

# Cray DataWarp: Administration and SLURM integration

*Tina M. Declerck, Iwona Sakrejda*

NERSC

Lawrence Berkeley National Lab

Berkeley, CA USA

TMDclerck@lbl.gov, ISakrejda@lbl.gov

*Dave Henseler*

Principal Engineer

Cray, Inc.

St. Paul, MN USA

dah@cray.com

*Abstract*—The National Energy Research Scientific Computing center (NERSC) is one of the Department of Energy’s (DOE) primary high performance computing centers supporting a wide range of research. One of the areas that large compute centers have worked to find a solution is the ability to efficiently move data to and from compute nodes. Cray is addressing this with their DataWarp technology. As new technologies are being developed and deployed, new tools are needed to address administration and troubleshooting. NERSC is collaborating with Cray to develop capabilities required to provide DataWarp functionality for our user base. In addition, integration into the workload manager is necessary to allow for effective resource utilization. This need is addressed jointly by NERSC in collaboration with SchedMD to implement key features that will facilitate integration of Cray’s DataWarp solution with SLURM. This paper will concentrate on the DataWarp administrative interface and the integration with SLURM covering the following areas.

- NERSCs DataWarp Requirements
  - Use Cases
  - Hardware considerations
  - Basic administration functionality
    - Configuration tools and options
    - Query hardware status
  - Monitoring
    - Identification of monitoring needs
    - Provide input for additional capabilities
  - Troubleshooting
    - Review of available tools
    - Inventory of special needs
  - Workload manager administrative capabilities
    - DataWarp instance management
- SLURM integration
  - Description of the requirements
  - Implementation details
  - Details of the tests used to evaluate the features
  - Work under development
- Test system hardware description

- Configurations tested

*Keywords*—DataWarp; SLURM; workload manager; SSD

## I. INTRODUCTION

The National Energy Research Scientific Computing Center (NERSC), located at Lawrence Berkeley National Laboratory (LBNL), is the production scientific computing facility for the United States Department of Energy (DOE) Office of Science. NERSC provides computational resources to support the scientific activities of over 5000 users worldwide, resulting in huge demands for resources and storage. As systems move toward exascale size and jobs become larger, filesystems are unable to perform at the rates required for many of the jobs running even today without extensive over provisioning which is not cost effective. In addition as the number of cores per CPU increase the amount of memory available to each is decreasing dramatically making it critical to find ways to get data to nodes more quickly to keep from starving the processors. Cray’s DataWarp solution has promise in providing this capability. This paper will describe design, initial implementation, first results of Cray’s DataWarp solution based on solid state disks (SSDs), their expected use at our facility, and details on the administrative requirements needed to use and maintain Cray’s DataWarp.

## II. BACKGROUND

What is Cray’s DataWarp? In very general terms it is Solid state disks (SSD) installed in the Cray XC40. The key value is the software layer Cray is developing and the integration with the workload managers (WLM) to allow access by user jobs in several different ways to work with different applications.

Why use Cray’s DataWarp solution? SSDs can be faster than a parallel file system especially with random access. Since the DataWarp nodes are blades in the system, they are more local to the nodes since the only network that it needs to use is Cray’s internal Aries network. In addition, depending on use it will have less contention than the file systems since they will be available only to the jobs requesting them. Random I/Os are also handled more efficiently.

The DataWarp service dynamically configures access to DataWarp instances on all compute nodes assigned to a job requesting it. Application I/O is forwarded from compute nodes to the instance on the DataWarp server nodes over the Aries network using the Cray Data Virtualization Service (DVS), which provides Posix based file system access to the DataWarp storage.

Cray's DataWarp technology is new and some features are still in progress. Since NERSC users would benefit from features not yet available, NERSC entered into an NRE project with Cray to help develop some of these. The timeline for this project only allows for some of the requested features to be available at this time. Much of this paper will provide information about the requested features and as much as possible about the expectations on how they will work. Since these are not yet available, the final product may vary from the descriptions provided in this document.

