How do I manage my data on the file system?

Amit Majumdar
Division Director, Data Enabled Scientific Computing
Rick Wagner
HPC Systems Manager

Outline

1. **General discussion**

2. **SDSC File Systems on HPC Machines**

3. **Use case 1: Janssen Pharmaceutical genomics analysis** (by Glenn Lockwood, ex-SDSC staff)

4. **Use case 2: in-situ visualization** (joint work w/Homa Karimabadi, UCSD)

5. **Summary**

1. General discussion

Dealing with Data: Choosing a Good Storage Technology for Your Application

Application Focus

Storage choices should be driven by application need, not just what's available.

But, applications need to adapt as they scale.

Writing a few small files to an NFS server is fine… writing 1000's simultaneously will wipe out the server.

If you use binary files, don't invent your own format. Consider HDF5.

Storage Technologies

File Systems

- ext4
- NFS
- Lustre
- PVFS
- FUSE

Devices

- memory
- block

Services

- Cloud
- MySQL
- CouchDB

Storage Technologies

File Systems

- ext4
- NFS
- Lustre
- PVFS
- FUSE

Devices

- memory
- block

Services

- Cloud
- MySQL
- CouchDB

Each has its own performance characteristics

Not all are available everywhere

File Systems

Classic access, POSIX, Windows

Most relevant:

- Local
- Remote
  - NFS, CIFS
  - Parallel (Lustre, GPFS)

Local file systems are good for small and temporary files

Network file systems very convenient for sharing data between systems

A Cautionary Tale

http://www.youtube.com/watch?v=gDfLXAtRJfY&feature=youtu.be

Devices

Raw block device (/dev/sdb) or RAM FS (/dev/shm)

Useful in specific cases, like fast scratch

Can be very good for small I/O

Services

Things accessed programmatically

Frequently the last thought for HPC applications: A MISTAKE

Databases
Cloud storage (Amazon S3)
Document storage (MongoDB, CouchDB)

Know What You Need

http://www.youtube.com/watch?v=F4OIDszDA9E

Order of Magnitude Guide

Storage	files/directory	file sizes	BW	IOPs
Local HDD	1000s	GB	100 MB/s	100
Local SSD	1000s	GB	1 GB/s	10000+
RAM FS	10000s	GB	10 GB/s	10000
NFS	100s	GB	100 MB/s	100
Lustre/GPFS	100s	TB	100 GB/s	1000
Cloud	Infinite	TB	10 GB/s	0
DB	N/A	N/A	N/A	10000

Choosing

My application needs to:

Write a checkpoint dump from memory from a large parallel simulation.

I should consider:

A parallel file system and a binary file format like HDF5.

Choosing

My application needs to:

Run analysis on remote systems and return the results to a web portal for users.

I should consider:

Cloud storage for results and input, and local scratch space for the job.

Choosing

My application needs to:

Randomly access many small files, or read and write small blocks from large files.

I should consider:

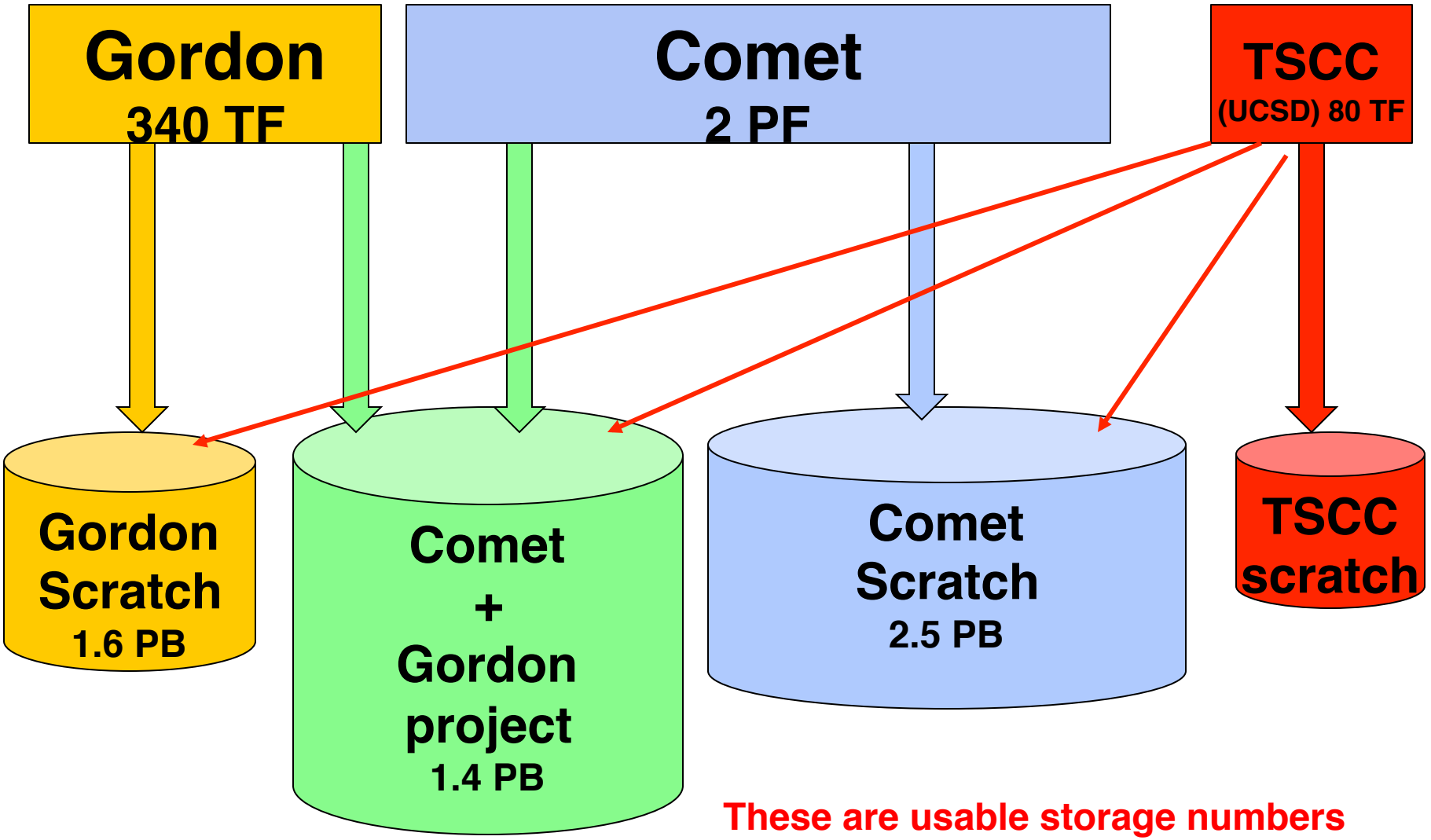
A database, RAM FS, or local scratch space.

