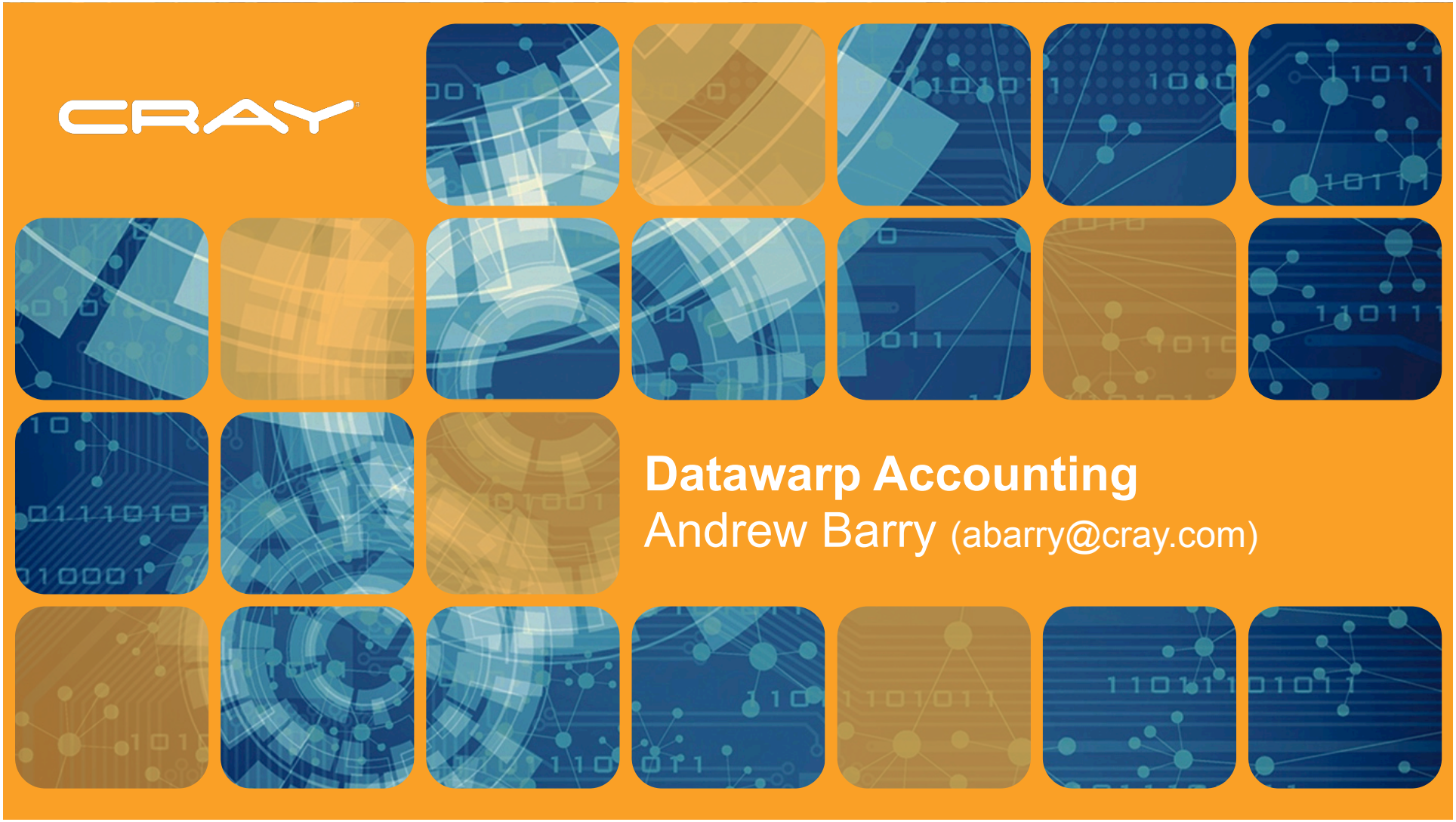


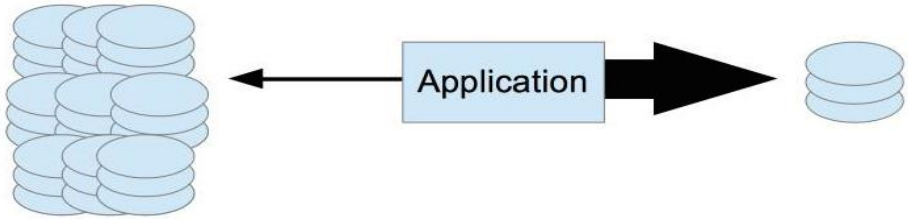
Datawarp Accounting

Andrew Barry (abarry@cray.com)

