Data Storage and Data Analysis Workflows for Research

Minnesota Supercomputing Institute
February 26, 2019

https://z.umn.edu/44jn
Tutorial Outline

• Hardware overview
• Systems overview
• Options at UMN
• Options at MSI
  • Storage hierarchy
  • Interfaces for managing data
  • Performance issues
• Use Cases
• Hands on
Storage Technologies
Computer Memory Hierarchy

- CPU Registers
- Processor Cache
- Random Access Memory
- Flash or USB Memory
- Hard Drive Disks
- Tape Backups & Web

Capacity & Granularity
Performance & Cost/Byte
Storage Technologies
Computer Memory Hierarchy

Capacity & Granularity

Processor Cache

Random Access Memory

Flash or USB Memory

Hard Drive Disks

Tape Backups & Web

Performance & Cost/Byte

© 2009 Regents of the University of Minnesota. All rights reserved.
Storage Technologies
Hardware

• Hard Disk Drives (HDDs)
  • Rotating rigid platters on a motor-driven spindle within a protective enclosure. Data is magnetically read from and written to the platter by heads that float on a film of air above the platter.

• SATA -- Serial Advanced Technology Attachment
  • Desktop
  • Low cost
  • up to 8 TB
  • ~ 6 Gb/s
  • ~1.2 million hours MTBF
    • 8hrs/day out of 1000 drives 1 will fail every 150 days

• SAS -- Serial Attached SCSI
  • Enterprise use
  • Costly
  • up to 8 TB
  • ~ 12 Gb/s
  • ~1.2 to 1.6 million hours MTBF
Storage Technologies
Hardware

• Solid State Drives (SSDs)
  • Use microchips which retain data in non-volatile memory chips.
  • No moving parts
  • less susceptible to physical shock
  • silent
  • very low access time
  • very expensive (Compared to HDDs)
  • MTBF ~1.5 million hours

• Hybrid HDD and SSD drives (SSHD)
  • SSDs add speed to cost effective media by acting as Cache
Storage Technologies
Hardware

• RAM Disk
  • Block of random-access memory (primary storage or volatile memory) that a computer's software is treating as if the memory were a disk drive (secondary storage).
  • Used to accelerate processing
  • No moving parts
  • Very low access time (Compared to HDDs and SDDs)
  • Very expensive (Compared to HDDs and SDDs)
  • Data lost when powered off or rebooted
Storage Technologies
Future of Storage

- Better conventional HHDs
  - Helium Filled
  - Shingled Magnetic recording (SMR)
  - Heat-assisted magnetic recording (HAMR)
- Better/Cheaper Solid State solutions?
  - Next-gen Phase Change Memory (PCM)
  - Could flatten complex data hierarchies?
- DNA digital data storage for archive storage
  - Very slow but extremely dense
Storage Technologies

How do we use these devices?

**Devices**
- Memory
- Block
- Arrays of Disks

**Filesystems**
- Disk File Systems
  - Ext4, ZFS
- Network File Systems
  - NFS, SMB
- Parallel File Systems
  - Panasas, Lustre, GPFS
- Special Cases
  - FUSE
    (Filesystem in Userspace)
  - CephFS

**Services**
- Cloud
  - Google drive, Dropbox, Amazon (S3)
- Databases
  - MySQL, CouchDB
# Storage Technologies

## Order of Magnitude Guide *

<table>
<thead>
<tr>
<th>Storage</th>
<th>Files/dir</th>
<th>File sizes</th>
<th>Band Width</th>
<th>IOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local HDD</td>
<td>1,000s</td>
<td>GB</td>
<td>100 MB/s</td>
<td>100</td>
</tr>
<tr>
<td>Local SSD</td>
<td>1,000s</td>
<td>GB</td>
<td>1 GB/s</td>
<td>10,000+</td>
</tr>
<tr>
<td>RAM FS</td>
<td>10,000s</td>
<td>GB</td>
<td>10 GB/s</td>
<td>10,000</td>
</tr>
<tr>
<td>NFS</td>
<td>100s</td>
<td>GB</td>
<td>100 MB/s</td>
<td>100</td>
</tr>
<tr>
<td>Lustre/GPFS</td>
<td>100s</td>
<td>TB</td>
<td>100 GB/s</td>
<td>1,000</td>
</tr>
<tr>
<td>Cloud</td>
<td>Infinite</td>
<td>TB</td>
<td>10 GB/s</td>
<td>0</td>
</tr>
<tr>
<td>DB</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>1,000</td>
</tr>
</tbody>
</table>

*From SDSC 2015 Summer institute: HPC and Long Tail of Science*
Storage Technologies
Data Redundancy

- **Mirroring**
  - Create identical copies of files

- **RAID (Redundant Array of Independent Disks)**
  - Multiple disks pooled into a single logical unit
  - RAID with N=2 is Mirroring
  - Larger disk pools (N>2) can save storage
  - Uses a parity to recreate missing data when drive is lost

- **Snapshot**
  - Creates a copy of the current state of the system to disk
  - Very fast, doesn’t delay subsequent writes.