Although not all jobs will benefit from use of Cray's DataWarp, many should see improved performance. Use cases provided by NERSC and Alliance for Computing at Extreme Scale (ACES) made up of Los Alamos and Sandia National Laboratories helped guide the development by Cray. Each of these will be discussed in detail.

The basic use cases as well as general SSD capabilities and features provide the background needed to understand some of the requirements.

### III. USE CASES

#### A. Checkpoint / Restart

One concern with large jobs is losing hours of processing because of a failed component, node crash, or system failure. Checkpoint / restart is one of the means applications use to guard against these failures. It allows a job to pick up and run from its last checkpoint. The problem with this currently is that hours of compute cycles are used doing I/O for checkpointing instead of running the jobs. Cray's DataWarp will allow jobs to be checkpointed more quickly and then asynchronously transfer the checkpoint to the parallel file system allowing the job to continue without having to wait for the write to complete to the PFS. This does leave a window of time that the job wouldn't be able to recover from a full system failure using that checkpoint, the previous checkpoint would still be available. Most other failure modes are recoverable. In addition, the checkpoint can remain on the DataWarp storage until the next checkpoint is ready to be written so recovery can be achieved immediately if necessary without having to wait for the checkpoint to be re-staged to the DataWarp from the PFS.

#### B. Pre and post staging for files

Some jobs require access to large files when they start. The time required to pull the files from the PFS increases the startup time for the job. Cray's DataWarp can be used to pre-stage files allowing the system to have better utilization since this can be accomplished prior to node assignment. In addition, writing data to the DataWarp prior to sending to the PFS or for intermediate data will allow jobs to continue to

process while the DataWarp writes the files to the PFS asynchronously.

#### C. Ability to stripe across multiple DataWarp nodes

Cray's DataWarp will have the ability to stripe across multiple SSDs for either cache or scratch usage. This increases the space available and the speed at which data can be read or written.

#### D. Compound jobs

It is not uncommon for a user or a group of users to access the same file for multiple different calculations. Cray's DataWarp will have the ability to have a persistent that will allow access to a file or a group of files by multiple jobs. These are not limited, so any user can access the instance. File permissions are used to restrict access to data. In this context the benefit is similar to the pre-staging but multiple jobs can take advantage. Also, the lifetime of a persistent instance is not related to any single job.

#### E. Implicit cache

Cache allows data to be stored in a temporary location, usually RAM, for faster access. In this case, the cache is an SSD that acts as an intermediary storage between RAM and disk. Although not as close as system memory it should allow faster access to data. Data that is stored in this fashion will be asynchronously synced to disk.

#### F. Private cache and scratch for swap and tmp usage

In private mode the DataWarp is configured as if it were a local swap or tmp. Only the node creating data has access to the data.

### IV. DATAWARP CONFIGURATIONS

Cray's DataWarp is intended to be a schedulable resource that can be utilized in two different ways. It can be scheduled as part of a job – a job instance, or as a persistent instance that can be accessed by many jobs and possibly users.

A DataWarp instance can be configured as one of 2 types:

- **Scratch:** When using a scratch instance all movement of data between the instance and the PFS must be explicitly requested using the DataWarp job script staging commands or the application C library API.
- **Cached:** When using a cached instance all movement of data between the instance and the PFS is done implicitly. The instance configuration and optional API can change the implicit behavior.

A scratch or cached configuration can be accessed in one or more of the following ways:

- **Striped:** In striped access mode individual files are striped across multiple DataWarp server nodes (aggregating both capacity and bandwidth *per file*) and are accessible by all compute nodes using the instance.
- **Private:** In private access mode individual files reside on one DataWarp server node (aggregating both

capacity and bandwidth *per instance*). For scratch instances the files are only accessible to the compute node that created them (e.g. like /tmp); for cached instances the files are accessible to all compute nodes (via the underlying PFS namespace) but there is no coherency between compute nodes or between compute nodes and the PFS. Private access is not supported for persistent instances because a persistent instance can be used by multiple jobs with different numbers of compute nodes.