2. SDSC File Systems on HPC Machines

SDSC HPC Machine File Systems: Comet and Gordon

- **nfs file system - /home when you login**
- **SSDs : local scratch and file systems**
- **Parallel file system Data Oasis (Lustre) – accessed from each machine (Gordon, Comet)**
  - Lustre scratch (can/will be purged)
  - Lustre projects (not backed up; not purged)
    - Some default (500 GB) allocation; additional needs proposal
  - Gordon Lustre aggregate performance: 100 GB/sec
  - Comet Lustre aggregate performance: 200 GB/sec

SDSC High Level File System Architecture



These are usable storage numbers

File system path

- **Lustre project location on Gordon and Comet:**

```
/oasis/projects/nsf/<allocation>/<user>
(show_accounts will show "allocation")
(mine (my userID majumdar): /oasis/
projects/nsf/sds128/majumdar)
```

- **Lustre scratch**
  - **Comet**

```
/oasis/scratch/comet/<user>
(mine: /oasis/scratch/comet/majumdar)
```

  - **Gordon**

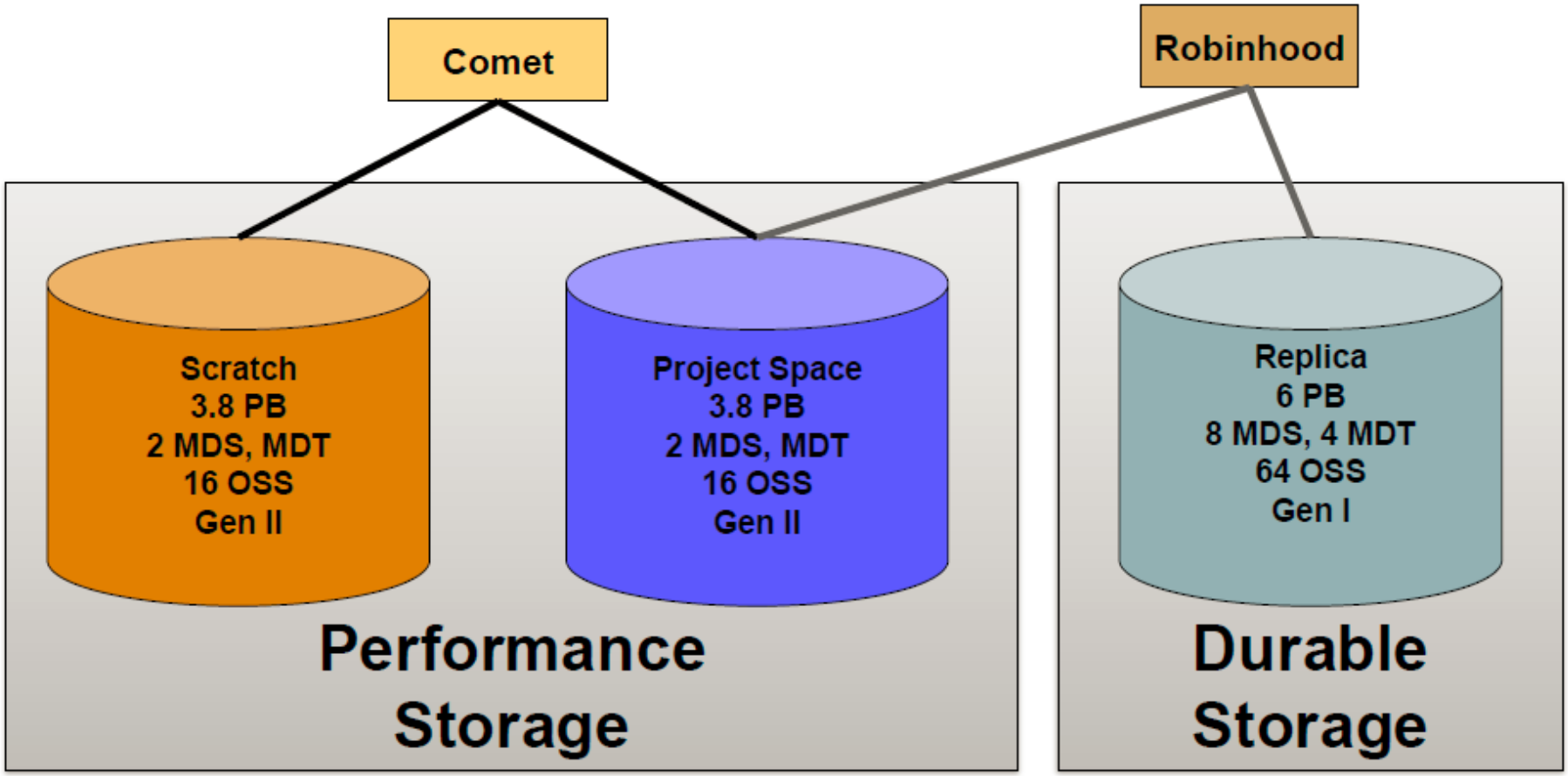
```
/oasis/scratch/<user>
(mine: /oasis/scratch/majumdar)
```

SSDs on Comet and Gordon

- **Comet node: 320 GB SSDs; ~230 GB usable (local disk)**
  - Access only during job execution
    `/scratch/$USER/$SLURM_JOBID`

- **Gordon – access only during job execution**
  - Each compute node has 300GB SSDs; ~280GB usable
    `/scratch/$USER/$PBS_JOBID`
  - Bigflash nodes have access to 4.4TB file system
    `/scratch/$USER/$PBS_JOBID` (and add to PBS script:
    `#PBS -l nodes=1:ppn=16:native:bigflash`)
  - vSMP nodes have access to 4.4 TB SSD file system
    `/scratch1/$USER/$PBS_JOBID (16-way and larger)`

Looking into the future

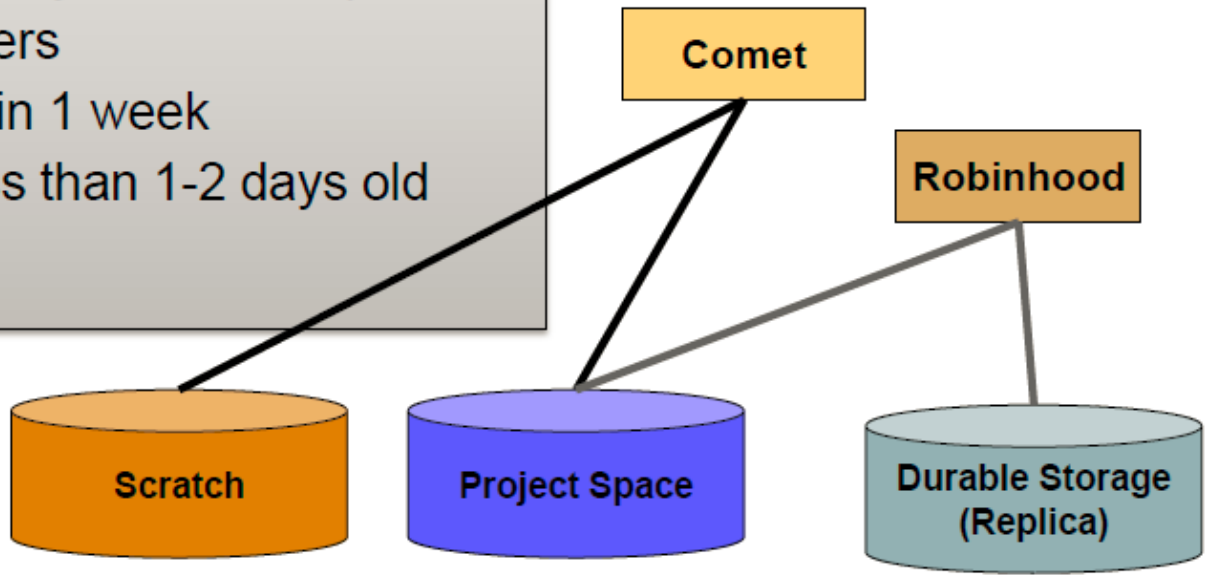


These are raw storage numbers

Replication and Migration

Durable Storage

- Reuse current Data Oasis servers as slightly stale replica
- Replicates allocate project space
- Think “disaster recovery” not “backup”
- Not accessible to users
- Goal is full sync within 1 week
- Exclude changes less than 1-2 days old
- Using Robinhood