- **Tape backup**
  - Refers to the media, portable
  - Typically less expensive
  - Offline for Disaster recovery purposes.
Storage Options at UMN

**Department**
- Workstation
- Departmental Servers

**OIT**
- Google Drive
- Isilon
- Block Storage

**MSI**
- Panasas
- Tier-2 CEPH
- Tier-3 Tape

**Library**
- DRUM, Data Repository for the U of M

**You**
- Laptop
- Mobile
<table>
<thead>
<tr>
<th>Purpose</th>
<th>Google Drive</th>
<th>OIT Isilon</th>
<th>OIT Block</th>
<th>MSI Panasas</th>
<th>MSI Tier-2</th>
<th>Dept Storage</th>
<th>Laptop/Desktop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big Data</td>
<td>✓ ?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share access</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Archival (very long-term) storage</td>
<td>✓</td>
<td>✓</td>
<td>✓ ?</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access on Campus Laptop/Desktop</td>
<td>✓ ?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legally protected data (Coming)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2009 Regents of the University of Minnesota. All rights reserved.
Gopher Science Network at UMN

UMN Network with Proposed Gopher Science Network (GSN)

Campus centers are currently severely constrained by 1 GbE connectivity.

St. Paul Campus
OpenCDN Group (Keller Hall)
MPC
UMGC CMRR
CMSP
Synder Hall

100 GbE to POP (internet 2)

MSI Core
Mesabi
Itasca
Panasas
Big Data Ceph

MSI provides a variety of research platforms for researchers and manages an internal 10/40/100 GbE network.

© 2009 Regents of the University of Minnesota. All rights reserved.
Ask Questions First

Not all data is created equal

- What do I want to do with the data?
  - How large are the files I’m storing?
  - How many files will I store?
  - How frequently will I access the data?
  - From what locations will I access the data?
  - In what format will the data be stored?
Storage at MSI
Storage Strategies
A Collaborative Effort

• You have data & real world needs.
• MSI has hardware, software, & expertise.

IF your data needs are vast
( huge, complex, compute intensive, … )
THEN MSI can help.

• Enabling HPC workflows is what MSI is about
• We are all in this together.
Store and Stage Data

What’s available at MSI:

- Shared file system: PanFS
- 2nd Tier Storage: CEPH
- 3rd Tier Storage: Tape
- Databases: Web servers
- Local Disk
- RAM disk
Shared File system

**What it is**

PanFS: Block storage; POSIX
Visible on all MSI systems
Persistence: duration of your account at MSI

**How you access it:**

Directories: home, shared, public, scratch
Shell commands: cp, mv, rm, grep, …
Applications: all POSIX file IO
Shared File system

Locations & Uses

- `/home/<group>/<user>`: Your private files
- `/home/<group>/shared`: Share with your group
- `/home/<group>/public`: shared with all MSI
- `/scratch.global`: Temp. files for multiple hosts

Limits

- `/home/<group>/*`: group quota (allocation)
- `/scratch.global`: 1 month lifetime & SLOW!
2nd Tier Storage

**What it is**

- CEPH: Object storage; S3
- Visible on all MSI systems and Web
- Persistence: duration of allocation

**How you access it:**

- By file only
- Files organized in “buckets”
- Shell: s3cmd
- Web URL & GLOBUS

[https://www.msi.umn.edu/content/second-tier-storage](https://www.msi.umn.edu/content/second-tier-storage)
CEPH: S3 interface

**Locations & Uses**

s3://<bucket name>/<file name>
s3cmd commands: ls; get; put
Save & stage large volumes of data

**Limits**

CEPH write access by user allocation
CEPH read access can be granted by user

© 2009 Regents of the University of Minnesota. All rights reserved.
3rd Tier Storage: Tape

