An Introduction to the TeraGrid Track 2D Systems Gordon

TG11 tutorial 7/18/2011

Robert Sinkovits
Gordon Applications Lead
San Diego Supercomputer Center

Gordon is a TeraGrid resource

- Gordon is one of three TeraGrid Track 2D systems
  - Award made in 2009
  - Prototype (Dash) available as TG resource since 4/1/2010
  - Full system will be ready for production 1/1/2012
  - Allocation requests accepted 9/15-10/15 for consideration at December TRAC meeting

UCSD SDSC: Design, Deployment, Support

Intel: Sandy Bridge processors, Motherboards, Flash drives

APPRO HPC Cluster Solutions: Integrator

ScaleMP: vSMP Foundation

NSF: Funding, OCI #0910847

Mellanox Technologies: 3D Torus

Why Gordon?

Designed for data and memory intensive applications that don't run well on traditional distributed memory machines

- Large shared memory requirements
- Serial or threaded (OpenMP, Pthreads)
- Limited scalability
- High performance data base applications
- Random I/O combined with very large data sets

Gordon Overview

- 1024 dual socket compute nodes

$$1024\ nodes \times 2\ \frac{sockets}{node} \times 6\ \frac{cores}{socket} \times 8\ \frac{flops}{core / cycle} \times 2.0\ GHz \approx 200\ TFlops *$$

$$1024\ nodes \times 64\ \frac{GB}{node} = 64\ TB\ DRAM$$

* Likely to be higher

- 64 I/O nodes

$$64\ nodes \times 16\ \frac{flash\ drives}{node} \times 300\ \frac{GB}{node} = 300\ TB\ \ flash\ memory$$

- Dual rail 3D torus InfiniBand QDR network

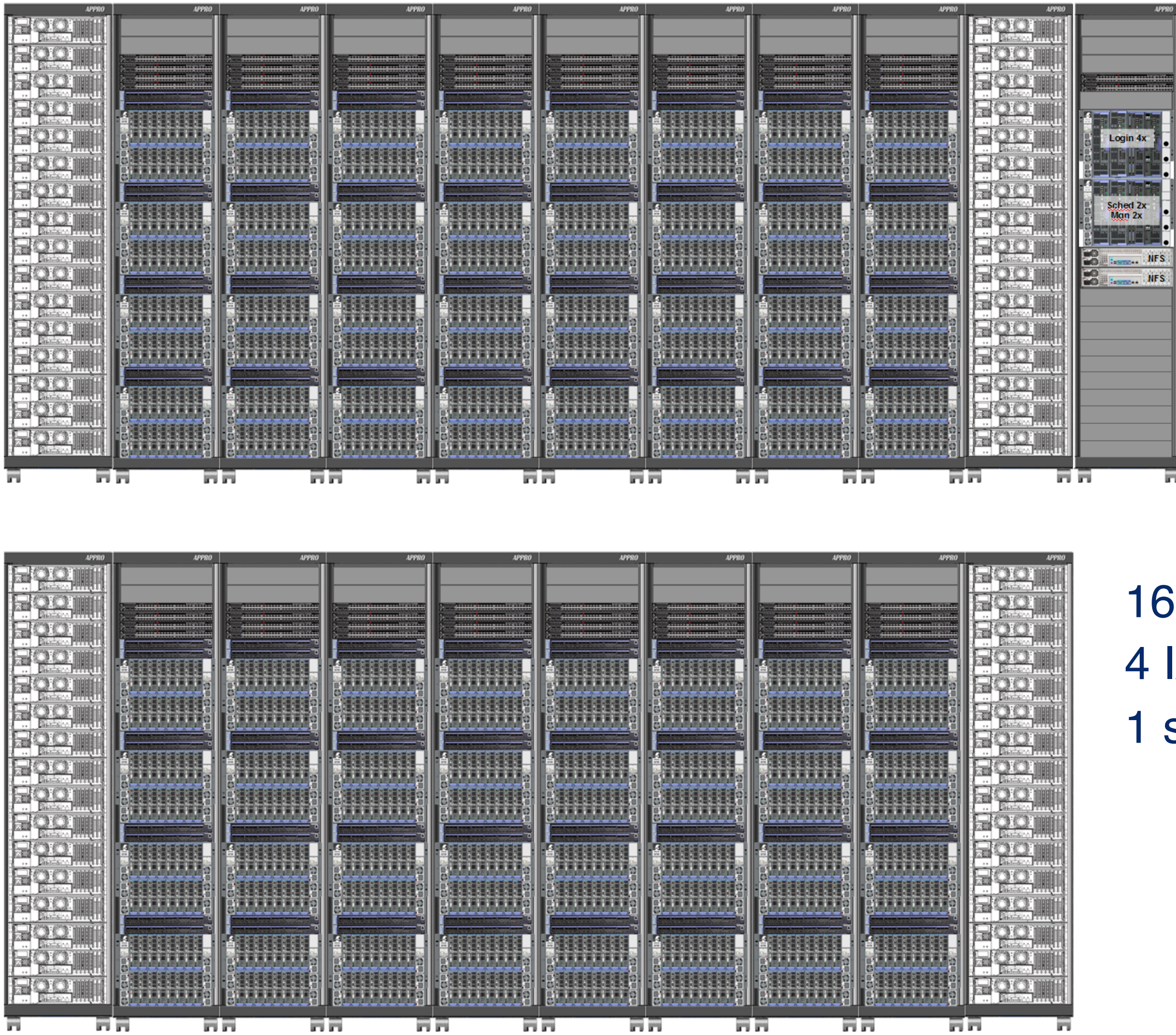
- Access to 4 PB Lustre-based parallel file system
  - Capable of delivering 100 GB/s to Gordon

Gordon is about more than raw compute power, but …

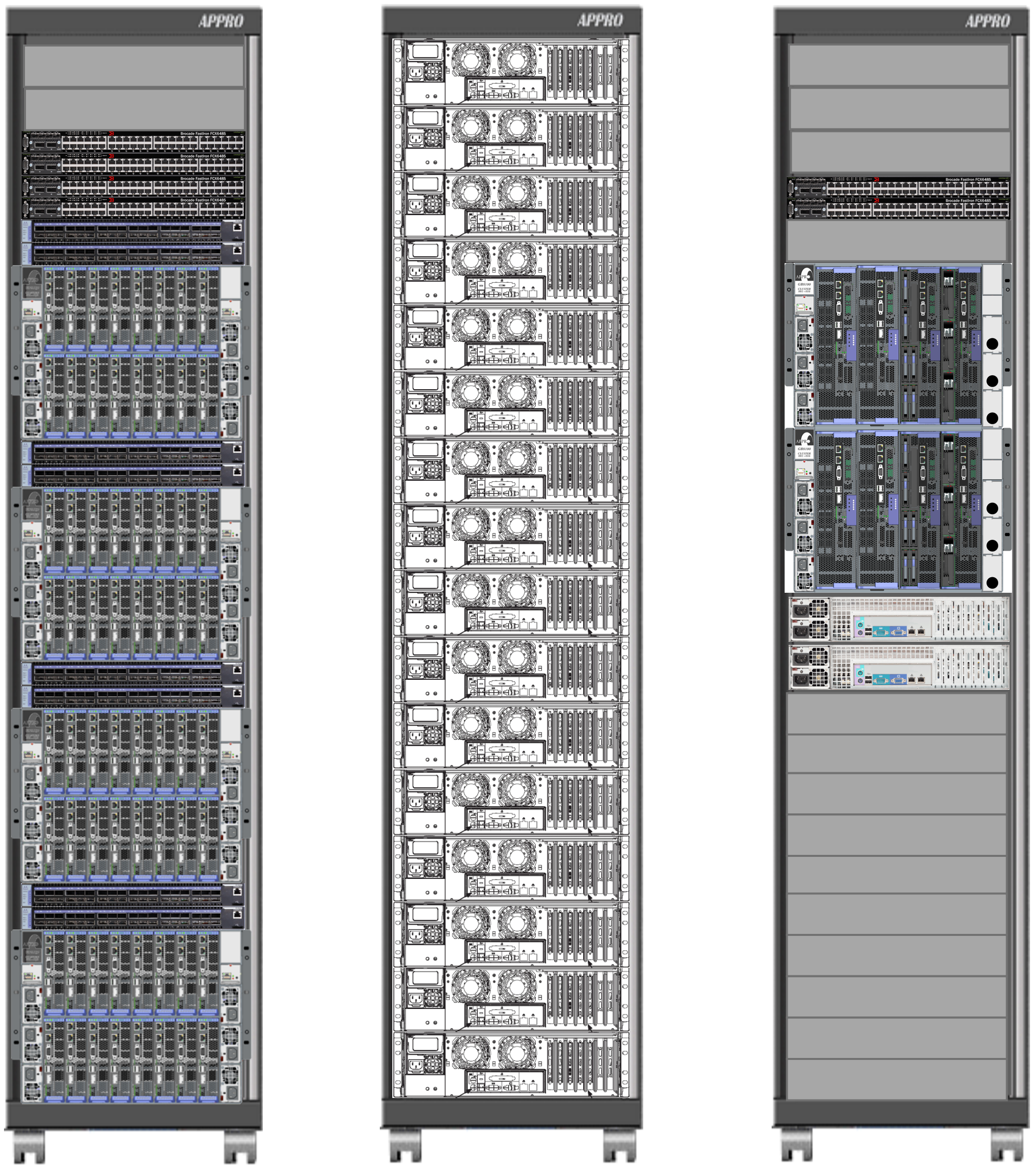
Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	548352	8162.00	8773.63
2	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C / 2010 NUDT	186368	2566.00	4701.00
38	Japan Atomic Energy Agency (JAEA) Japan	BX900 Xeon X5570 2.93GHz , Infiniband QDR / 2009 Fujitsu	17072	191.40	200.08
39	King Abdullah University of Science and Technology Saudi Arabia	Shaheen - Blue Gene/P Solution / 2009 IBM	65536	190.90	222.82
40	Shanghai Supercomputer Center China	Magic Cube - Dawning 5000A, QC Opteron 1.9 Ghz, Infiniband, Windows HPC 2008 / 2008 Dawning	30720	180.60	233.47
41	Government France	Cluster Platform 3000 BL2x220, L54xx 2.5 Ghz, Infiniband / 2009 Hewlett-Packard	24704	179.63	247.04
42	Taiwan National Center for High-performance Computing Taiwan	ALPS - Acer AR585 F1 Cluster, Opteron 12C 2.2GHz, QDR infiniband / 2011 Acer Group	26244	177.10	231.86
43	EDF R&D France	Ivanhoe - iDataPlex, Xeon X56xx 6C 2.93 GHz, Infiniband / 2010 IBM	16320	168.80	191.27
44	Swiss Scientific Computing Center (CSCS) Switzerland	Monte Rosa - Cray XT5 SixCore 2.4 GHz / 2009 Cray Inc.	22032	168.70	211.51