Tracking data

http://www.youtube.com/watch?v=N2zK3sAtr-4

Database logos courtesy of RRZEicons
http://commons.wikimedia.org/

3. Use case 1: Janssen Pharmaceutical genomics analysis (by Glenn Lockwood)

Another Topic: Moving Data

Mahidhar talked about Globus and other data transfer tools earlier

Example from an industrial project with Janssen Pharmaceuticals on large-scale genomic analysis. See Glenn Lockwood's webinar:

http://www.sdsc.edu/Events/ipp_webinars/index.html

http://www.sdsc.edu/Events/ipp_webinars/large_scale_genomics.pdf

Step #1: Data Requirements

- **Input Data**
  - Raw reads from 438 full human genomes (fastq.gz)
  - 50 TB compressed data from Janssen R&D
- **Output Data**
  - + 50 TB of high quality mapped reads
  - + small amount (< 1TB) of called variants
- **Intermediate (scratch) Data**
  - + 250 TB
- **Performance**
  - Data must be stored **online**
  - High bandwidth to storage (> 10 GB/s)

Step #3 (from the Janssen use case): Loading Data

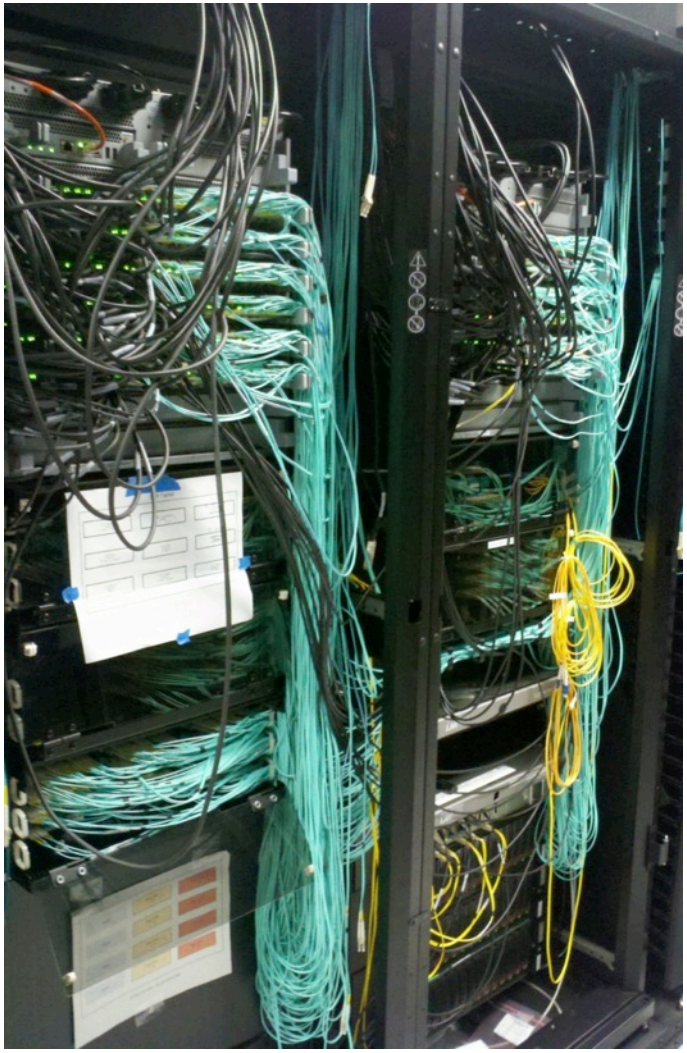
- **Input: raw reads from 438 full human genomes**
  - 50 TB of compressed, encrypted data from Janssen
  - 4,230 files (paired-end .fastq.gz)

How do you get this data into a supercomputer?

They don't exactly have USB ports

Pictured: 2 of 12 Data Oasis racks; each blue light is 2x4 TB disks

Loading Data via Network



External

- 100 Gbit/s connectivity to outside world

Internal

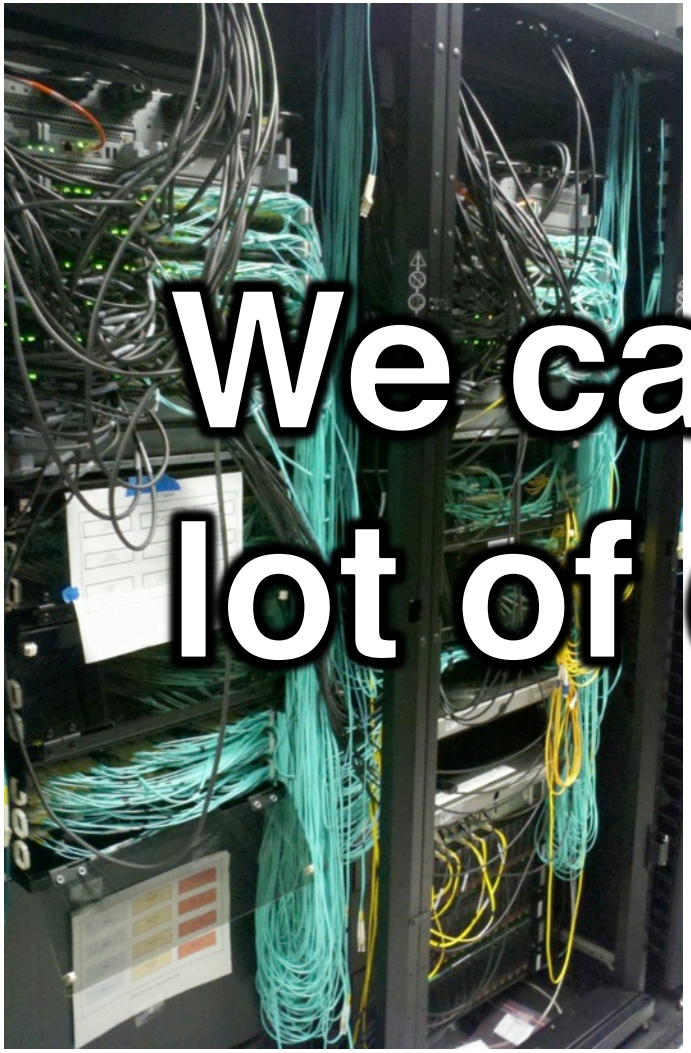
- 60 Tbit/s switching capacity
- 100 Gbit/s from Data Oasis to edge