**What it is**
- Blackpearl: LTO-7 tape (6 - 15 TB per tape)
- Visible: MSI HPC systems
- Persistence: ~5 years
- This is a service: NOT just tapes

**How you access it:**
- Purchase: $456 per “unit” (= 1 redundant pair of tapes)
- Large files: 1-1000 GB (approx)
- Latency: 1-7 days to recover data (approx)
- For more info: send email to help@msi,umn.edu
Databases & Web Services

What it is

Database services & servers managed by MSI
Visible worldwide on hosts with web access
Persistence: lifetime of project

How you access it:

Web URL
Shell: wget or database clients
Get access through a coordinated MSI project
Databases

Locations & uses
URL: www.msi.<name>
Share data with a community
Informatics applications

Limits
Capacity & bandwidth specific to project
Local Disk

What it is

Non-RAIDed Disk or SSD: POSIX
Visible on host system only
Persistence: duration of PBS job

How you access it:
Shell commands: cp, mv, ...
Applications: all POSIX file IO
Local Disk

Locations & Uses
/scratch.local
[/<user>/<path>]/<file name>
Scales well to many hosts writing to their own files
⇒ Good place for your scratch/work directory

Limits
Scope: local host and life of PBS job
relatively poor bandwidth, except for fragmented IO
Typical capacity: 420 GB
RAM Disk

**What it is**
Local system memory
Visible only on local host
Persistence: duration of PBS job

**How you access it:**
Shell commands: cp, mv, ...
Applications: all POSIX file IO
RAM Disk

Locations & uses
/dev/shm
  [/path]/<file name>
Scalable to many hosts reading their own files
High bandwidth and low latency
Efficient fragmented IO

Limits
About ½ system memory (32 GB on a Mesabi node)
Scope: local to node and only during PBS job.
### Data Hierarchy: Mesabi Compute Node

<table>
<thead>
<tr>
<th></th>
<th>Capacity</th>
<th>Latency</th>
<th>Bandwidth</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>60 MB</td>
<td>~ 10 ns</td>
<td>~ 3 TB/s</td>
<td>In Process</td>
</tr>
<tr>
<td>Memory</td>
<td>64 GB - 1 TB</td>
<td>~ 100 ns</td>
<td>~ 30 GB/s</td>
<td>In Process</td>
</tr>
<tr>
<td>RAM Disk</td>
<td>32 GB - 512 GB</td>
<td>~ 0.1 ms</td>
<td>~ 400 MB/s * N</td>
<td>POSIX IO</td>
</tr>
<tr>
<td>SSD</td>
<td>440 GB</td>
<td>~ 0.26 ms</td>
<td>~ 400 MB/s</td>
<td>POSIX IO</td>
</tr>
<tr>
<td>Local Disk</td>
<td>420 GB</td>
<td>~ 24 ms</td>
<td>~ 100 MB/s</td>
<td>POSIX IO</td>
</tr>
<tr>
<td>PanFS</td>
<td>5.3 PB + ...</td>
<td>~ 2 ms</td>
<td>30 - 200 MB/s</td>
<td>POSIX IO</td>
</tr>
<tr>
<td>CEPH</td>
<td>2.4 PB + ...</td>
<td>~ 1 sec</td>
<td>60 - 1400 MB/s</td>
<td>By File (S3)</td>
</tr>
<tr>
<td>WAN</td>
<td>→ Infinity</td>
<td>~ 1 sec</td>
<td>1 - 60 MB/s</td>
<td>By Web service</td>
</tr>
</tbody>
</table>

- Cache to register bandwidth based on HPL efficiency
- I’ve measured memory BW at 28 GB/s; cache: 267 GB/s
- Latencies and bandwidths are as measured in real apps.
Interfaces
(Getting Started)
Move data to and from MSI

Applications, utilities, & services

- **scp**  can push to msi from external host
- **wget**  Pull from within MSI only
- **Git**   Pull or push from within MSI only
- **s3cmd** Push data to and pull data from CEPH
- **Globus**  Web based control from anywhere

Access to MSI

Must be within UofM domain  (use UofM VPN)
Must go through an MSI front end server

- login.msi.umn.edu  or  NX  or  NICE
Secure Copy (scp)

- Login to MSI host
- Copy files to/from a remote host (r_host)

**Login to MSI**

ssh <msi_user>@login.msi.umn.edu

**Copy to MSI**

scp <r_user>@<r_host>:<path>/<file> <path>
scp -r <r_user>@<r_host>:<path> <path>

**Copy from MSI**

scp <file> <r_user>@<r_host>:<path>
scp -r <path> <r_user>@<r_host>:<path>
Get Files from web (wget)

- Run client (wget) from MSI host
- Get files, source code, data posted on web
  Files must be posted on a server that support wget
  You must have the URL

On an MSI host: get a file from the web:

wget <URL>
Repositories (git)

- Sharing data & source with others: Version control
- Can run git locally or with a github
- UofM github: https://github.umn.edu
- Documentation: https://training.github.com

On MSI host: command prompt

get add
get commit
get merge
CEPH (s3cmd)

What is it good for?