A **conservative** estimate of core count and clock speed probably puts Gordon around #30-40 on the Top 500 list

Gordon Rack Layout



16 compute node racks
4 I/O node racks
1 service rack

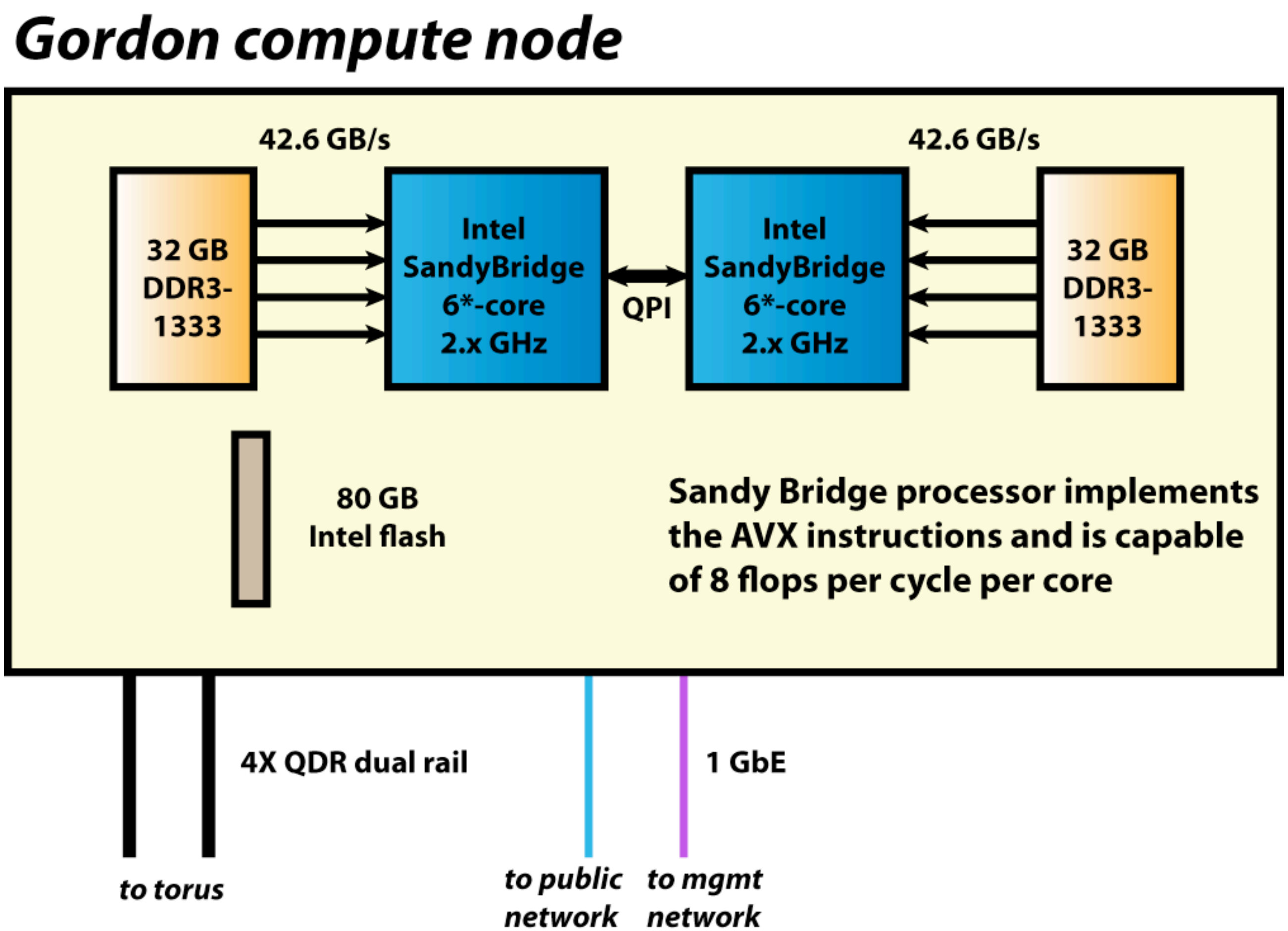


CN Rack ION Rack Service Nodes Rack

Compute node racks:
4 Appro subracks
64 blades

ION racks:
16 Gordon I/O nodes

Service rack:
4 login nodes
2 NFS servers
2 Scheduler nodes
2 management nodes



summary

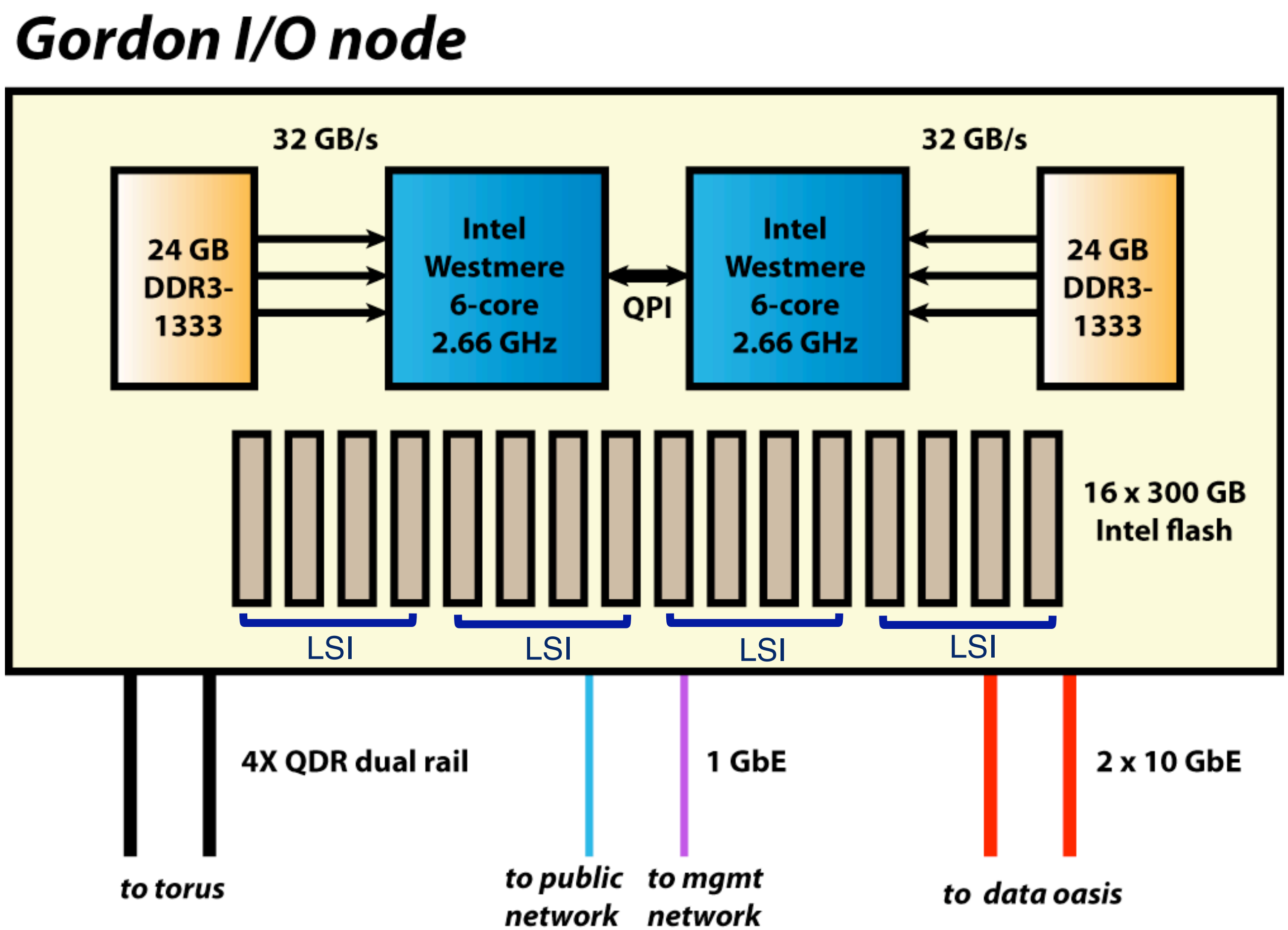
64 GB DRAM

12+ cores

2.0+ GHz

80 GB flash

For more information on AVX, see http://software.intel.com/en-us/avx/



summary

48 GB DRAM

12 cores

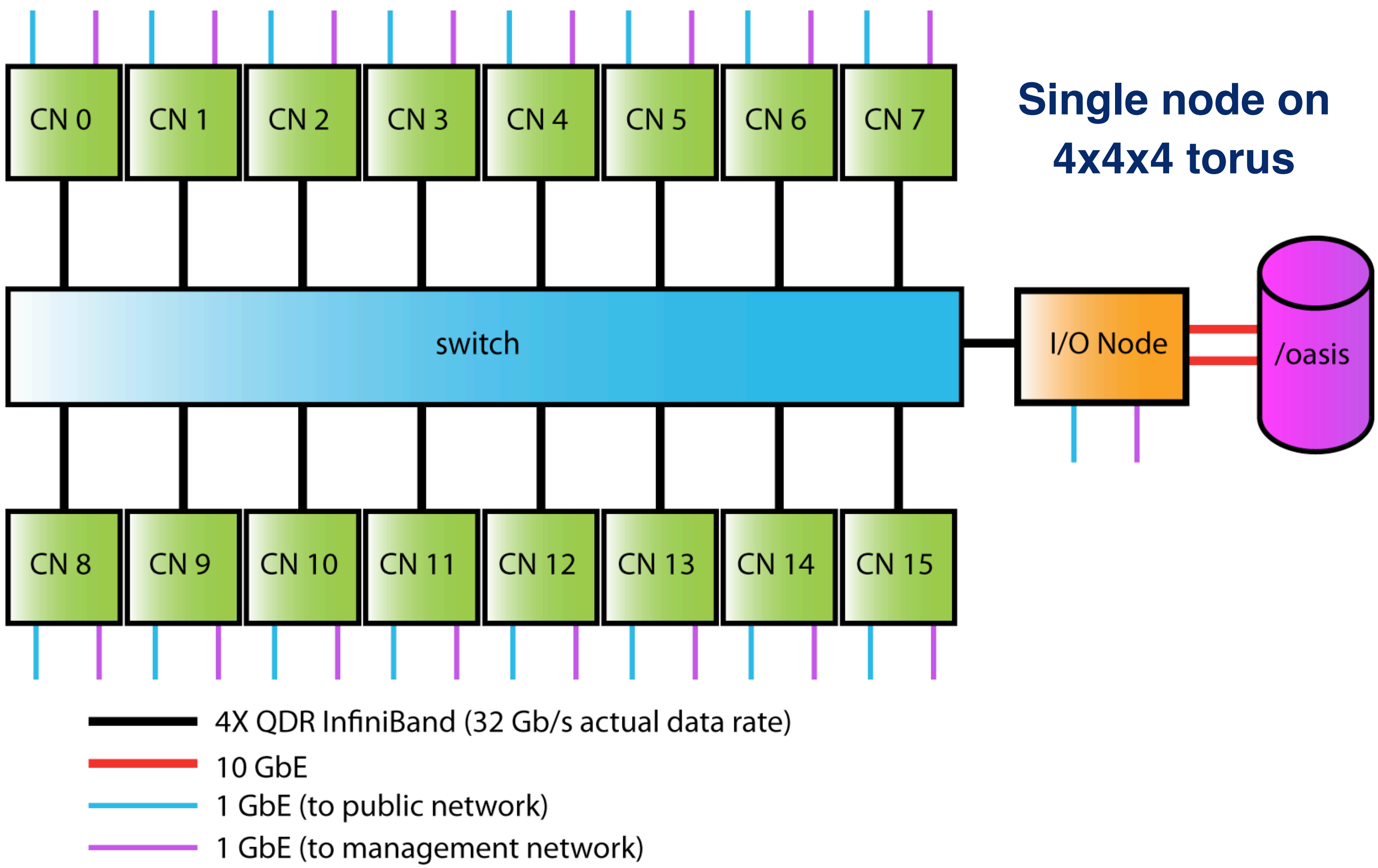
2.66 GHz

4.8 TB flash

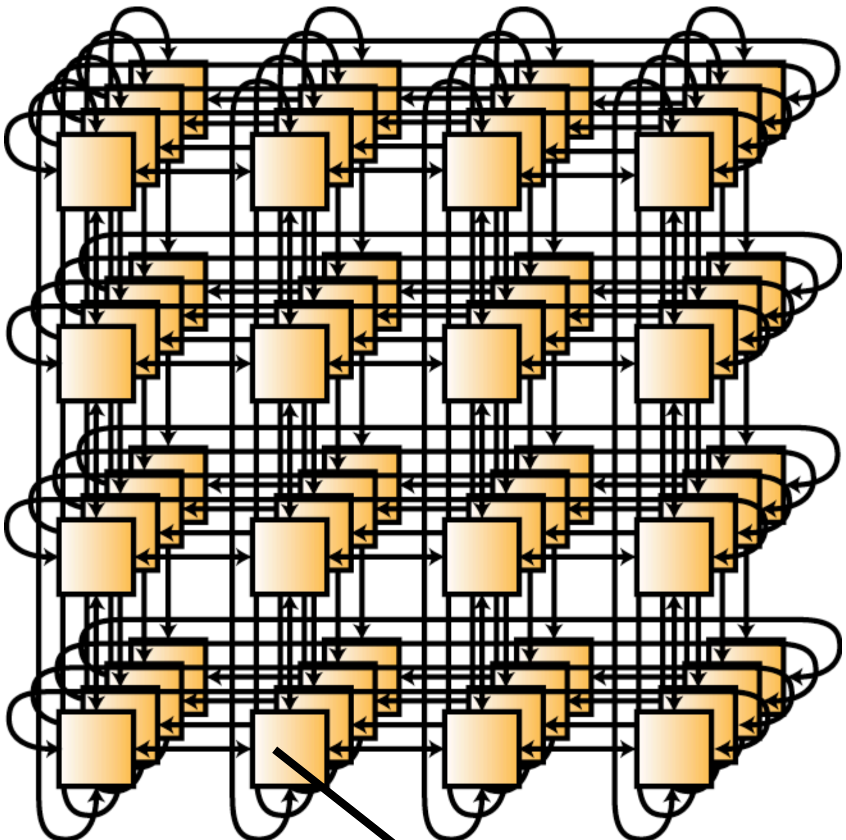
Bonded into single channel