Gordon

- 20 Gbit/s from IO nodes to Data Oasis
- 40 Gbit/s dedicated storage fabric (IB)

Pictured: 2x Arista 7508 switches, core of the HPC network

Loading Data via Network



We can move a lot of data

External

- 60 Gbit/s connectivity to outside world
- 100 Gbit/s connection installed and being tested
- [illegible]
- 60 Tbit/s switching capacity
- [illegible] from Data Oasis to edge [illegible] summer

Gordon

- 20 Gbit/s from IO nodes to Data Oasis
- 40 Gbit/s dedicated storage fabric (IB)

Pictured: 2x Arista 7508 switches, core of the HPC network

Loading Data via "Other Means"

- **Most labs and offices are not wired like SDSC**
  - Janssen's 50 TB of data behind a DS3 link (45 Mbit/sec)
  - Highest bandwidth approach...

Left: 18 x 4 TB HDDs from Janssen
Right: BMW 328i, capable of > 1 Tbit/sec

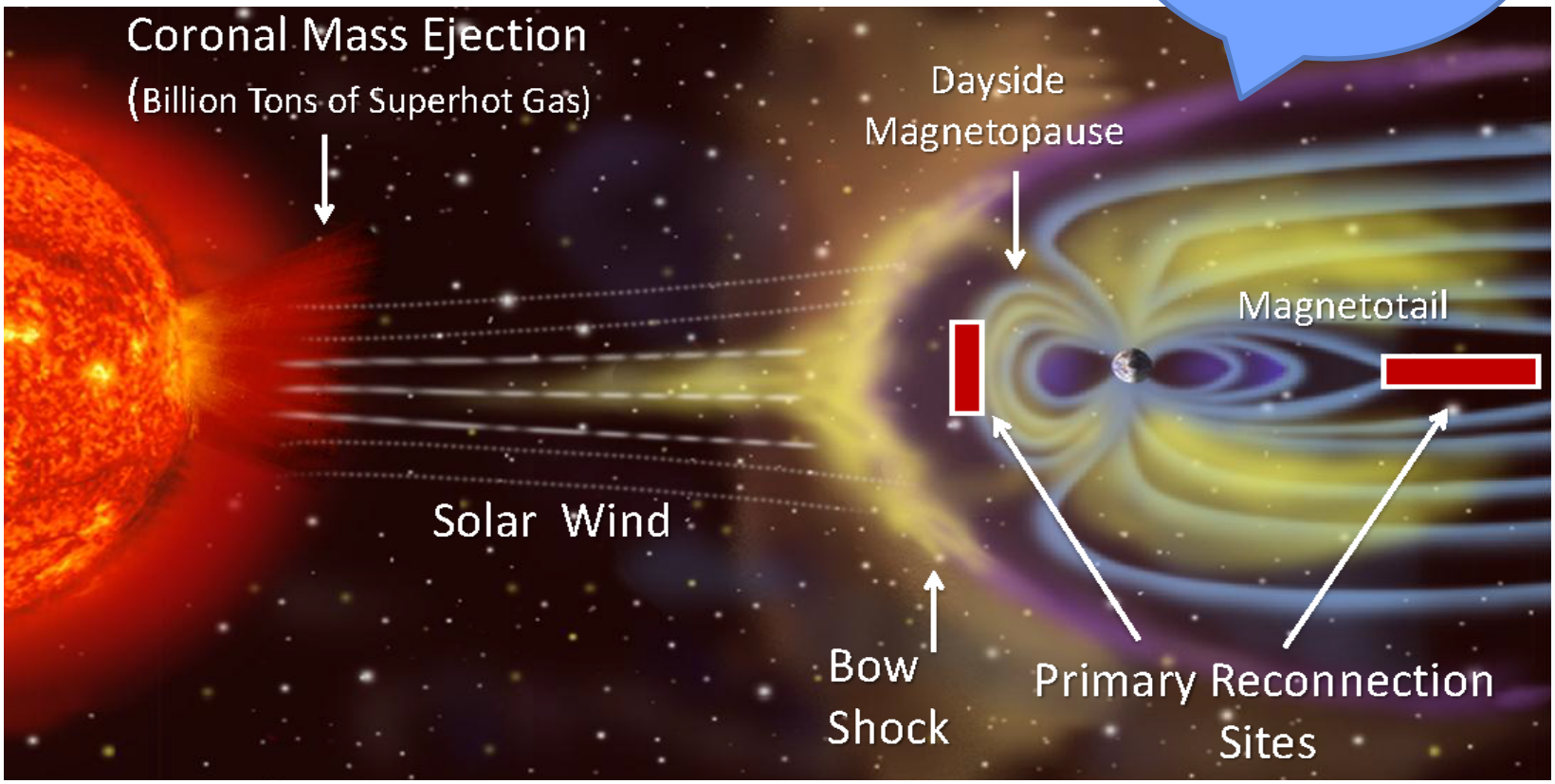
Step #3: Loading Data

- **Use the network if possible:**
  - .edu, federal labs (Internet2, ESnet, etc)
  - Cloud (e.g., Amazon S3)
    - 100 MB/sec to **`us-west-2`** disk-to-disk
    - 50 MB/sec to **`us-east-1`** disk-to-disk
  - SDSC Cloud Services (> 1.6 GB/sec)
- **Sneakernet is high bandwidth, high latency**
  - Sneakernet is error-prone and tedious
  - Neither scalable nor future-proof
  - For Janssen, it was more time-effective to transfer results at 50 MB/s than via USB drives and a car trunk