- Move large volumes of data to and from CEPH
- Stage and share data for processing
- High bandwidth: up to 1,400 MB/s

From MSI Linux shell (command prompt)

```
s3cmd mb s3://<bucket>
s3cmd put <file>  s3://<bucket>
s3cmd get s3://<bucket>/<file>  <directory>
s3cmd ls s3://<bucket>
```

Globus

**What is it good for?**

- Move data between sites across WAN & between PanFS and CEPH
- Web GUI driven
- Move LARGE directory trees with a few mouse clicks
- Runs in background

**How to use**

- Login to GLOBUS website w/ your UofM ID
- Register your certificate ID with Globus endpoints
- Use web GUI to drag and drop between endpoints

[www.globus.org](http://www.globus.org)
Globus Home Page: start here

currently

254 PB

... and counting
Select:
Log in

Will use UofM Internet ID
Select University of Minnesota

Log in to use Globus Web App

Use your existing organizational login
e.g. university, national lab, facility, project, Google or Globus ID
(Your Globus username and password used prior to February 13, 2016 is now Globus ID)

University of Minnesota

Continue

Globus uses CILogon to enable you to Log In from this organization. By clicking Continue, you agree to the CILogon privacy policy and you agree to share your username, email address, and affiliation with CILogon and Globus. You also agree for CILogon to issue a certificate that allows Globus to act on your behalf.

Didn't find your organization? Then use Globus ID to sign up.
NOTE:

Use your UofM ID here
Select: 1st endpoint field
Select Globus Endpoint

MSI Home Directories: umnmsi#home

Endpoint
umnmsi#home
owner: umnmsi@globusid.org
no description provided
Authenticate with MSI Account

Please authenticate to access this endpoint

Login Server: wallib40e433700.msl.umn.edu

Username: [blank]
Password: [blank]

Label This Transfer: [blank]

Transfer Settings:
- sync - only transfer new or changed files
- delete files on destination that do not exist on source
- preserve source file modification times
- verify file integrity after transfer
- encrypt transfer
Folders & Files at MSI

Transfer Files

- Endpoint: umnmsi\#home
- Path: ~

Start by selecting an endpoint.

Label This Transfer

Transfer Settings
- sync - only transfer new or changed files
- delete files on destination that do not exist on source
- preserve source file modification times
- verify file integrity after transfer
Etner 2nd Endpoint

Physics Endpoint

umnphys#data

Same UofM authentication
Connected to Physics Server

Endpoint in phys. connected to a 200 TB disk system

This physics endpoint is in the same domain as MSI ⇒ did not need to authenticate again.
Example: pipe directory tree

About 4 levels deep

Irregular

Hundreds of directories

Thousands of files

~0.6 GB

```bash
stratus:data %
stratus:data %
stratus:data %
stratus:data % ls pipe
an l2p1ak0 l2p1ck0 l2p1ek0 lp01dk0 pipe01ck0 pipe01d plots vpko
get l2p1bk0 l2p1dk0 l2p1fk0 pipe01c pipe01ck1 pipe01dk0 run_notes
stratus:data %
stratus:data % ls pipe/l2p1ak0
run set inputs sets system templates
stratus:data %
stratus:data % ls pipe/l2p1ak0/sets
0 1200 1600 200 2300 2700 3000 3400 3800 500 900
100 1300 1700 2000 2400 2800 3100 3500 3900 600
1000 1400 1800 2100 2500 2900 3200 3600 400 700
1100 1500 1900 2200 2600 300 3300 3700 4000 8000
stratus:data %
stratus:data % ls pipe/l2p1ak0/sets/1200
lineX_y00z00_p_k_omega.xy lineZ_xm4y00_p_k_omega.xy lineZ_xp4y00_p_k_omega.xy
lineX_y00z00_U.xy lineZ_xm4y00_U.xy lineZ_xp4y00_U.xy
lineX_y00z99_p_k_omega.xy lineZ_xm8y00_p_k_omega.xy lineZ_xp8y00_p_k_omega.xy
lineX_y00z99_U.xy lineZ_xm8y00_U.xy lineZ_xp8y00_U.xy
lineZ_x00y00_p_k_omega.xy lineZ_xm9y00_p_k_omega.xy lineZ_xp9y00_p_k_omega.xy
lineZ_x00y00_U.xy lineZ_xm9y00_U.xy lineZ_xp9y00_U.xy
stratus:data %
```
Source, Destination, & GO!

