



DVS Client-Side Caching

- **New option for DVS**
  - Available in CLE 6.0UP04 release
- **Mitigate Potential Bandwidth issues with I/O subset**
  - Improves I/O bandwidth
  - Reduce network latency costs
  - Lowers overall network traffic
  - Decreased load
    - DVS servers
    - Backing filesystems
    - Storage

Interfaces

- **Existing DVS 'cache' mount command option**
  - Now provides 'w' write option instead of only readonly 'ro'
    - mount –t dvs –o rw,cache…
    - /pfs /dvsmnt dvs rw,cache…
- ***DVS_CACHE* environment variable and IOCTL commands**
  - *Application control of caching as necessary*
  - DVS_CACHE=on / DVS_CACHE=off
- ***Cray Datawarp WLM job scripts***

Design

- **Implemented as a write-back type of cache**
- **Application writes target local in-memory cache**
  - Low latency & high throughput writes
  - Aggregation of data to be written back to servers
    - More optimal amount of data to be written
    - Lower number of total network transactions
  - Local caching of data read or written
- **Close-to-open coherency**
- **Cache page write-back heuristics**

Linux VFS Address Space

- **Utilizes Linux kernel page cache**
  - Kernel maintains memory utilization
  - Cache control interfaces
- **Local DVS filesystem *address_space*_operations**
  - *write_begin()*
  - *write_end()*
  - *writepage() & writepages()*

Close-to-open Coherency

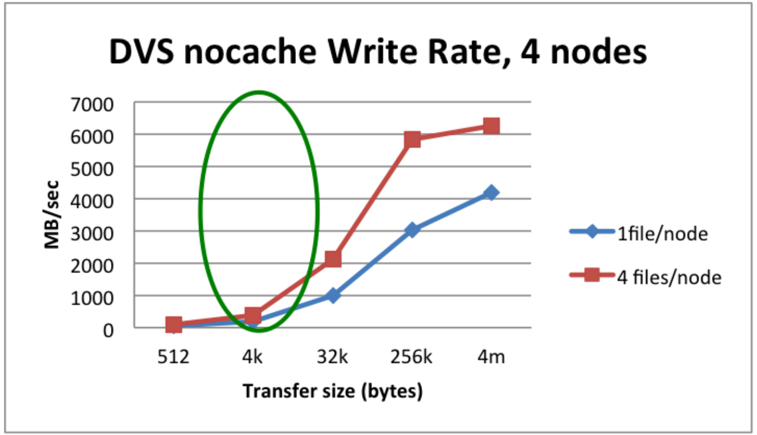
- Similar model as used by NFS
- Reads only guaranteed to see file data available on the server at file open time
- File write data cached not guaranteed to be written back to storage until file close time
- Does not imply newer data won't be read or written back
- 4kb kernel page size granularity

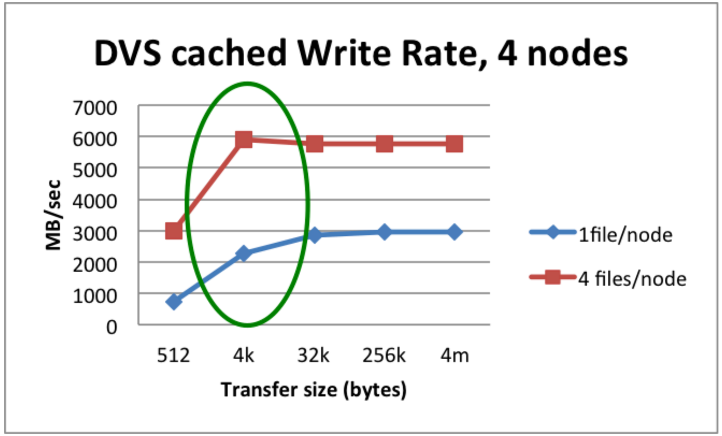
DVS Inode Attribute Handling

- **Possible inode attributes and cached data more current on client than server**
  - Change from existing DVS model
  - Prevent local client inodes from picking up stale server attributes
- **Metadata operations handled normally**
- **Writes and implicit size changes take effect at page writeback**

Results

- **Increased bandwidth for small file I/O**
  - 10x IOR increase
  - 100x IOPERF increase
  - TOPNET – HDF5 – 664 seconds to 114 seconds
  - Customer TOPNET benchmark – 58 to 34 minutes
  - Nastran to DataWarp – 9:55 to 6:51





Summary

- DVS client-side cache mitigates a potential downside of a network I/O forwarder
- Provides a new tier of file data storage in local high-speed memory on compute nodes
- Optimized writeback of aggregated data decreases file system access latency and network and server load
- Benchmark testing show bandwidth increases of 100x

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

Q&A

Bryce Hicks
bryceh@cray.com