4. Use case 2: in-situ visualization

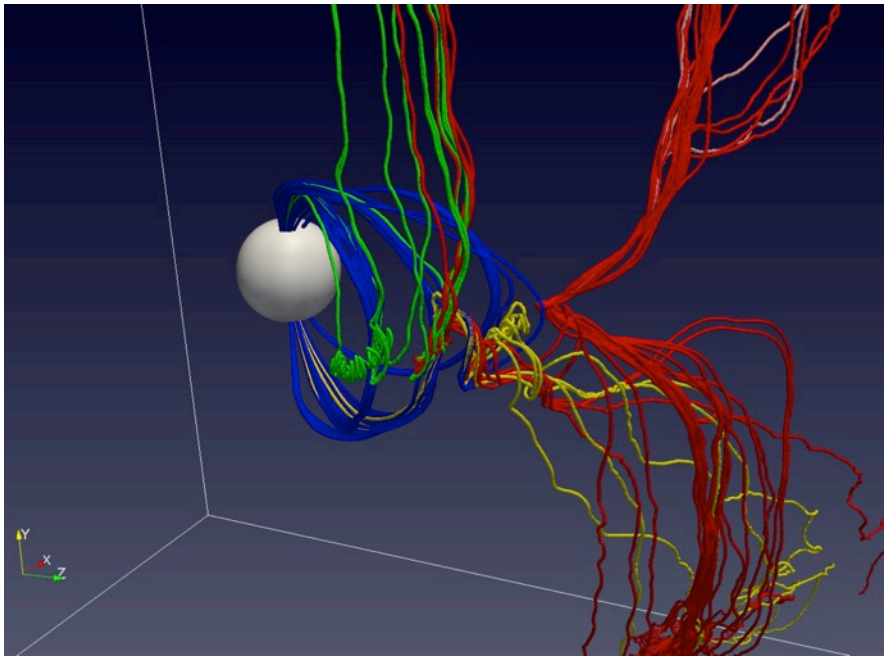
Space Weather

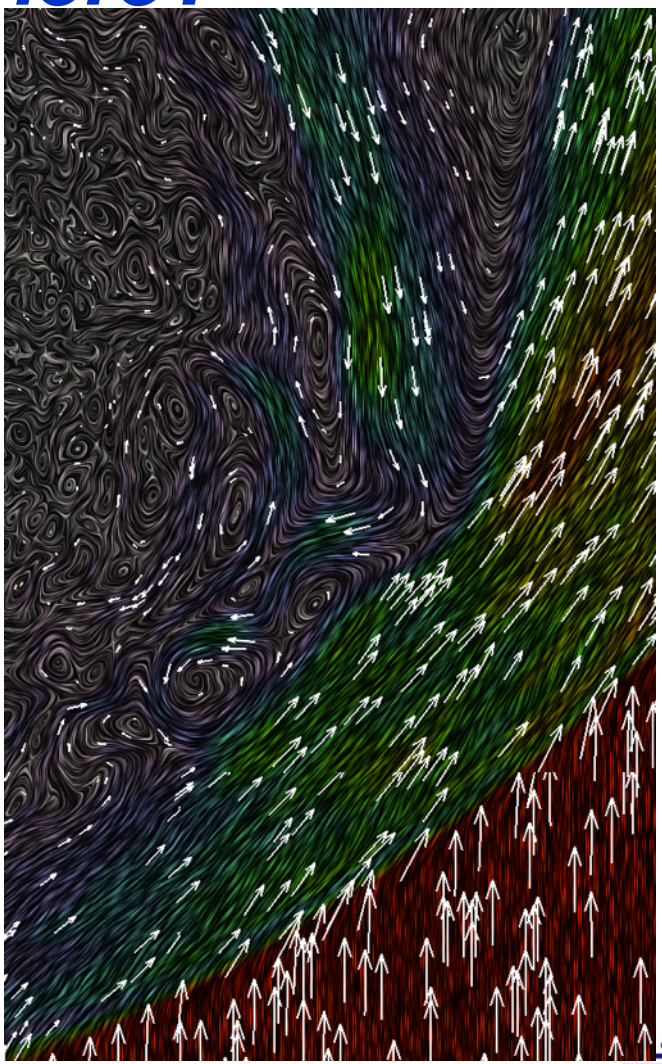
Earth's magnetic field provides a protective cocoon but it breaks during strong solar storms



90 million miles or ~ 100 Suns

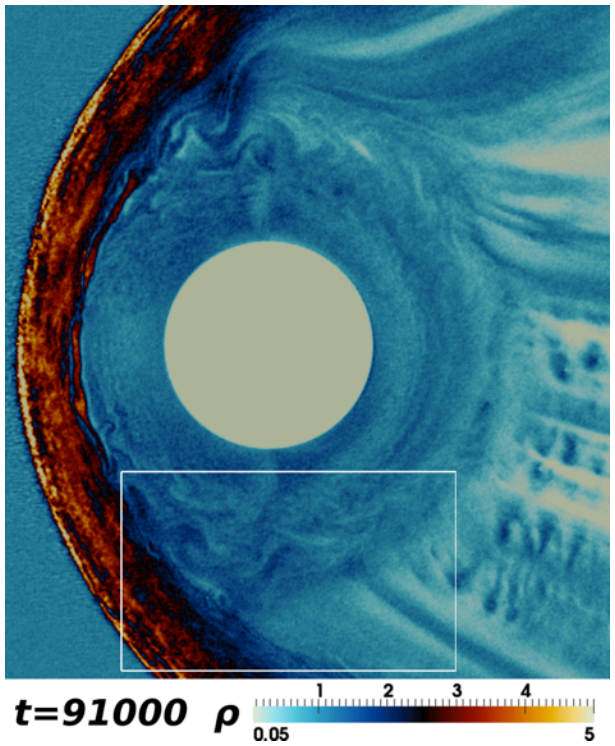
What are the source of vortices inside the Magnetosphere?

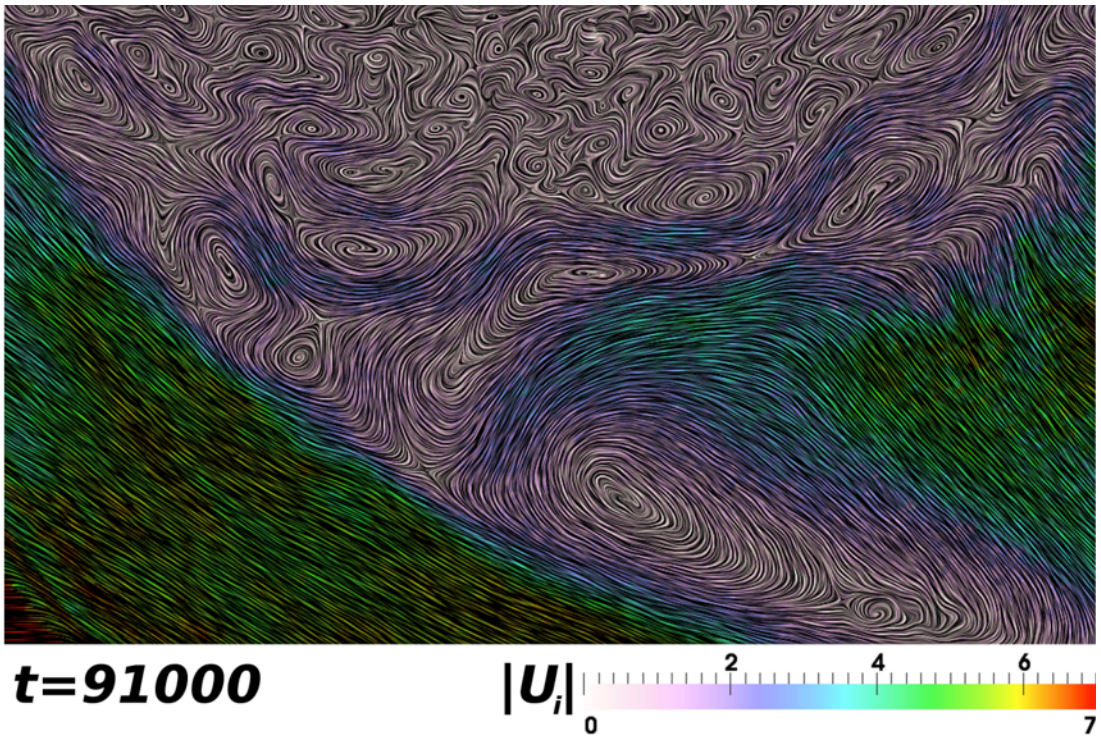




- Confirmed by spacecraft observations
- Vortices at the MP boundary often have FTE's near the vortex core. Are the FTE's causing the vortex? Or…
- Are these driven by Kelvin-Helmholtz instability?