Agenda – Datawarp Accounting

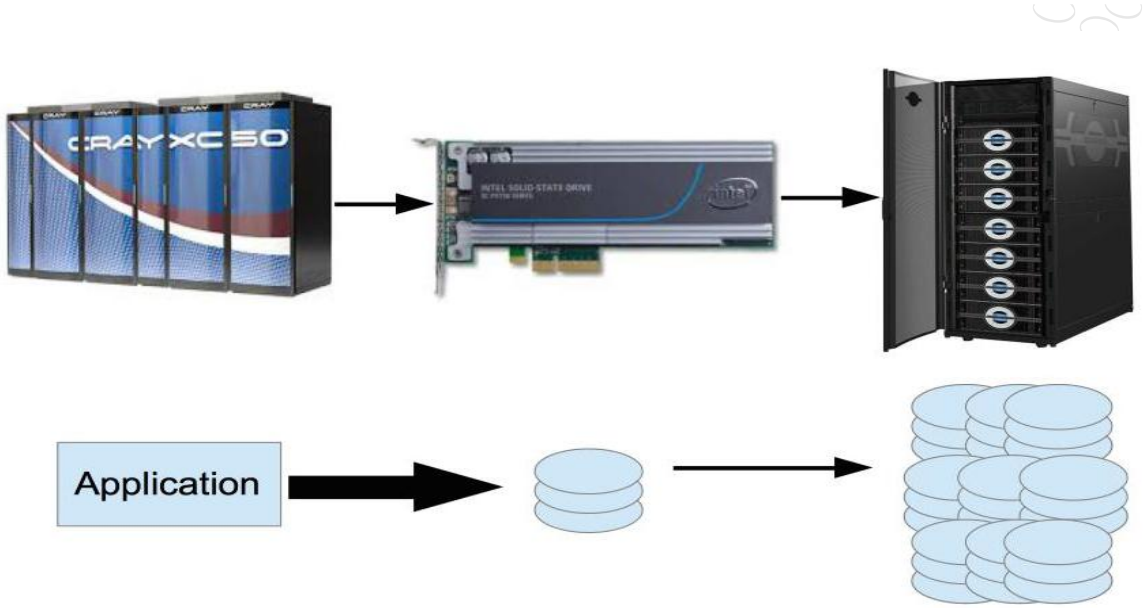
- **Purpose**
  - Give Administrators tools they need to manage Datawarp resources
- **Benefit/Value**
  - Administrators can bill users, educate users on best practices, and plan for the future.
- **Datawarp statistics**
- **Resource Utilization Reporting plugins for Datawarp**
- **Libdatawarp**
- **Case studies of Data and interpretation**
- **Summary**
- **Q&A**

Datawarp



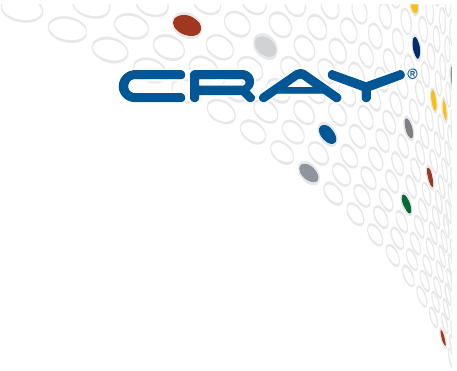
- Burst Buffers on Flash drives
- Versus spinning disks: high bandwidth, lower capacity
- Limited write cycles per device