~ 1.6 GB/s bandwidth

Simplified single rail view of Gordon connectivity showing routing between compute nodes on same switch, I/O node, and data oasis.



3D Torus Interconnect



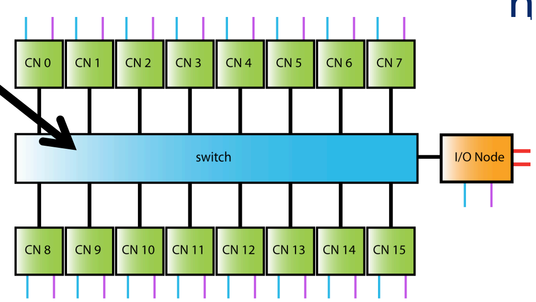
Gordon switches connected in dual rail 4x4x4 3D torus

Maximum of six hops to get from one node to furthest node in cluster

Fault tolerant, requires up to 40% fewer switches and 25-50% fewer cables than other topologies

Scheduler will be aware of torus geometry and assign nodes to jobs accordingly

Note – 3 connections between neighboring nodes, only 1 shown

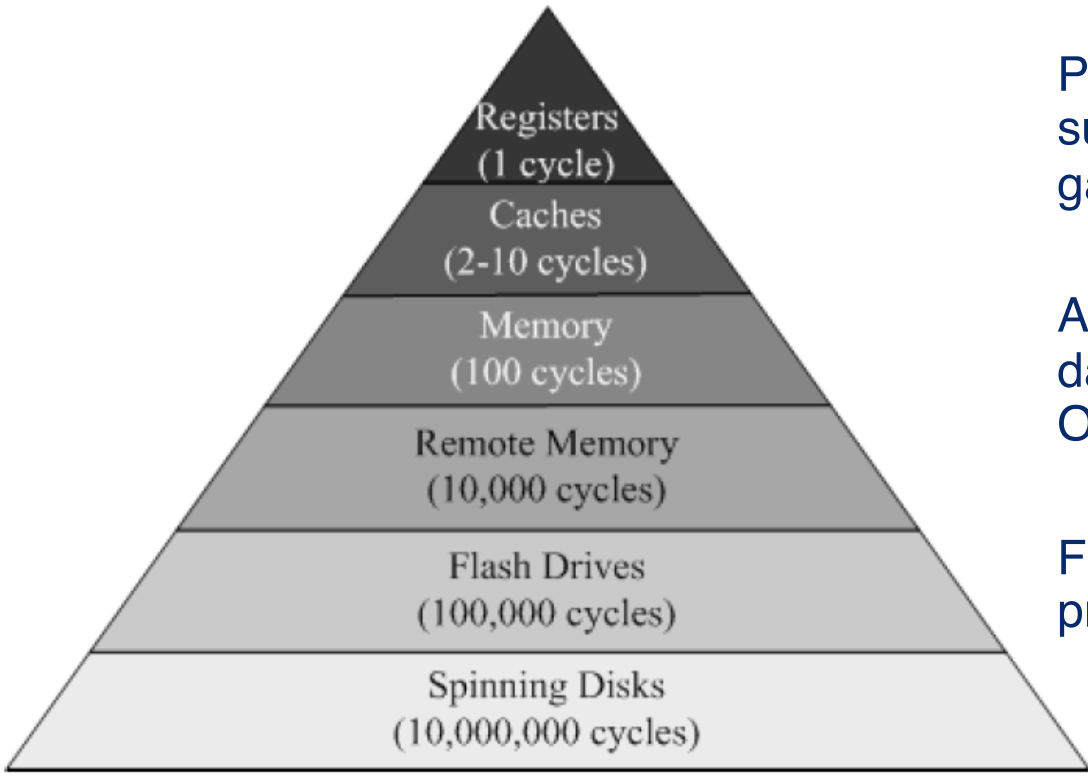


Flash drives have a number of advantages over hard disks in terms of performance, reliability, and range of operating conditions