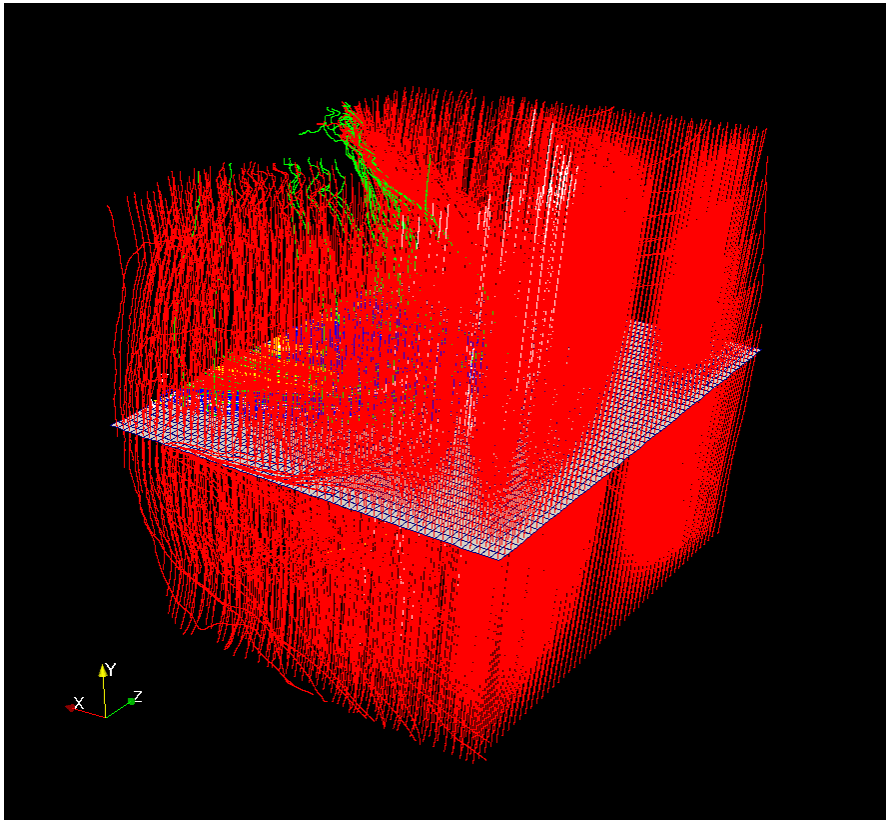
Why In-situ for this study?

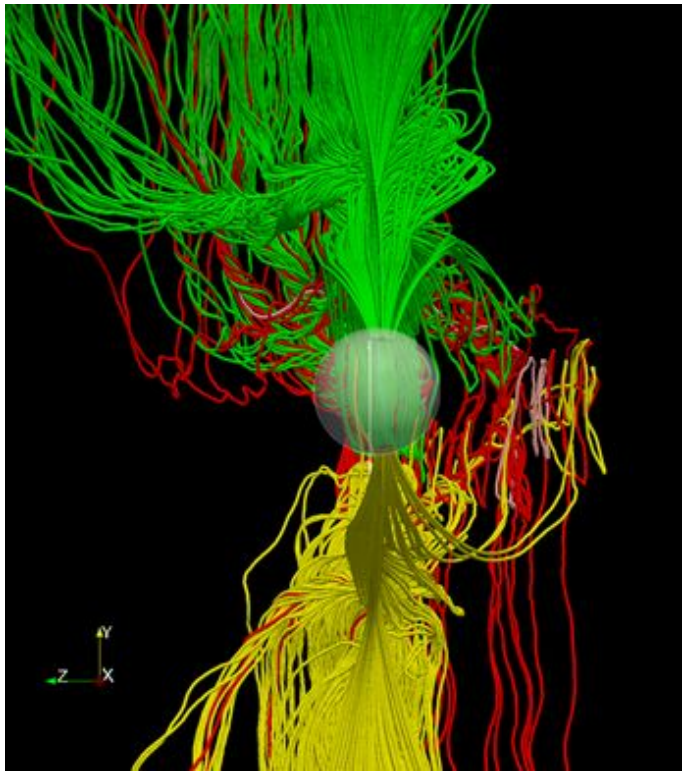




- Simulation is I/O bound and runs are limited by disk quotas
- With in-situ we reduce I/O costs by using “extracts”
- And increase temporal resolution
- Removes need for post processing – saves disk and computational time.

Information Overload : Looking for Flux Ropes

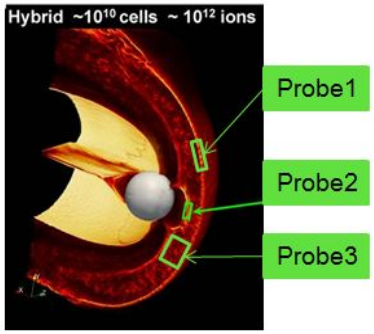




In Situ Visualization

- For many of the analysis, need high time resolution data dumps. However, this is not practical since each data dump can take over 10 TB.
- In situ visualization provides a possible solution.

In Situ Visualization Using Intelligent Probes



Features of Intelligent Probes

Targeted data collection
Ability to mimic spacecraft-like trajectories
Creation of instrument-like products
Generation of high resolution data products