Brows to source and destination

**Source**
Folder: pipes
Could be a file or a directory.

**Destination**
path: /data/uchu
File transfer Requested

Temporary notice

Confirms submission of request

Transfer Request Submitted Successfully. Task ID: 7c6c016-84ad-11e5-9941-22000b96db58
The image shows a screenshot of a web interface for monitoring data transfer activities. The interface is part of the Globus platform, which is used for advanced computational research. The page displays a list of activity status, with a focus on small transfer activities that are nearly complete or in progress. The status is indicated with checkmarks for completed transfers and a cancel symbol for a cancelled transfer. The page also indicates that there are loading tasks for further data. The text on the page reads:

- Small transfer
- ~3 min

The interface includes options for managing data, groups, and support, with a tab for viewing activity status.
Click on request to see details.

7788 files
476 folders
598 MB

~3.5 MB/s
Larger & Fewer Files

More efficient

From Physics

32 files
1 folder
200 GB
38 min.
88 MB/s

From NCSA

220 GB
300+ MB/s
SUCCEEDED - 7cbbc016-84ad-11e5-9
94f-22000b96db58

Globus Notification <no-reply@globus.org> 11:44 AM (1 hour ago)

to me

TASK DETAILS
Task ID: 7cbbc016-84ad-11e5-994f-22000b96db58
Task Type: TRANSFER
Status: SUCCEEDED
Is Paused: No
Request Time: 2015-11-06 17:40:36Z
Deadline: 2015-11-07 17:40:36Z
Completion Time: 2015-11-06 17:43:31Z
Total Tasks: 8265
Tasks Successful: 8265
Tasks Expired: 0
Tasks Canceled: 0
Tasks Failed: 0
Tasks Pending: 0
Tasks Retrying: 0
Command: API 0.10 go
Label: n/a
Source Endpoint Name: mesi#panfs
Destination Endpoint Name: umnphys#data
Source Endpoint: d62d1e8d-6d04-11e5-ba46-22000b92c6ec
Destination Endpoint: e4c16ea6-6d04-11e5-ba46-22000b92c6ec
Sync Level: n/a
Data Encryption: No
Use Cases
(HPC Workflows)
Cross OS Workflows

Use case
Complex geometry & physics
Computationally intensive solutions
Use commercial software (example: ANSYS)

The issue
ANSYS Workbench & GUIs run best on MS Windows
ANSYS solvers scale excellently on Mesabi (Linux cluster)

The solution
Setup model & view results w/ GUIs on Citrix VMs
Run solvers on Linux cluster
Use PanFS home directory as the glue
Data Intensive Workflows

**Use case:**
Need to process many large files
Need to access various subsets of data in many ways

**The issues:**
Total volume of data is too large for group quota
Fragmented IO slow on shared file system
MANY users on shared file system → very slow access

**The Solution:**
Stage full data set on CEPH in may files
Stream needed files to RAM disk in PBS jobs
Process on RAM disk and save results to PanFS or CEPH
Storage & Workflows

The point of saving data is to use it.
⇒ Store data with your workflows in mind.
Goals

Give user groups a way to use CEPH, that is

- Easy = easier than what they are doing now
- Reliable = manage & share with confidence
- Fast = faster than PanFS
- Flexible
  - Wide variety of workflows
  - Interactive and automated
  - Other storage & repositories
Approach: Data Hierarchy

- **Project**: One or more *datasets*
- **Dataset**: A sequence \((0,1,2, \ldots, N)\) of *items*
- **Item**: A collection of one or more *names*
- **Name**: A reference to a file, object, or directory

- Datasets and projects also have:
  - **Locations**: directories, buckets, repositories, ...
  - **Small data**: inputs, highly reduced results.
  - **Methods**: scripts or apps that manage or process items.
  - **Workflows**: chains of methods that lead to results.
Example: MHD Model