	flash	HDD
latency	✔	
bandwidth	✔	
power consumption	✔	
storage density	✔	
stability	✔	
price per unit		✔
total cost of ownership	?	?

Besides price, the one drawback of the flash drives is that they have a limited endurance (number of times a memory cell can be written and erased). Fortunately, the technological gains (better NAND gates, wear leveling algorithms, etc.) are improving endurance

For data intensive applications, the main advantage of flash is the low latency



Performance of the memory subsystem has not kept up with gains in processor speed

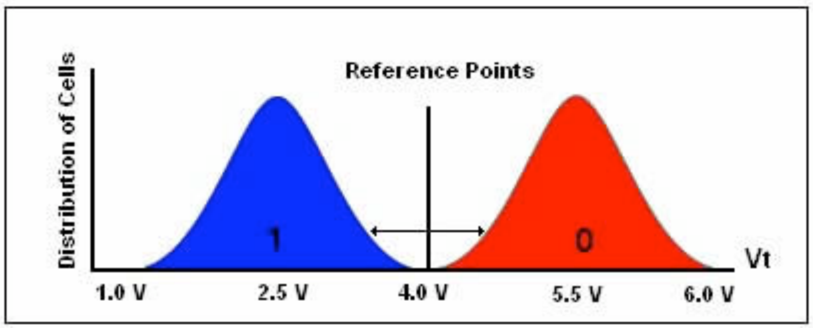
As a result, latencies to access data from hard disk are O(10,000,000) cycles

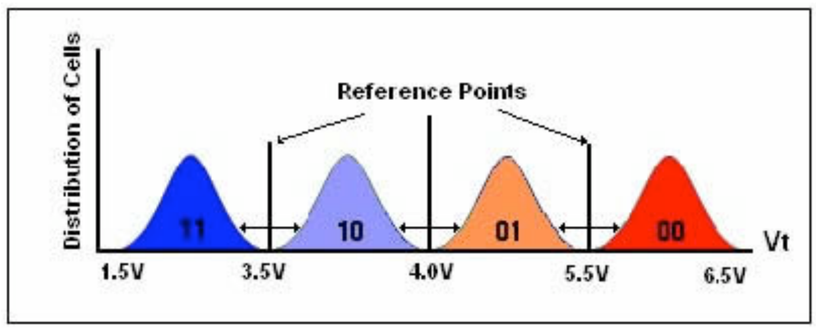
Flash memory fills this gap and provides O(100) lower latency

Flash memory comes in two varieties: SLC and MLC

Value	State
0	Programmed
1	Erased

Value	State
00	Fully Programmed
01	Partially Programmed
10	Partially Erased
11	Fully Erased





SLC - single-level cell
1 bit/cell = 2 values/cell

lower storage density
more expensive
higher endurance

MLC – multi-level cell
2 bit/cell = 4 values/cell

higher storage density
less expensive
lower endurance

Intel flash drives to be used in Gordon are similar to the Postville Refresh drives but will be based on enterprise MLC (eMLC) technology and have a higher endurance than consumer grade drives

	Intel X25-M G2 (34nm)	Intel X25-M G3 (25nm)
Codename	Postville	Postville Refresh
Capacities	80/160GB	80/160/300/600GB
NAND	IMFT 34nm MLC	IMFT 25nm MLC
Sequential Performance Read/Write	Up to 250/100 MB/s	Up to 250/170 MB/s
Random 4KB Performance Read/Write	Up to 35K/8.6K IOPS	Up to 50K/40K IOPS
Max Power Consumption Active/Idle	3.0/0.06W	6.0/0.075W
Total 4KB Random Writes (Drive Lifespan)	7.5TB - 15TB	30TB - 60TB
Power Safe Write Cache	No	Yes
Form Factors	1.8" & 2.5"	1.8" & 2.5"

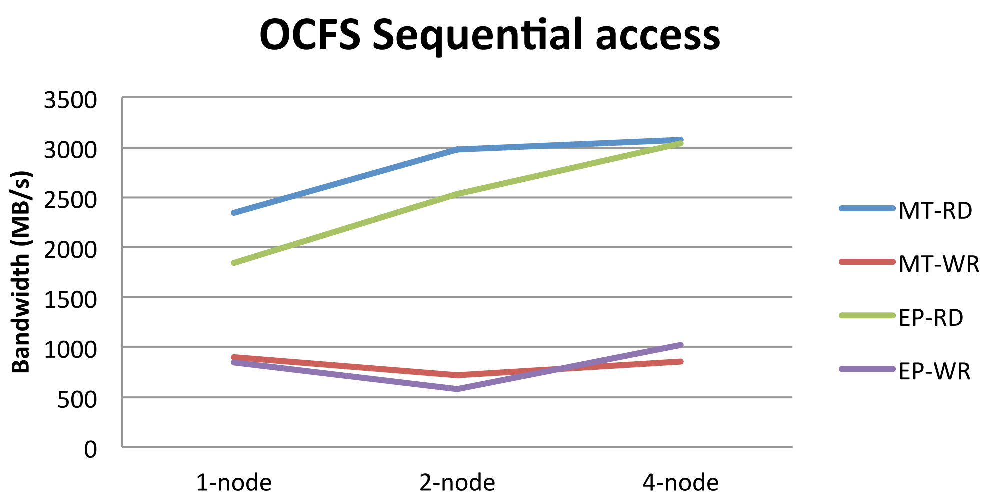
Flash performance testing – configuration

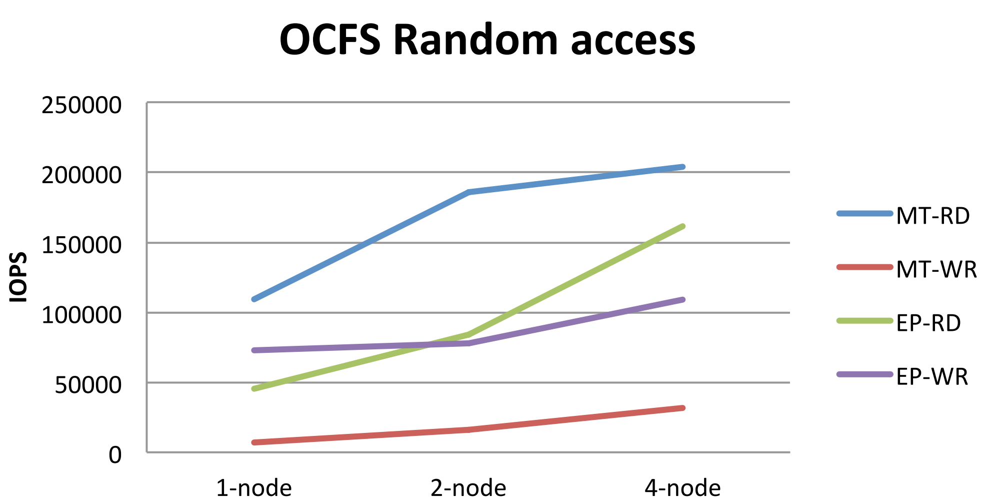
- One server with 16 Intel Postville Refresh drives
- Four clients
- All five nodes contain two hex-core Westmere processors
- Clients/servers connected using DDR InfiniBand
- iSER (iSCSI over RDMA) protocol