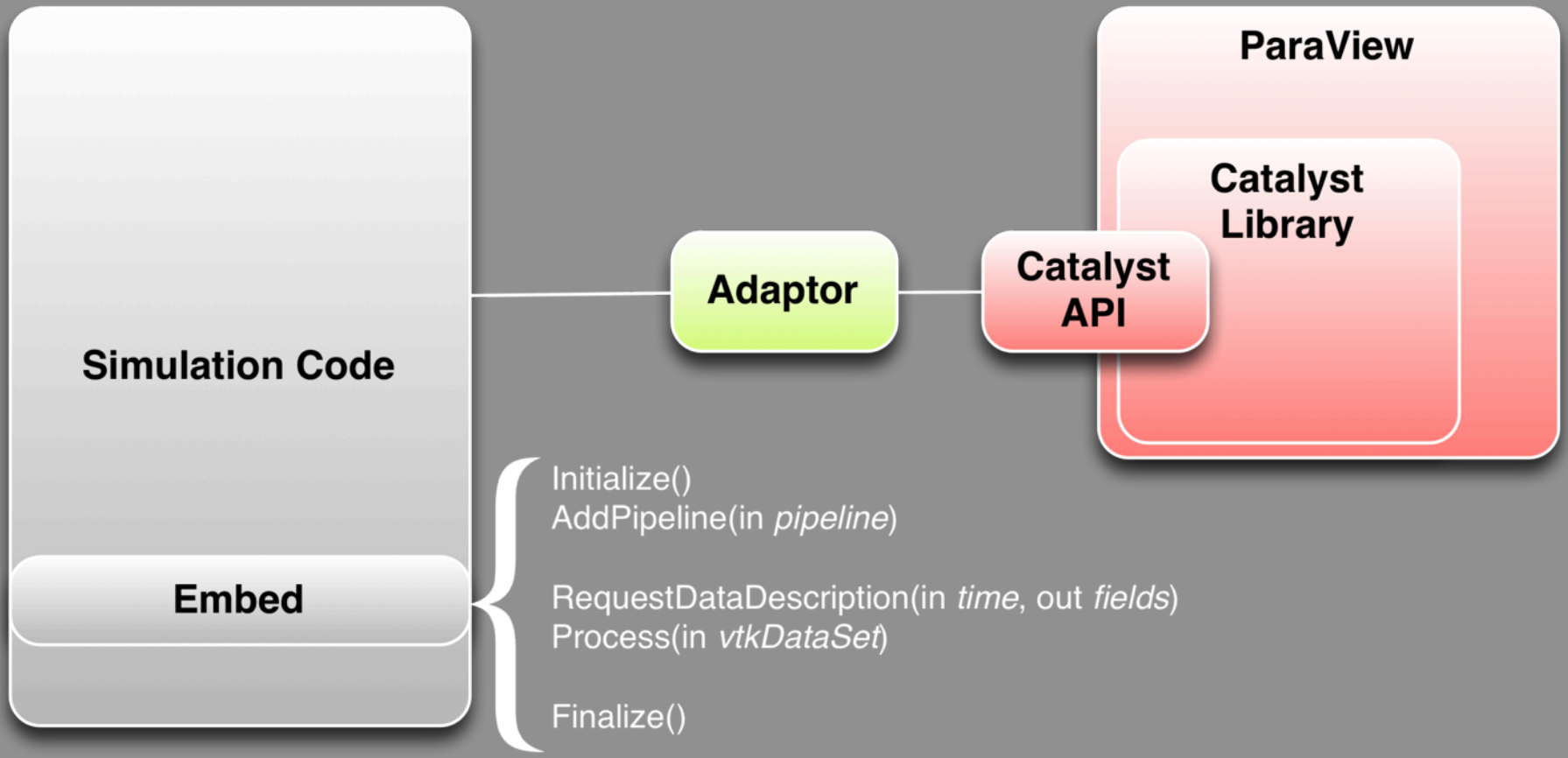
Selective Outputs

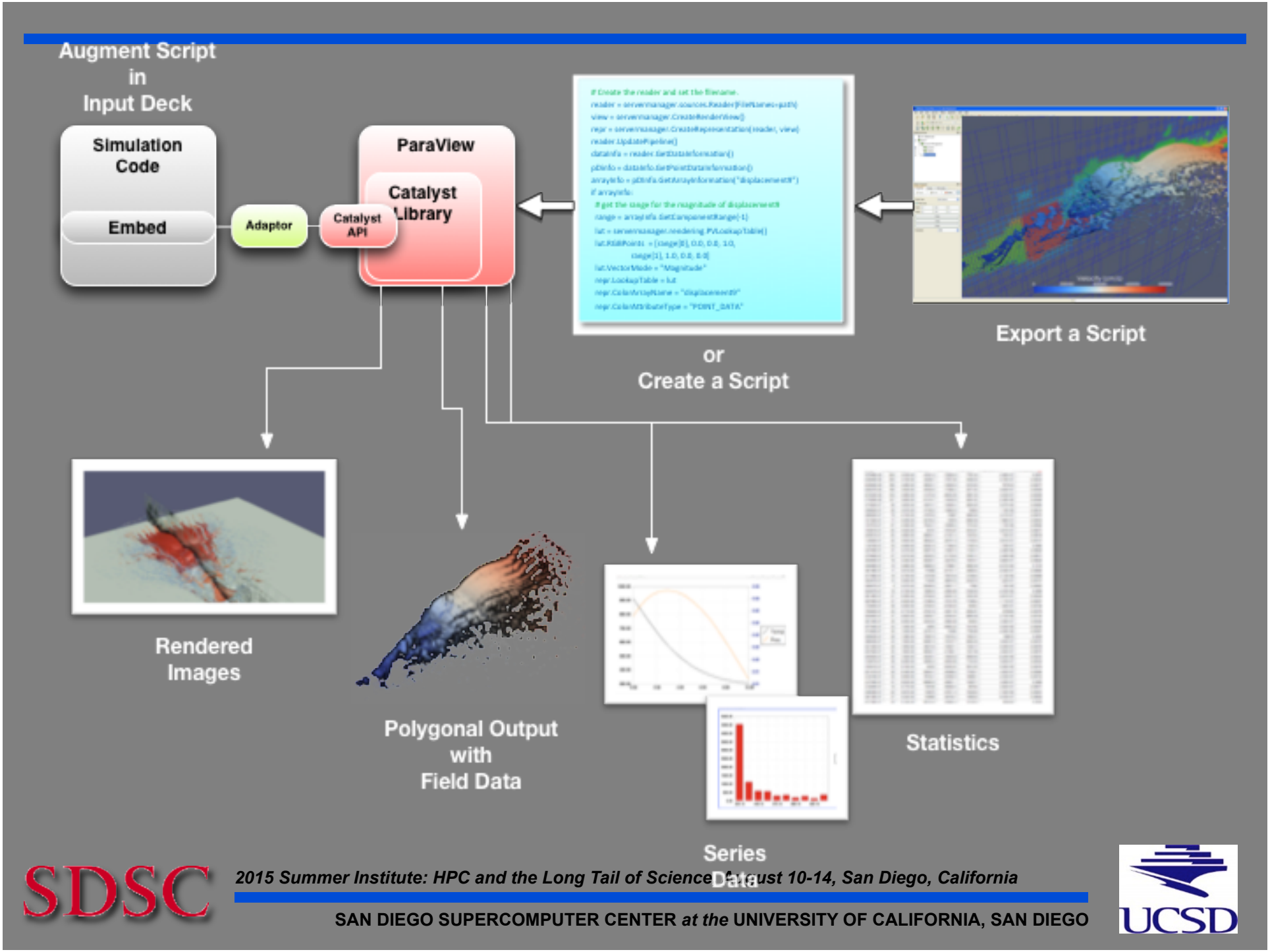
Images and visualizations
Data from probes (less raw data)
Statistics

Enables

Direct comparison with spacecraft data
Orbit optimization
Instrument design and testing

ParaView Catalyst Architecture




Augment Script
in
Input Deck
Simulation
Code
Embed
Adaptor
Catalyst
API
ParaView
Catalyst
Library
or
Create a Script
Export a Script
Rendered
Images
Polygonal Output
with
Field Data
Series
Data
Statistics