- 82 time snapshots of 3D state variables
  - 7 real*4 fields (RHO, Vx, Vy, Vz, Bx, By, Bz)
  - Billion cell mesh ($1024^3$)
  - Each snapshot: 28 GiB in 8 files:

<table>
<thead>
<tr>
<th>Size</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3758358528</td>
<td>zme04-0080-000</td>
</tr>
<tr>
<td>3758358528</td>
<td>zme04-0080-001</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>3758358528</td>
<td>zme04-0080-007</td>
</tr>
</tbody>
</table>
Start: Data on PanFS

Command Prompt> ls /home/dhp/public/imhd/zme04/dumps

restart_set1-000  zme04-0016-002  zme04-0032-006  zme04-0049-002  zme04-0065-006
restart_set2-000  zme04-0016-003  zme04-0032-007  zme04-0049-003  zme04-0065-007
zme04-0000-000  zme04-0016-004  zme04-0033-000  zme04-0049-004  zme04-0066-000
zme04-0000-001  zme04-0016-005  zme04-0033-001  zme04-0049-005  zme04-0066-001
zme04-0000-002  zme04-0016-006  zme04-0033-002  zme04-0049-006  zme04-0066-002
zme04-0000-003  zme04-0016-007  zme04-0033-003  zme04-0049-007  zme04-0066-003
zme04-0000-004  zme04-0017-000  zme04-0033-004  zme04-0050-000  zme04-0066-004
zme04-0000-005  zme04-0017-001  zme04-0033-005  zme04-0050-001  zme04-0066-005
zme04-0000-006  zme04-0017-002  zme04-0033-006  zme04-0050-002  zme04-0066-006
zme04-0000-007  zme04-0017-003  zme04-0033-007  zme04-0050-003  zme04-0066-007

...

• 658 files
• 2.6 TiB
Register A New Dataset

Command Prompt> cd /home/dhp/public/imhd/zme04/dumps
Command Prompt> available new zme04 ['Closed loop ...']

description="Closed loop B-field, k=[32,62], ampb=0.01, powb=0, mesh=1024"
...
export   datasetdir="/home/dhp/public/imhd/zme04/dumps"
export   dataset_s3="s3://dhp-imhd-zme04-dumps"
export   input_dir="/home/dhp/dhp/.available/imhd_minimal"
export   results_dir=/home/dhp/dhp/data/post/zme04
...
export   one_seq_item="adump_names.sh zme04 8"
List Datasets & Select One

Command Prompt> available

Label    Description
bl102    IMHD, single loop init B-field on a 256x256x256 mesh
zmc03    Closed loop B-field, k=[8,16], ampb=0.01, powb=0, mesh=512
...
zme04    Closed loop B-field, k=[32,62], ampb=0.01, powb=0, mesh=1024

Command Prompt> cd any_directory

Command Prompt> available zme04

Data in:  /home/dhp/public/imhd/zme04/dumps
Closed loop B-field, k=[32,62], ampb=0.01, powb=0, mesh=1024
Copy Data to CEPH

Command Prompt> qsub sync.pbs

#!/bin/bash -l
#PBS -l nodes=1:ppn=1,walltime=40:00:00
#PBS -j oe
cd $PBS_O_WORKDIR
available s3sync

PBS job took 23.5 hr. To copy 2.6 TB to CEPH
List Available Data

Command Prompt> summarize_data.sh

/scratch.global/dhp/zme04
  Complete     [0-66], [70-80]
  Incomplete   67, 69
  Missing      68

/home/dhp/public/imhd/zme04/dumps
  Complete     [0-81]

s3://dhp-imhd-zme04-dumps
  Complete     [0-81]
Get One Time Snapshot

Command Prompt> cd /dev/shm/dhp
Command Prompt> available zme04
Command Prompt> time get_one_from_s3.sh 80
real 1m0.333s
user 1m53.599s
sys 1m9.520s

-rw------- 1 dhp dhp 3758358528 Nov 27 20:43 zme04-0080-000
-rw------- 1 dhp dhp 3758358528 Nov 27 20:45 zme04-0080-001
...
-rw------- 1 dhp dhp 3758358528 Nov 27 20:57 zme04-0080-007

• Pulled 28 GiB from CEPH to RAM in ~60 sec
• Run on a Mesabi compute node, in /dev/shm/dhp
View Total Energy