- OCFS testing – 16 flash drive configured as a single RAID 0 device

- XFS testing – one flash drive exported to each client

Flash performance – parallel file system



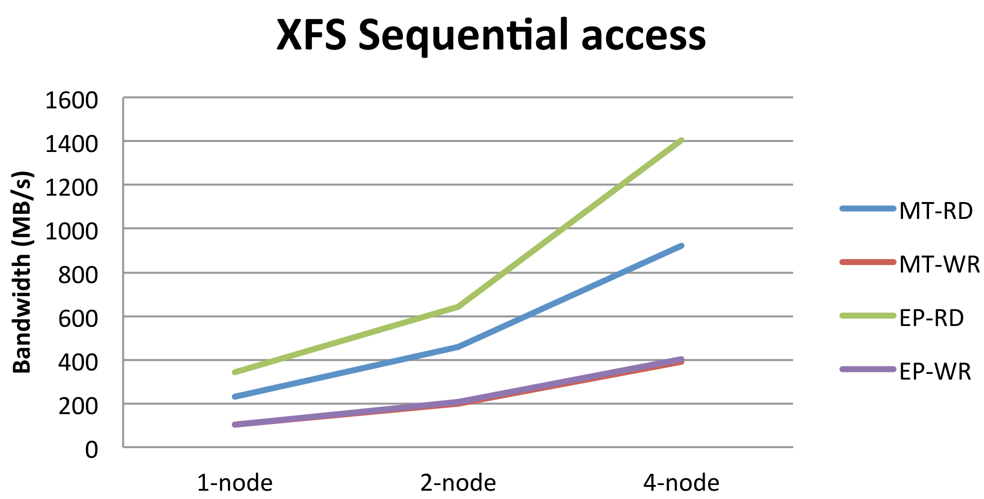


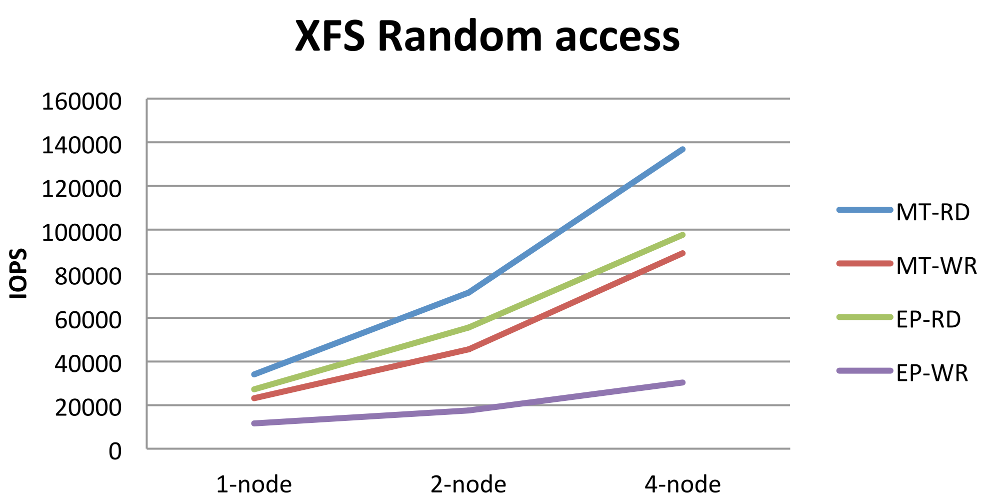
Performance of Intel Postville Refresh SSDs
(16 drives → RAID 0)
with OCSF (Oracle Cluster File System)

I/O done simultaneously from 1, 2, or 4 compute nodes

MT = multi-threaded
EP = embarrassingly parallel

Flash performance – serial file system





Performance of Intel Postville Refresh SSDs

(4 drives, w/ one drive exported to each node)

I/O done simultaneously from 1, 2, or 4 compute nodes

MT = multi-threaded

EP = embarrassingly parallel

Flash drive – spinning disk comparisons

Intel X25-M flash drives (160 GB)

vs.

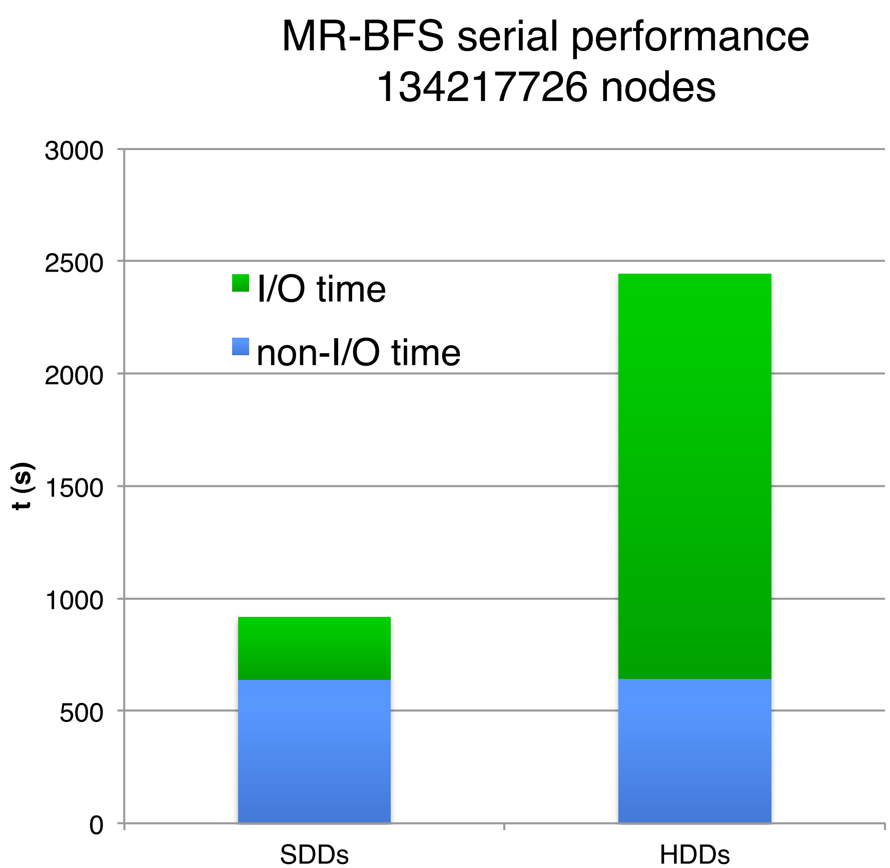
Seagate Momentus hard drives
(SATA, 7200 RPM, 250 GB)

Differences between Dash and Gordon

	Dash	Gordon
InfiniBand	DDR	QDR
Network rails	single	double
Compute node processors	Nehalem	Sandy Bridge
Compute node memory	48 GB	64 GB
I/O node flash	Postville Refresh	Intel eMLC
I/O node memory	24 GB	48 GB
vSMP foundation version	3.5.175.17	?
Resource management	Torque	SLURM

When considering benchmark results and scalability, keep in mind that nearly every major feature of Gordon will be an improvement over Dash. As user note that there will be differences in the environment

Flash case study – Breadth First Search



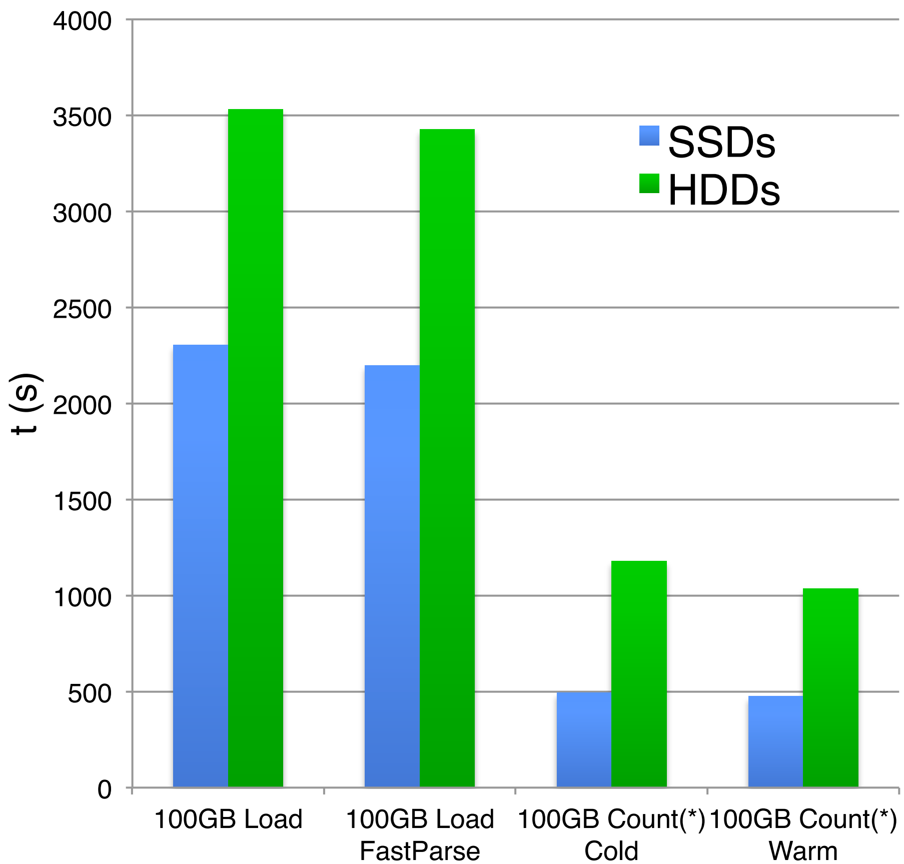
Implementation of Breadth-first search (BFS) graph algorithm developed by Munagala and Ranade

Benchmark problem: BFS on graph containing 134 million nodes

Use of flash drives reduced I/O time by factor of 6.5x. As expected, no measurable impact on non-I/O operations