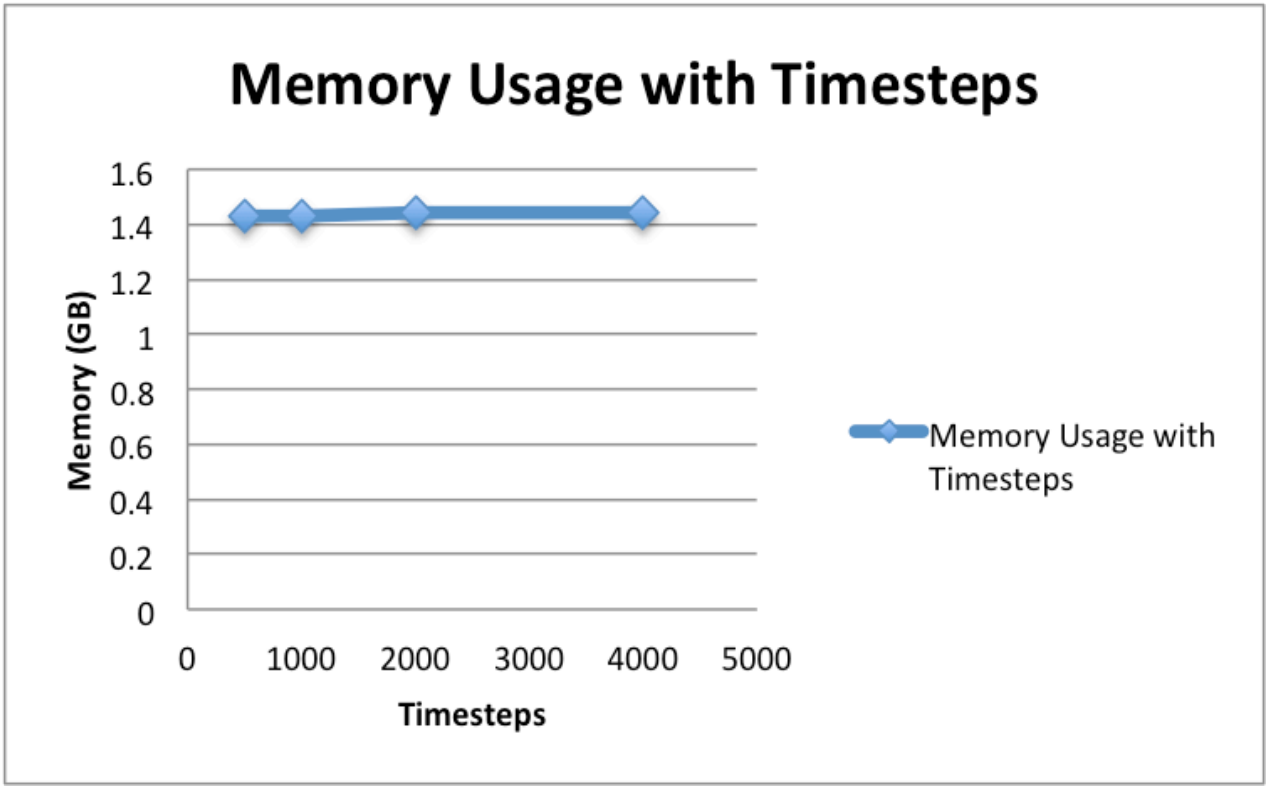
Results – Memory Overhead of In-Situ Viz

- The first aspect studied was the memory overhead of the in-situ visualization on the simulations. Table shows results for varying grid and particle counts.
- The results show that the memory overhead is fairly constant at around 360MB/core.

Memory overhead comparison with varying grid and varying particle counts.

Cores	Grid Size	Millions of Particles	GB RAM / Core	
			non-in-situ	in-situ
32	$64^2 \times 256$	33.5	0.97	1.34
64	64×256^2	67	0.98	1.34
128	128^3	134	0.98	1.34

Results – Memory Overhead of In-Situ Viz



Variation of memory usage with timesteps for the in-situ visualization test case.

Performance Impact of In-Situ Viz

- Test runs conducted with In-Situ Viz output *every timestep* (extreme case). Run times compared with normal case where output is every 500 steps.
- Second set is weak scaling with varying particle counts and grid sizes. 20-35% impact on run time with 8 cores/node. Performance impact consistent up to 1k cores.

Wall clock time (for 500 timestep run) variation with different core counts (fixed 8 cores/node), with a varying grid size, and varying particle counts.

Cores	Grid Size	Millions of Particles	Wall-clock time (sec)	
			non-in-situ	in-situ
32	64^2 x 128	33.5	280.35	340.64
64	64 x 128^2	67	289.43	380.45
128	128^3	134	311.91	401.88
256	1 2 8 ^{2} x 256	268.4	318.64	430.58
512	1 2 8 x 256^2	536.9	354.36	453.75
1024 ‡	256^3	1073.74	441.03	546.77
‡ — 16 cores per node				

Performance and Memory Impact of In-Situ Viz

- Test runs conducted with In-Situ Viz output *every timestep* (extreme case). Run times compared with normal case where output is every 500 steps.
- Strong scaling with problem size fixed at 64x128x128, 67M particles.
- Memory overhead is almost constant and performance impact is reasonable given the extreme case considered.

Wall clock time and memory overhead (for 500 timestep run) variation with different core counts (fixed 8 cores/node). Grid size fixed at 64x128x128 and with 67 M particles.

Cores	Wall-clock time (sec)		GB RAM / Core
	non-in-situ	in-situ	
32	506.09	621.61	0.37
64	289.43	380.45	0.36
128	185.20	278.28	0.36

Performance and Memory Impact of In-Situ Viz

- Test runs conducted with In-Situ Viz output *every timestep* (extreme case). Run times compared with normal case where output is every 500 steps.
- Strong scaling with problem size fixed at 128x256x256, 536.9M particles.

Wall clock time and memory overhead (for 500 timestep run) variation with different core counts (fixed 8 cores/node except for 1024 core case). Grid size fixed at 128x256x256 and with 536.9M particles.

Cores	Wall-clock time (sec)		GB RAM / Core
	non-in-situ	in-situ	
128	1017.10	1072.97	0.37
256	569.82	643.63	0.37
512	354.36	453.75	0.36
1024 ‡	280.53	406.81	0.36
‡ — 16 cores per node			

In-Situ Visualization to manage data

- **In Situ Visualization implemented for H3D code using Paraview Catalyst architecture.**
- **Performance and memory overhead tests run on Gordon for both weak and strong scaling cases.**
- **Results show memory overhead is reasonable and stays constant as the core count is increased.**
- **Performance impact is 20-30% for extreme case of viz output every timestep. For practical cases it is expected to be <5%.**
- **Significant saving in data output on file system and overall processing pipeline**

5. Summary

- **Understand I/O and storage characteristics and needs of your application**
- **Understand all the storage systems of the HPC machine: local/shared/parallel/(archive), remote**
- **Choose the right file system for now**
  - As you scale up or change your applications the storage and I/O strategies need to be adapted
- **Be very careful of what is not backed up** (you back it up; HPC centers are not responsible)
- **Know that you can impact 1000s of users** (negatively or positively) **based on what you do**