- **Loadbalanced:** In loadbalanced access mode individual files are replicated (read only) on multiple DataWarp server nodes (aggregating bandwidth but not capacity *per instance*) and compute nodes choose one of the replicas to use. Loadbalanced mode is useful when the files are not large enough to stripe across a sufficient number of nodes.

There is a separate file namespace for every instance (job and persistent, type (scratch and cached) and access mode (striped, private, loadbalanced) except persistent/private is not supported. The file path prefix for each is provided to the job via environment variables.

Within a scratch instance, the directory structure (and associated directory overhead) for the instance is managed differently depending on the access mode: for striped access one of the DataWarp servers manages the directory for the instance and for private and loadbalanced access each DataWarp server manages the directories for files assigned to it. Within a cached instance, the directory structure resides on the PFS itself and every DataWarp server can access it directly.

Every DataWarp server is responsible for PFS I/O for the files and stripes that reside on that server. This provides parallel access to the PFS but from fewer nodes than if directly accessed from compute nodes. For a scratch instance, PFS I/O is explicitly initiated by the job or application and transfers an entire file to or from the DataWarp server nodes it resides on. For a cached instance, PFS I/O is implicitly initiated by the DataWarp service based on modified data thresholds, the DataWarp file system becoming full or reads of data not currently present. The DataWarp service manages space using least recently used tracking and provides read ahead and write behind functionality such that the file system never becomes full as it is always able to flush modified data to or reclaim data from the PFS.

## V. SSD CONSIDERATIONS

Although there are many benefits to using SSD technology, it is not without its own considerations. System administrators will have to be aware of the failure modes and ensure the health of the hardware. One of the things to be aware of is that SSDs are a consumable resource -- they wear out over time. The length of their life is described as their endurance. This is usually described using the number of "drive (or diskful) writes per day" (DWPD) for a period of time (usually 5 years). This varies for different SSDs. SSD cards have spare blocks that are used to replace blocks that are worn out. In general, the more spares that are available the more drive writes per day the SSD can handle. A drive write

per day is defined as the number of writes it takes to fill the SSD. For instance, if it provides three drive writes per day and the SSD is a 400 GB device it should be able to have 1200 GB of data written per day for the designated life. If the life is 5 years, 2,190,000 GB is the total amount of data the SSD can have written before it is no longer operable.

$$\text{DWPD} * \text{Device size} * \text{life in years} * \text{days per year} = \text{Total} \\ ((3 * 400) * (5 * 365)) = 2,190,000 \text{ GB}$$

It is possible for a job to wear out an SSD in a very short period of time if large amounts of data are written.

Wear-leveling balances the use of the blocks on an SSD to ensure blocks are getting used evenly. This can cause a shorter life because there aren't enough available blocks to use although some blocks may have several more writes available. Two types of wear-leveling are generally in use. Dynamic wear-leveling ensures new writes or re-writes are written to new areas on the SSD. However, if there is static data those blocks may still be active when the SSD is no longer operable. Static wear-leveling uses the same method as dynamic wear-leveling for dynamic files but also relocates static files occasionally to free those blocks for additional writes. (1)

The TRIM command allows the operating system to identify which blocks are no longer considered in use so they can be deleted. One of the differences between a disk and an SSD is how deletes and format is done. Although these can happen on a disk without affecting performance, on an SSD the performance can be significantly affected. The TRIM command allows garbage collection to occur without affecting performance.

Another factor is performance. The ability to access the DataWarp must be faster than accessing the parallel file system or it loses much of its advantage, as seen in the use cases. The ability to stripe across multiple DataWarp nodes will allow for higher performance.

## VI. ADMINISTRATIVE FEATURES

The administrative features focus in two primary areas -- hardware status and monitoring, and workload manager related features. Hardware status and monitoring include things such as error reporting, wear monitoring, and software and firmware updates. The workload manager features are more extensive and include ways to control user instances, taking a DataWarp node offline with the ability to move data, update access permissions, etc.