Problem converted from I/O bound to compute bound

Flash case study – LIDAR

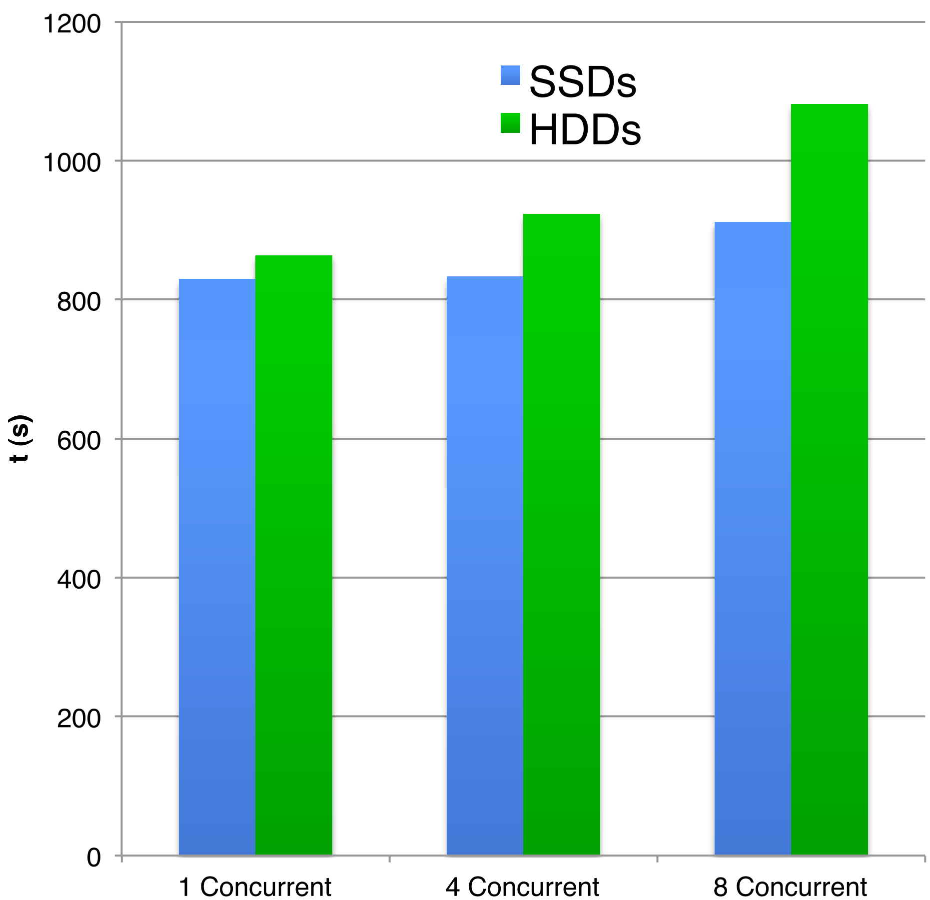


Remote sensing technology used to map geographic features with high resolution

Benchmark problem: Load 100 GB data into single table, then count rows. DB2 database instance

Flash drives 1.5x (load) to 2.4x (count) faster than hard disks

Flash case study – LIDAR

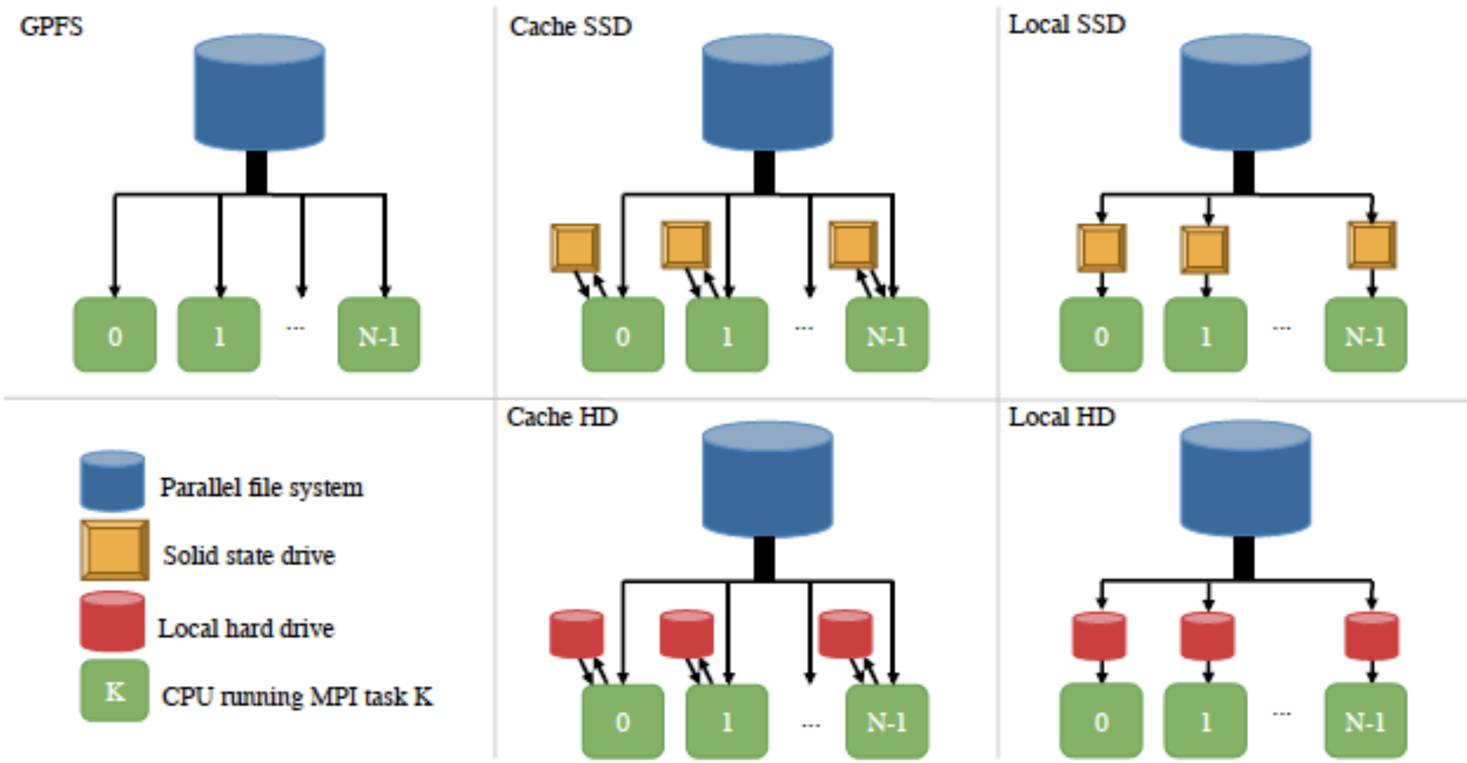


Remote sensing technology used to map geographic features with high resolution

Comparison of runtimes for concurrent LIDAR queries obtained with flash drives (SSD) and hard drives (HDD) using the Alaska Denali-Totschunda data collection.

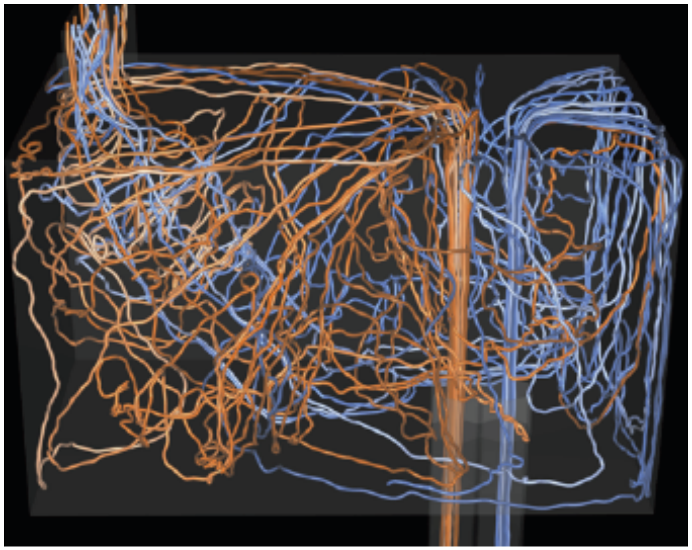
Impact of SSDs was modest, but significant when executing multiple simultaneous queries

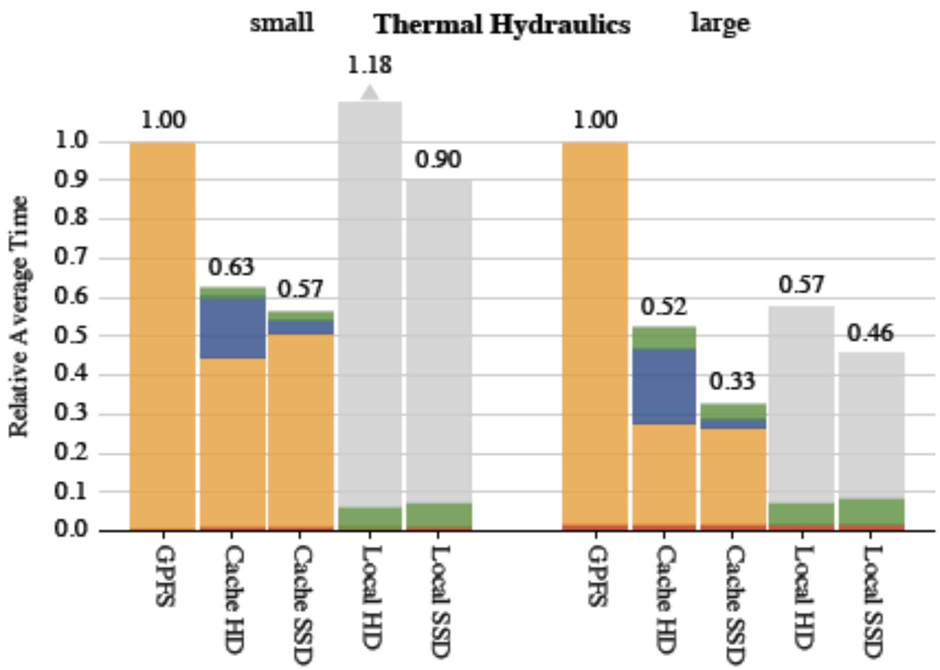
Flash case study – Parallel Streamline Visualization