Datawarp Cache



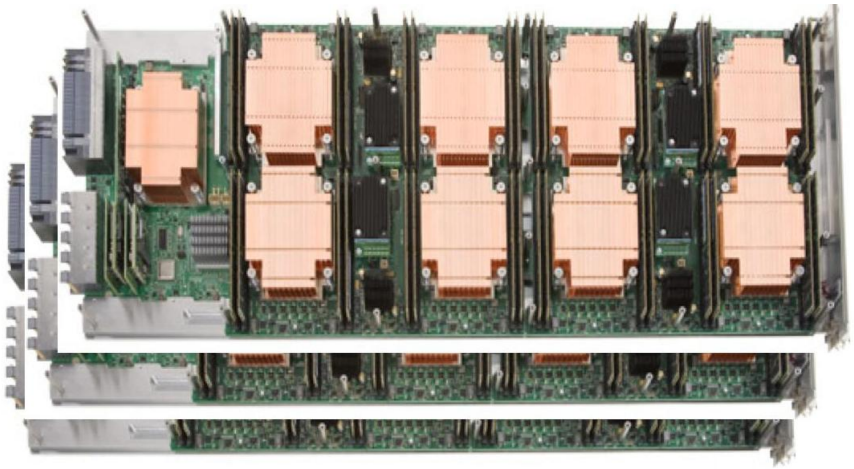
- Flash acts as a cache for Parallel filesystem
- Datawarp cache is dedicated to an user/application

What Data Do You Want?

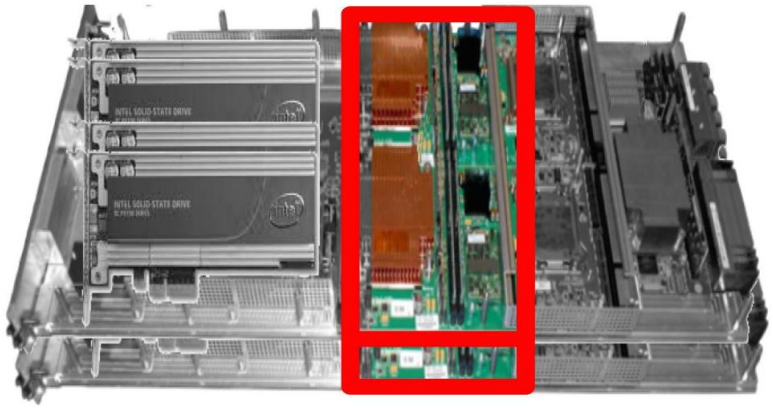


- Are the users getting good performance?
- Are users sharing well?
- Who is using up all the drive writes?
- Does the system have sufficient capacity?
- ??? Question unknown, but provide a lot of data