Command Prompt> ee 80 TE view
#Wall Clock: read,work,out,full:  14.323  13.706  5.348  33.381

- Processed 28 GiB in ~33 sec
- Used file:  formulas.e3d
  Automatically copied from  $input_dir
  
  \[ V = Vx \ V_y \ V_z \]
  \[ V_2 = \text{dot}(V,V) \]
  \[ KE = 0.5 \ \text{RHO} \ \text{V}_2 \]
  \[ B = Bx \ By \ Bz \]
  \[ BE = 0.5 \ \text{dot}(B,B) \]
  \[ TE = BE + KE \]

- Uses all 7 fields
## Read & Process Times

<table>
<thead>
<tr>
<th></th>
<th>/dev/shm</th>
<th>/home</th>
<th>/scratch.global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy From S3</td>
<td>60 sec</td>
<td></td>
<td>~91 s</td>
</tr>
<tr>
<td>Full time for App</td>
<td>33 sec</td>
<td>~605 s</td>
<td>620 - 1912 s</td>
</tr>
<tr>
<td>App Read Time</td>
<td>14 sec</td>
<td>~585 s</td>
<td>600 - 1894 s</td>
</tr>
<tr>
<td>dd -bs 8MiB</td>
<td>6 sec</td>
<td>104-297 s</td>
<td>~199 s</td>
</tr>
<tr>
<td>md5sum</td>
<td>62 sec</td>
<td>~115 s</td>
<td>~120 s</td>
</tr>
</tbody>
</table>

- 28 GiB in 8 files
- Data pre-staged weeks before.
- First time reads (data NOT cached)
- App reads 2 MiB chucks with seeks

- **Nothing beats RAM disk.**
- **IO to & from RAM can scale to multiple nodes.**
BE, KE, & Vorticity

Command Prompt> eem 80 "BE KE vort" view 3x1 z=0
Wallclock: 3.6 sec
Sweep Through Data

#PBS -l nodes=4:ppn=24,walltime=01:00:00  
#PBS -j oe

module load parallel

cd $PBS_O_WORKDIR
uniq $PBS_NODEFILE > nodes
available zme04

seq 0 81 | parallel --jobs 1 --sshloginfile nodes --workdir $PWD ./PROC {} {#}

- Use 4 Mesabi nodes
- "PROC" : A script to process one snapshot.
- One instance of "PROC" script on each node at a time
- GNU parallel sequences over all 82 snapshots
#!/bin/bash -l
source dataset.info
item=$1
task=$2

proc_dir=/dev/shm/dhp.$task          # Working directory
# Get inputs
cp -r $input_dir $proc_dir
# Get data
cd $proc_dir
get_one_from_s3.sh $item

# Do not display
touch do_not_display

# Process data
ee $item KE zprof
ee $item BE zprof
ee $item vort zprof

# Save results
cp *.zprof $results_dir/sweep

# Cleanup
rm -rf $proc_dir
Parallel Processing of Data on CEPH

RED: CEPH ⇒ /dev/shm
BLUE: Calculate 3 profiles
4 Mesabi Nodes
2.46 TB in 37 min.
⇒ 1.1 GB/s sustained
Thank You
Hands-On
Project lifecycle

• Get & build an application

• Run application, generate data, examine results

• Organize and save data

• Share data

• Clean up
Get Application

Get example from web & unpack

  firefox http://tinyurl.com/z8n4d36
  ⇒ Download cycles.tarz

  mv ~/Downloads/cycles.tarz .

  OR:

  ⇒ cp /home/dhp/public/cycles.tarz .

  tar xvfz cycles.tarz

Go into directory and build example application

  cd cycles

  make
Test application

Run application to get synopsis

./cycles

Should get synopsis: usage: cycles <fx> <fy>
App. takes two command line arguments.
These can be integers or floats.