Camp et al, accepted to IEEE Symp. on Large-Scale Data Analysis and Visualization (LDAV 2011)

Flash case study – Parallel Streamline Visualization

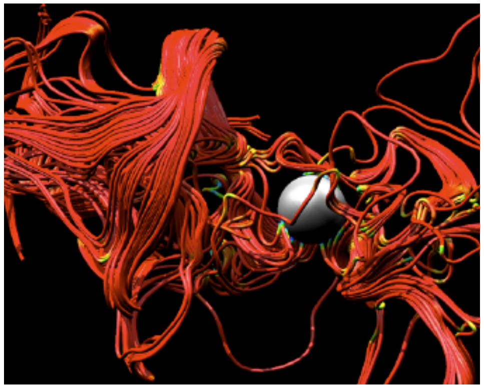


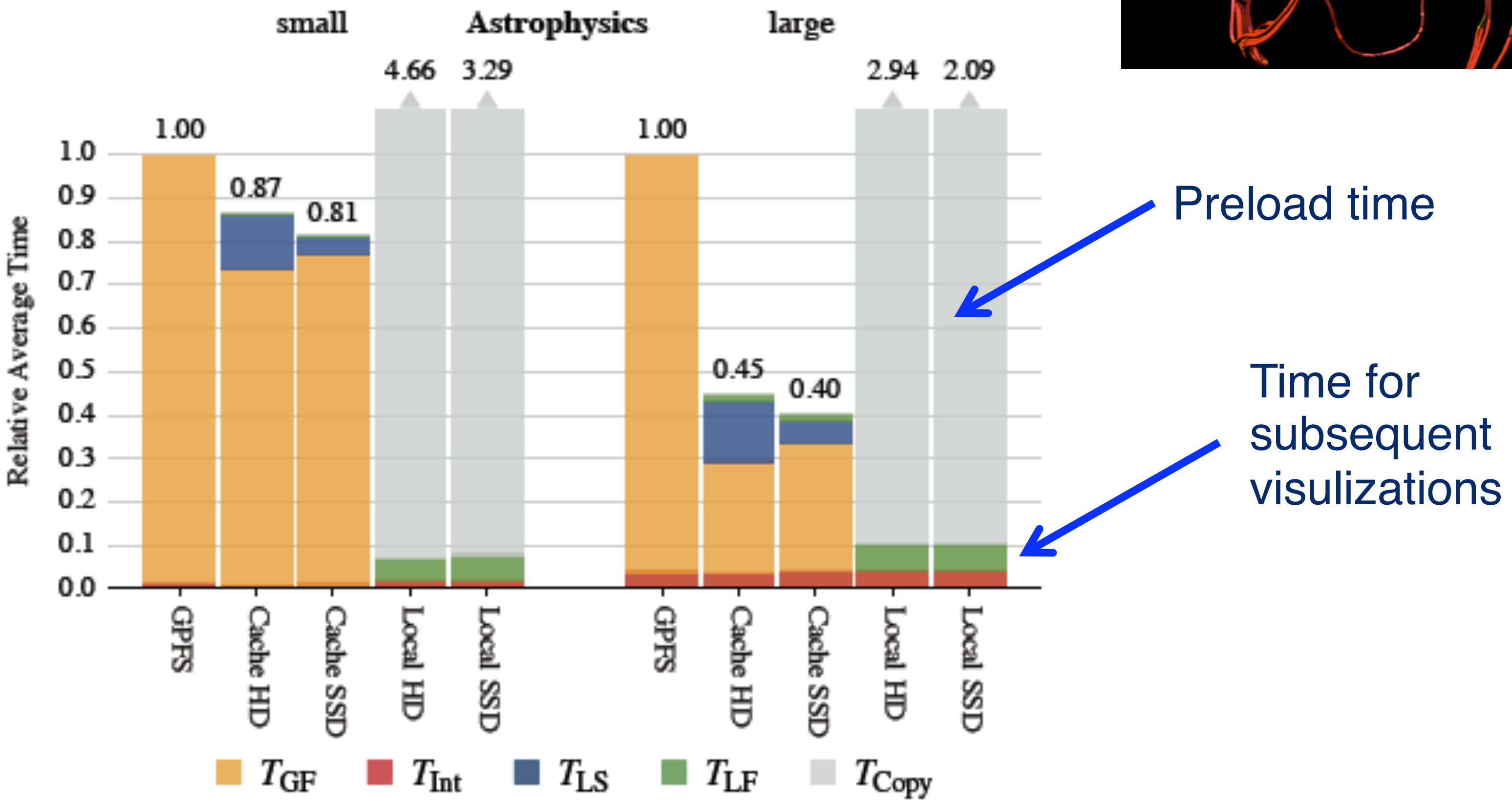


Caching data to drives results in better performance than reading directly from GPFS or preloading into local disk. SSDs perform better than HDDs

Camp et al, accepted to IEEE Symp. on Large-Scale Data Analysis and Visualization (LDAV 2011)

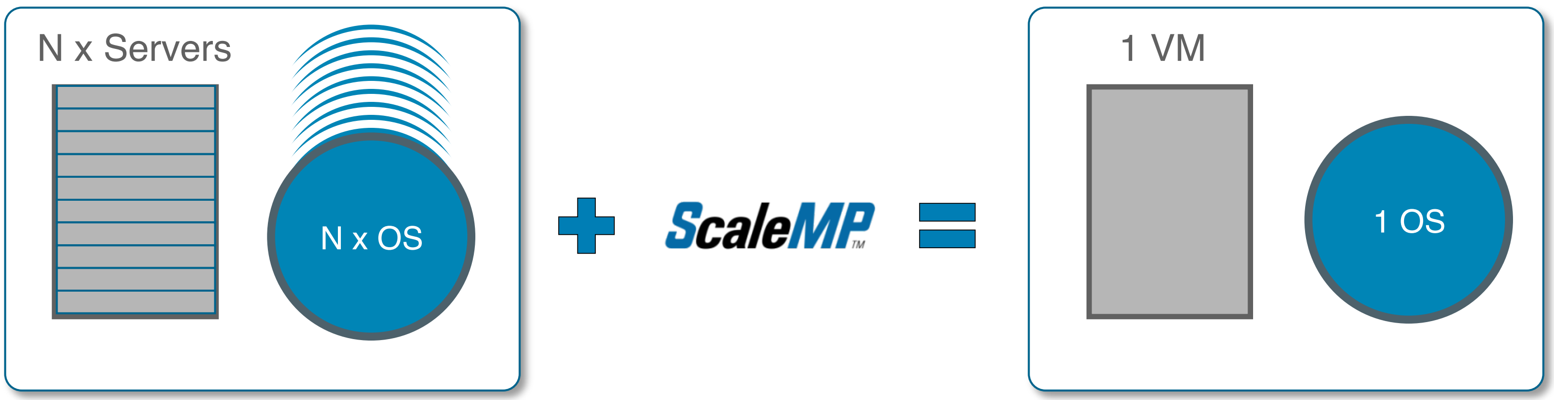
Although preloading entire data set into flash typically takes longer than just reading from GPFS, still worth doing if multiple visualizations will be performed while data is in flash





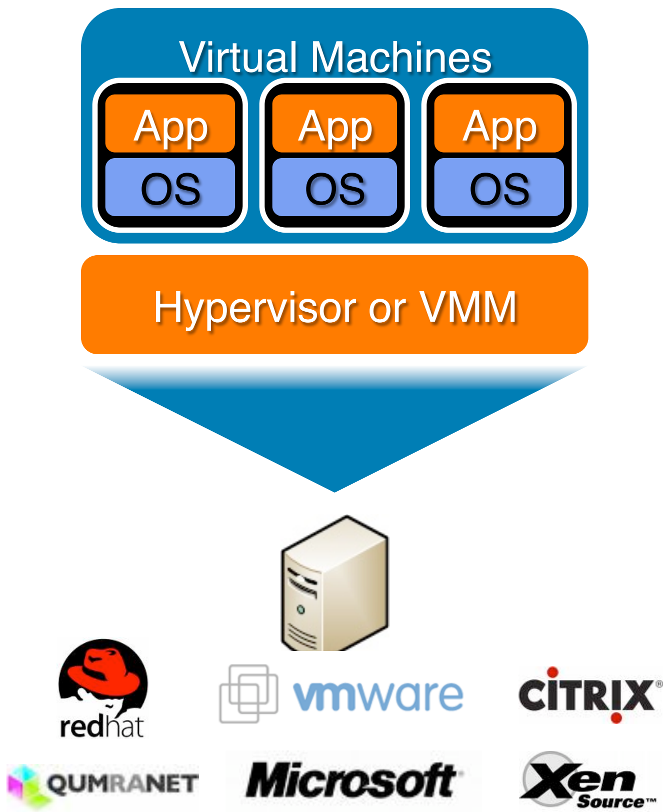
Camp et al, accepted to IEEE Symp. on Large-Scale Data Analysis and Visualization (LDAV 2011)