Nodes: compute, server



Compute Nodes



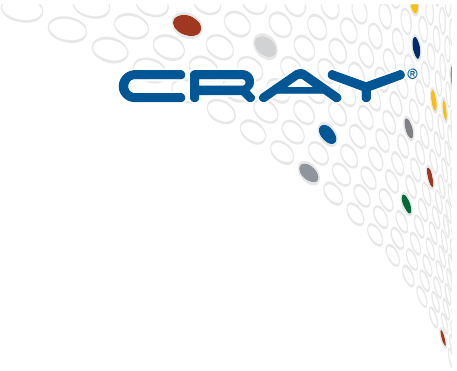
Datawarp Server Nodes

Some Stats Here

Datawarp Filesystem

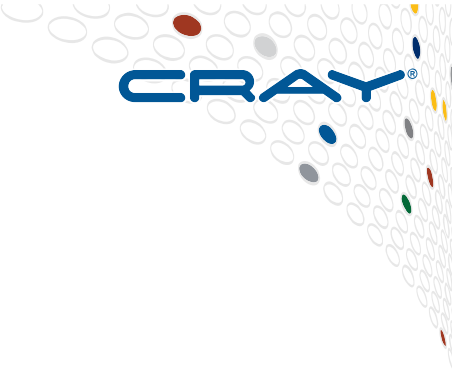
Fragments

Compute Node Datawarp Statistics



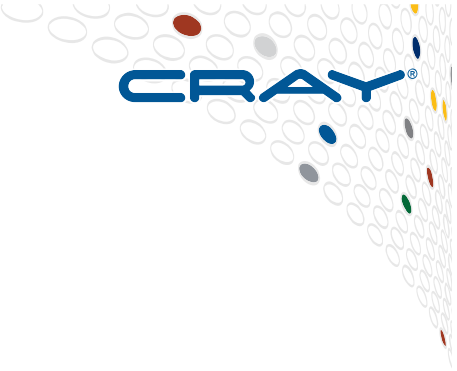
- Inodes Created
- Files Created
- Bytes Read
- Bytes Written
- Max File Offset Read
- Max File Offset Written

Server Node Statistics



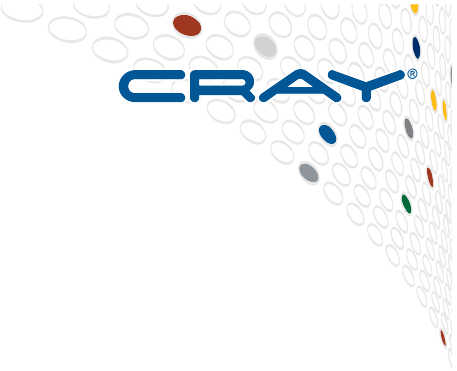
- Max offset read/written; total, per-fragment
- Bytes read/written
- Bytes Staged in/out
- Cache and pfs byte in/out
- Capacity highwater; total, per-fragment
- Maximum write window

How Do I Get the Accounting Data?



- **Resource Utilization Reporting (RUR)**
  - Lightweight, low noise
  - Simple configuration, installed as part of CLE
  - Not a performance profiling tool for software development
- **Libdatawarp library**
  - Provides user application on compute nodes access to accounting data
- **Slurm**
  - Will include same metrics as does RUR
  - Support in future version

RUR Datawarp Plugins

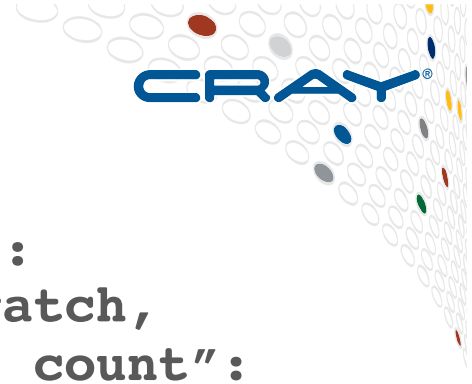


- **DWS plugin**
  - CLE6.0up02
  - DVS statistics from compute nodes
- **DWS_server plugin, DWS_job_server plugin**
  - CLE6.0up04
  - Statistics from server nodes
  - Per-namespace & per-fragment statistics
  - Job plugin captures statistics for stage-in and stage-out
- **RUR**
  - Output data to SMW logs, text files, user home dir, etc.
  - Configuration like other RUR plugins (see S-2393)

RUR Output

- Found in SMW log, or other files, as configured
- Includes apid, jobid, command name
- Includes user id
- Can be very verbose if there are a lot of Datawarp server nodes
- Next 3 slides are an example of the full output, but only one filesystem with only two fragments
- Later slides include only snippets for clarity

RUR Output Scratch Namespaces



```
Uid: 16443, apid: 123050, jobid: 2546, cmdname:
disk_test1, plugin: dws_server {["dwtype": scratch,
"realm_id": 657, "server_count": 2, "namespace_count":
1, "token": "2546.sdb", "Namespaces": {"217":
{"bytes_read": 167888204, "bytes_written": 404418,
"files_created": 16, "max_offset_read": 167888204,
"max_offset_written": 404418, "stage_bytes_read": 0,
"stage_bytes_written": 0, "files_create_threshold": 0,
"file_size_limit": 0, "stripe_size": 16777216,
"stripe_width": 4096, "substripe_size": 16777216,
"substripe_width": 4096}}
…
```