### A. Status & Monitoring

In most ways, the DataWarp hardware is the same as other hardware in the system. There are hardware problems that will result in the node not being available. Monitoring for these hardware errors is effectively the same as for other hardware in the Cray and no additional action is needed. However, since this hardware is able to be used in jobs and can be reserved for specific jobs, information regarding unavailable DataWarp nodes must be passed to the workload manager. The workload manager aspects of the DataWarp will be covered in the next section.

Since the DataWarp is a consumable resource, it is important to be able to both monitor the available life of an SSD and ensure a poorly executed code can't consume the life of an SSD in a short period of time. This information should be accessible via command line or API so sites will be able to utilize local tools for monitoring. In addition, if there is an SSD related health event information should be logged to the console or system log. The ability to monitor this data should be available while the system is active.

Other capabilities include the ability to query the firmware levels of the SSD, bit error rates, and general health.

### *B. Workload manager*

Many of the requirements from an administrative perspective are related to the workload manager. Interactions with the DataWarp are instantiated through directives in the WLM job scripts. So tools are needed within the WLM to debug issues with DataWarp similar to how issues are found and resolved with jobs now.

One of the most basic functions an administrator will need is the ability to query the DataWarp instances both job and persistent. The information needed will include who requested the instance, where the instance is located (which DataWarp nodes and the instance name), size and duration, instance parameters, start time, and owning job (if applicable).

Similar to needing the ability to kill jobs for various reasons, an administrator may need to kill an existing job or persistent DataWarp instance. One difference is that data may exist on the DataWarp instance and additionally need the ability to drain the data prior to removing the instance especially in the case of a persistent instance. However, there may also be cases where purging the data is a better option. It is preferable if the administrator can choose to either purge or drain the DataWarp instance.

In a similar vein, it may be necessary to restrict access by a specific user or group or to allow access to specific users or groups. If the use of the DataWarp is very successful it is likely there will also need to be a way to ensure fair use by limiting the size and/or duration of DataWarp instances.

The DataWarp nodes may need to be removed from service for hardware or for other activity. An administrator will need to be able to initiate draining of nodes and keep them getting allocated to new jobs. Existing job instances should be able to complete. Persistent instance will need to be migrated to other DataWarp nodes or the data written to permanent storage.

One frequent question from users is "Why isn't my job starting?" There can be many reasons for this. Use of DataWarp is one more thing that will need to be checked. Seeing the status of getting access to a DataWarp instance from the workload managers perspective will be critical in diagnosing a problem that is keeping jobs from running or causing them to die prematurely.

Although at our site it is rare, on occasion the need to alter a job as an administrator is needed. A DataWarp instance may also need to be altered. In the queued state a user should be able to alter some of their settings but once a job is running

administrative intervention is usually required. The ability to change the duration, size (if possible), add or change user access, and modify certain instance parameters are useful features.

In order to ensure the best use of Cray's DataWarp, jobs should be distributed as evenly as possible so they maintain a similar level of wear across the SSDs.

Finally, the ability to get DataWarp usage statistics at the end of each job including bytes in / out preferably per server.

### *C. SLURM Integration*

In order to improve effective use of DataWarp and enhance user experience, NERSC signed a contract with SchedMD for development of additional capabilities for the SLURM workload management software.

This effort will provide new features for Cray DataWarp scheduling and management using Slurm in the following areas.

#### *1) Job Prioritization*

Job priorities are calculated by a plugin and the most commonly used is priority/multifactor, which computes a priority based upon job age, size, fair-share (how over- or under-serviced the user is), Quality Of Service (QOS) and queue. Each of these factors is given a weight in the priority calculation. SchedMD will modify the multifactor plugin so that DataWarp space will be included as an additional factor in job priority calculation, associated changes will also be made to the Slurm configuration file, and sprio command.

#### *2) Advanced Reservations*

Slurm has the ability to create an advanced reservation of resources including compute nodes and licenses for specific users or groups. This mechanism insure that sufficient resources are available for work deemed critical. SchedMD will add the ability to reserve DataWarp space. Development work will include changes to the scontrol command (to set and view the limits), the reservation management module, and the DataWarp plugins, and associated documentation.