Try an example

./cycles 1 2

You should get 1001 lines: 2 columns of numbers
Run a test case & plot results

Script test1:

./cycles 3 5 > cyc_3_5.dat
gnuplot -persist cyc_3_5.plt

Run it:

./test1
Try your own Parameters

Script test2
./cycles $1 $2 > cycles.dat
gnuplot -persist cycles.plt

Try several examples
./test2 2 3
./test2 13 25
./test2 2 3.02
Parameter space study

**Script test3**

```bash
#!/bin/bash
for j in $(seq 1 2 7)
do
  for i in $(seq 2 2 8)
do
    ./cycles $i $j > cyc_${i}_${j}.dat
  done
done
done
ls -l cyc*.dat
```

*Run it and generate output files (cyc*.dat)*

`./test3`
Organize & your data

**Make an output directory**

mkdir output
mv *.dat output

**Make a zipped tar file**

tar cvfz output.tarz output

**Share with other members of your group**

cp -r output ~/../shared
chmod -R g=u-w ~/../shared/output
Save data to CEPH

**Make a bucket and save a file**

s3cmd mb s3://${USER}_mytest
s3cmd put output/cyc_2_1.dat s3://${USER}_mytest

**Save all data files to bucket**

for i in output/*
do
    s3cmd put $i s3://${USER}_mytest
done

Which is faster?

**or save tar archive**

s3cmd put output.tarz s3://${USER}_mytest
Use data on CEPH

Get a data file from bucket

s3cmd get s3://${USER}_mytest/cyc_2_3.dat

Desktop & Web access to CEPH

https://www.msi.umn.edu/support/faq/what-are-some-user-friendly-ways-use-second-tier-storage-s3
Clean up

The situation
Immediate analysis is done.
Data is organized, shared, and saved (on CEPH)
Assume the data is a large fraction of your group quota

Time to clean up
Fine to save source, scripts, and inputs in your home directory
Better to have them organized where you and your group can find it
⇒ Remove the large data files
NO LONGER NEEDED

Set Keys For s3cmd

Run A Setup Shell Script  (only do this once)

On: login.msi.umn.edu
/home/tech/public/porter/ceph/scripts/setup_s3cfg

What it does

Creates a small file in:  ~/.s3cfg
Which contains your personal access keys for CEPH

You can now:

Use s3cmd command on all MSI Linux systems
Can use s3cmd in batch jobs
Post processing example

Have: raw data from an MHD turbulence model.
Mesh res: 256x256x256
Full state info: (density, velocity, B-field)
Individual snapshot size: 470 MB
300+ snapshots in time

Want: Power spectra of velocity field
Post-process each time snapshot
Can be done independently
Calculation (including IO) takes ~16 s
Serial workflow

Command Status
----------------- --------
./do1spc 0000 FINISHED
./do1spc 0001 FINISHED
./do1spc 0002 INPROGRESS
./do1spc 0003 NEW
./do1spc 0004 NEW
...

Run app. on state 0002
Raw data on PanFS
Generate V-spectra
copy to output directory

e6a02-0000-000
e6a02-0001-000
e6a02-0002-000
e6a02-0003-000
e6a02-0004-000
...
e6a02-0000-V3.spc3v
e6a02-0001-V3.spc3v
e6a02-0002-V3.spc3v

© 2009 Regents of the University of Minnesota. All rights reserved.
Serial Throughput (0-9)

Lines show span of time each work item took

1 work item = process one time snapshot
Parallel workflow

... ./d01spc 0007 FINISHED
./d01spc 0008 FINISHED
./d01spc 0009 INPROGRESS
./d01spc 0010 INPROGRESS
./d01spc 0011 INPROGRESS
./d01spc 0012 NEW
./d01spc 0013 NEW
./d01spc 0014 NEW
...

Run app. on state 0009
Run app. on state 0010
Run app. on state 0011

... e6a02-0007-V3.spc3v
e6a02-0008-V3.spc3v
e6a02-0009-V3.spc3v
e6a02-0010-V3.spc3v
e6a02-0011-V3.spc3v
...

© 2009 Regents of the University of Minnesota. All rights reserved.
Parallel throughput (0-40)

1 Mesabi node
20 Workers

Each worker grabs next work item as soon as it finishes

Variable times:
  - Shared PanFS
  - Variable loads
Parallel Throughput (0-299)

1 Mesabi node
20 Workers
Processed:
  300 files
  330 sec.

1 worker:
  300 files
  ~4800 sec
Process data from CEPH

**Workflow with raw data on CEPH**

- Use `s3cmd` to pull raw data files
  
  CEPH $\Rightarrow$ RAM disk
  
  Process on RAM disk then copy results to PanFS

**Issue**

- If not staged on CEPH SSDs, getting 440MB can take ~17s

**Overlap copy from CEPH with calculation**

- 1 work item = process 5 consecutive states
  
  work on state $i$ while pulling state $i+1$
Parallel throughput from CEPH

1 Mesabi node
20 Workers
Processed:
  300 files
  390 sec.

Compare to same data off of PanFS:
  330 sec