RUR Output Scratch Fragments


```
“fragments”: {
“3141”: {“fragment_id”: 3141, “server_name”: “nid00343”,
"fs_capacity": 8796093022208, ”capacity_used":
3518437208883, "capacity_max": 4398046511104,
"max_window_write": 1073741824, "write_high_water":
4294967296, "write_moving_avg": 536870912,
“write_limit”: 0},
“3142”: {“fragment_id”: 3142, “server_name”: “nid00344”,
"fs_capacity": 8796093022208, ”capacity_used":
3518437208883, "capacity_max": 4398046511104,
"max_window_write": 1073741824, "write_high_water":
4294967296, "write_moving_avg": 536870912, “write_limit”:
0}}]}
```

RUR Output Cache

```
“fragments”: {
“3144”: {“fragment_id”: 3141, “server_name”:
“nid00343”, “capacity_highwater”: 4398046511104,
“fs_capacity”: 4398046511104, “max_offset_read”:
167888204, “max_offset_write”: 167888204, “pfs_read”:
41972051, “pfs_written”: 41972051, “cache_read”:
335776408, “cache_write”: 335776408,
“window_write_bytes”: 20986024,
“window_write_seconds”: 8}}]}
```

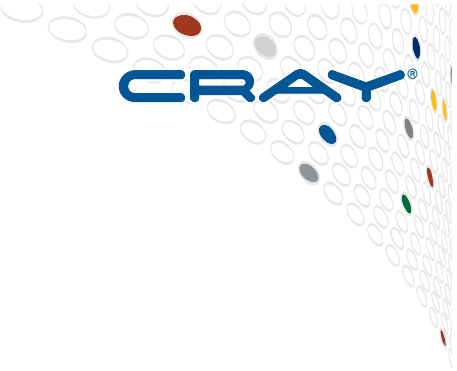
Libdatawarp

- Same data as dws_server plugin
- Users can do periodic sampling

```
#include <datawarp.h>
buf = dw_get_accounting_data_json(dwfs_path,
&retval);

#include <datawarp_cache.h>
buf = dwc_get_accounting_data_json(dwcfs_path,
&retval);
```

Datawarp Accounting Use Cases



- Tracking Disk Writes
- Over-allocating Capacity
- Excess Staging
- Dissimilar Stripe Allocation

Tracking Disk Writes

- Are users writing a LOT to disks, wearing them out?
- Sometimes it needs to happen, but be careful.
- Bill users for excessive writes?

```
Uid: 16443, apid: 24104… {‘Bytes_written’:
240TB…}
```

Over-allocating Capacity

- **Are user capacity estimates accurate?**

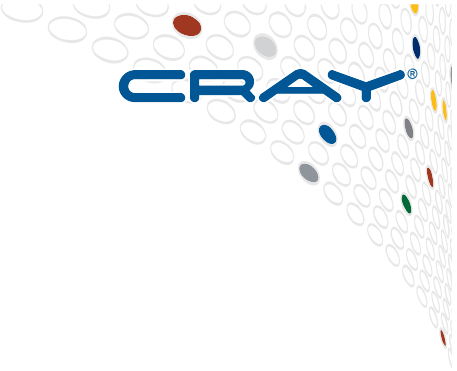
```
Uid: 16771, apid: 14654, …
{‘fs_capacity’: 32TB, ‘capacity_used’:
4.09GB…}
```

Excess Staging

- User preloads tons of data, then uses hardly any of it.
- Seems like a waste, though not always.