#### *3) Resource Limits*

Slurm supports a wide variety of resource limits enforced by user, account, or Quality Of Service (QOS). Separate limits can be configured on each of these entities both on a per job basis and an aggregate basis for all jobs associated with a specific user, account or QOS. SchedMD will add Slurm support for configurable per-job and aggregate DataWarp space limits. These limits will be implemented for both the QOS and association entities. Per job limits will support both a fixed size and proportional limits (e.g. maximum of X GB of DataWarp space for each allocated compute node). Slurm development work will include changes to database tables, various remote procedure calls (RPCs), the sacctmgr command (to set and view the limits), the DataWarp plugin (to provide enforcement), and associated documentation.

SchedMD developers obtained access to Alva where the production release is available and the work commenced. This work will be incorporated into the Slurm product and be generally available by the end of Summer 2015 with the beta

test available ~beginning of August and incorporated in Slurm version 15.08 upon release.

More information about SLURM support for the priority multifactor is available online at

[http://slurm.schedmd.com/priority\\_multifactor.html](http://slurm.schedmd.com/priority_multifactor.html)

More information about SLURM support for advanced reservations is available online at

<http://slurm.schedmd.com/reservations.html>

More information about SLURM resource limits is available online at

[http://slurm.schedmd.com/resource\\_limits.html](http://slurm.schedmd.com/resource_limits.html)

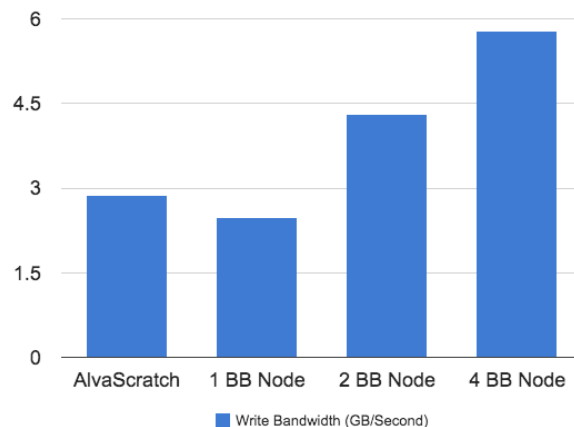
## VII. TEST DATA

### A. Hardware Description

The test and development system (TDS), Alva, has been used to test the initial DataWarp software. It contains 3 DataWarp blades for a total of 6 nodes and 12 SSDs. The SSD cards connect to the x8 PCIe gen3 slots. The service nodes contain one 2.6GHz 8-core Sandy Bridge processor with 32 GB of DDR3 memory. The SSDs are Fusion SX300-1600 SSD cards. They have a 1.6 TB capacity. Cray has not provided information regarding the production DataWarp SSDs so the Fusion SSDs may or may not be used in the production DataWarp nodes.

### B. Checkpointing Evaluation

One of the early tests on Alva evaluation testbed included benchmarking of checkpointing. We evaluated the benefit of DataWarp storage for checkpointing in various configurations and compared it with performance of lustre based scratch for an Adaptive Mesh Refinement (AMR) based simulation such as NyX. Results are shown in Fig. x



## VIII. CONCLUSION AND FUTURE WORK

The DataWarp design offers some interesting methods to allow more efficient data operations on the Cray XC40. Although this is a work in progress the technology and implementation look promising. We plan to continue to work with Cray to develop features that will allow our user base to utilize the DataWarp technology. This paper discussed the work that will be coming to test the features and functionality defined and identify any additional features that may be beneficial in using Cray's DataWarp technology.

### ACKNOWLEDGMENT

This work was supported by the Director, Office of Science, Office of Advance Scientific Computing Research of the U.S. Department of Energy under contract No. DEAC02-05CH11231.

### REFERENCES

- [1] [http://en.wikipedia.org/wiki/Wear\\_leveling](http://en.wikipedia.org/wiki/Wear_leveling)
- [2] "Fujitsu PRIMERGY Servers Solid State Drives – FAQ", version 1.1 2014-03-31, <http://globalsp.ts.fujitsu.com/dmsp/Publications/public/wp-solid-state-drives-faq-ww-en.pdf>