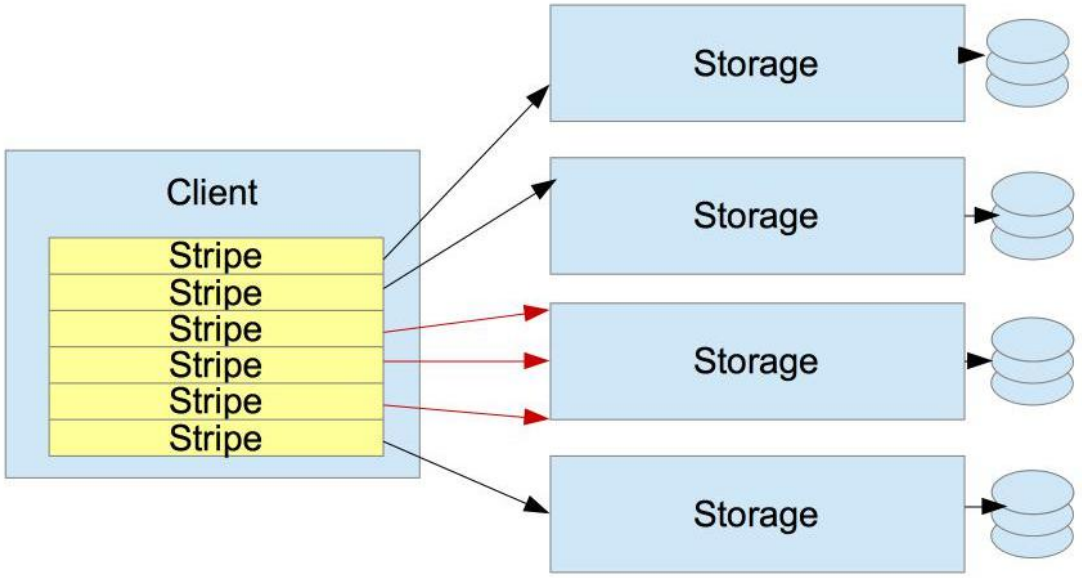
```
Uid: 16443, apid: 24186…
{‘bytes_read’: 2.3GB, ‘stage_bytes_read’:
415.2GB,
‘stage_bytes_written’: 0GB…}
```

Dissimilar Stripe Allocation



- User experienced large variability in Datawarp performance from one run to the next.
- Correlated with low available capacity, but not that low.
- Dissimilar stripe allocation
- Shows up in accounting data
- ‘Equalize Fragments’ configuration option to prevent this

Dissimilar Stripe Allocation 2



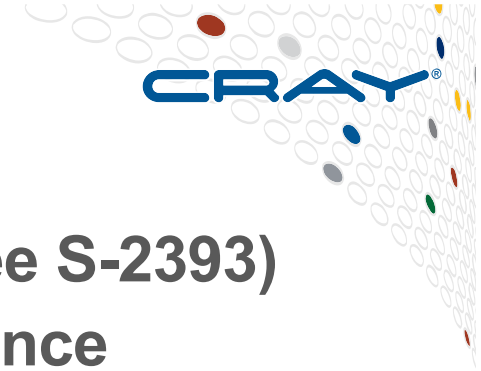
- Too much traffic going to a single storage node
- Due to fragmentation of free space

Dissimilar stripe Allocation 3

```
Dws_server{fragments:{
'Nid00346': {'capacity_max': 250GB},
'Nid00347': {'capacity_max': 250GB},
'Nid00221': {'capacity_max': 750GB},
'Nid00222': {'capacity_max': 250GB}}
```

Nid00221 is serving 3 times as much space, will be hit by 3 times as much I/O traffic

Equalize Fragments



- Admin config option to Datawarp Services (See S-2393)
- Off by default, but recommended for performance
- Causes allocations from different DW server nodes to be equal in size
- Balances performance of nodes, removes bottleneck
- May cause difficulty for WLM scheduling jobs

Dissimilar stripe Allocation 4

```
Dws_server{fragments:{
‘Nid00346’: {‘capacity_max’: 400GB},
‘Nid00347’: {‘capacity_max’: 400GB},
‘Nid00221’: {‘capacity_max’: 400GB},
‘Nid00222’: {‘capacity_max’: 400GB}}
```

Note that this overprovisions the allocation (1600GB allocated > 1500GB requested)
This overprovisioning preserves balanced free space

Summary

- Datawarp offers users very fast storage, but with limited capacity, and limited write endurance of devices.
- Tracking which users utilize Datawarp, and how, allows administrators to better coordinate the resources available to the users, and plan for the future.
- RUR plugins can give basic utilization data to admins, with low overhead. Libdatawarp gives the same data to users.

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.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, REVEAL, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.

Q&A, Feedback

Andrew Barry
abarry@cray.com

CUG.2017.CAFFEINATED COMPUTING

Redmond, Washington May 7-11, 2017