Introduction to vSMP

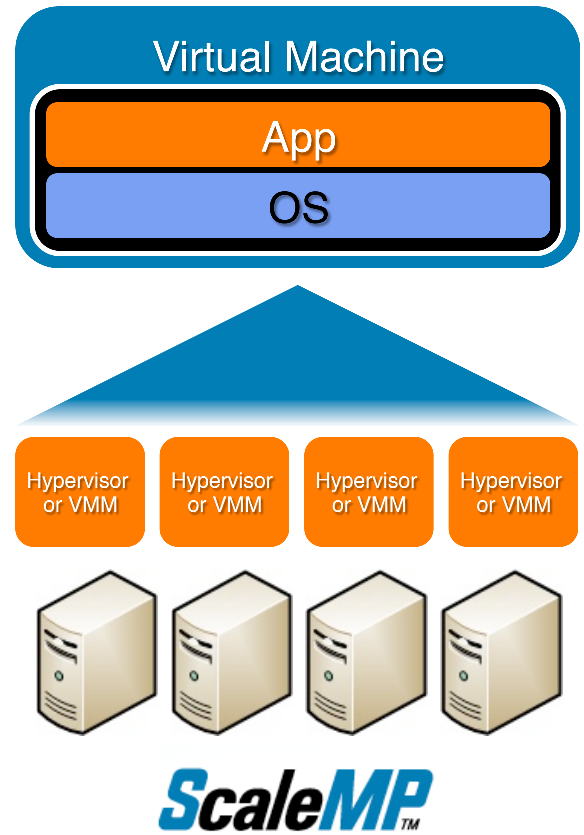


Virtualization software for aggregating multiple off-the-shelf systems into a single virtual machine, providing improved usability and higher performance

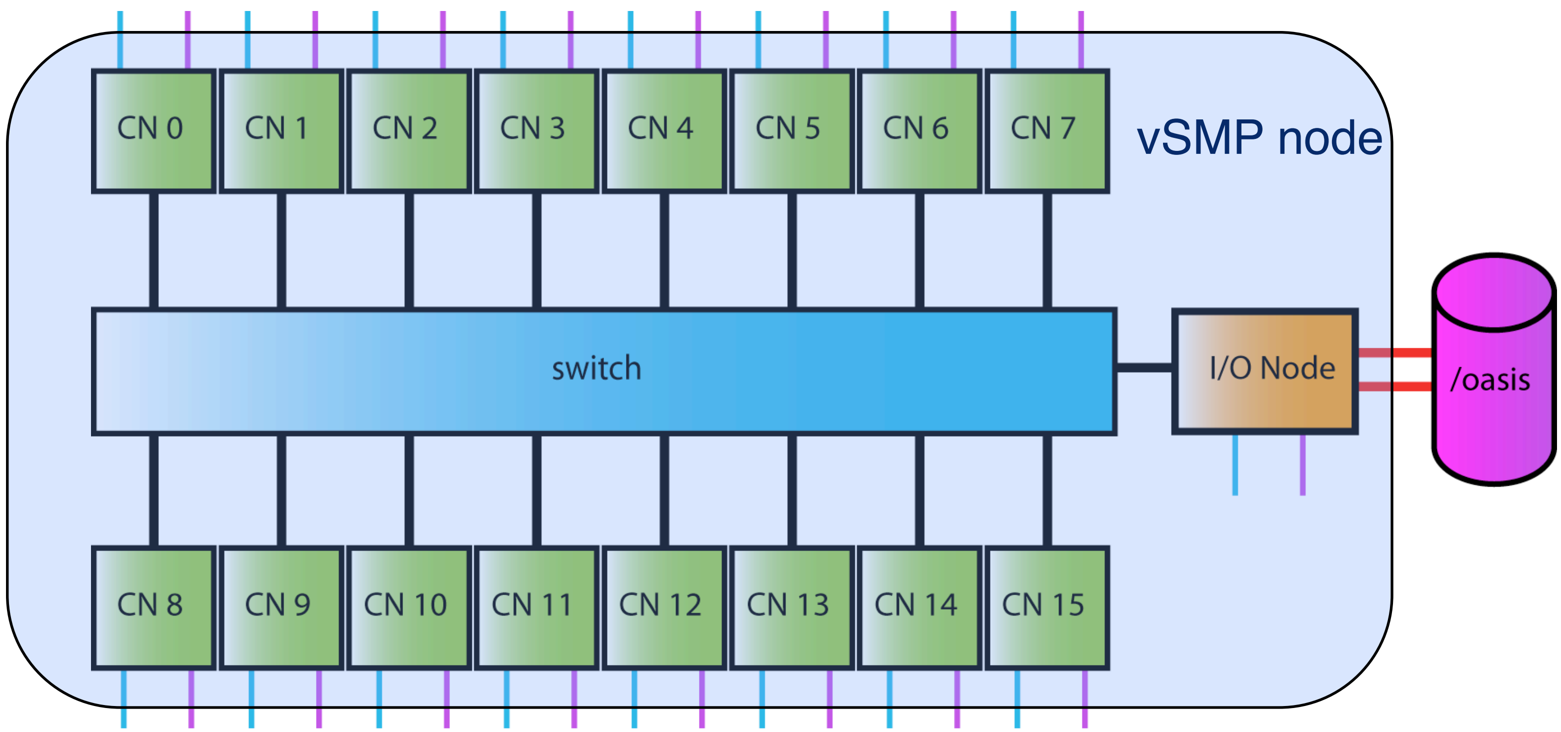
PARTITIONING



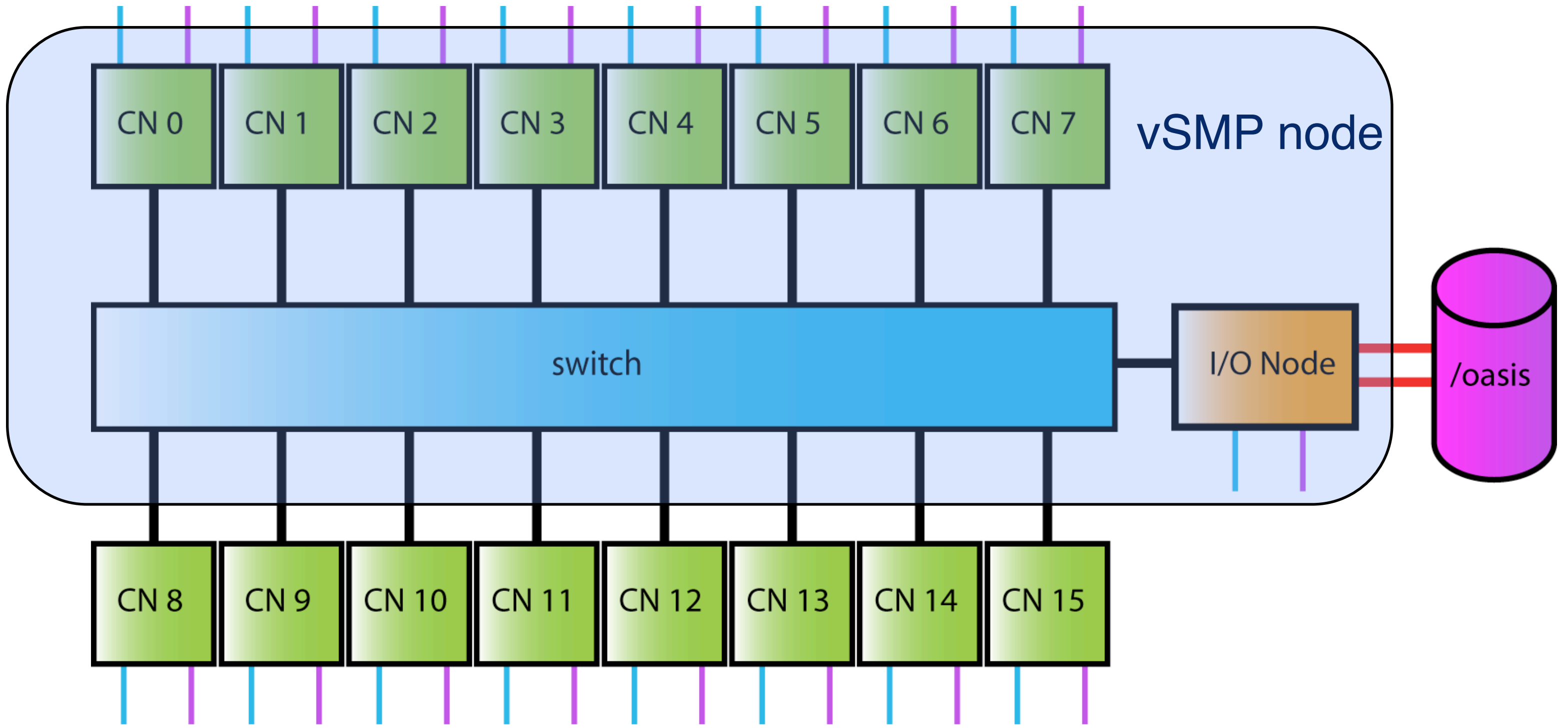
AGGREGATION



vSMP node configured from 16 compute nodes and one I/O node



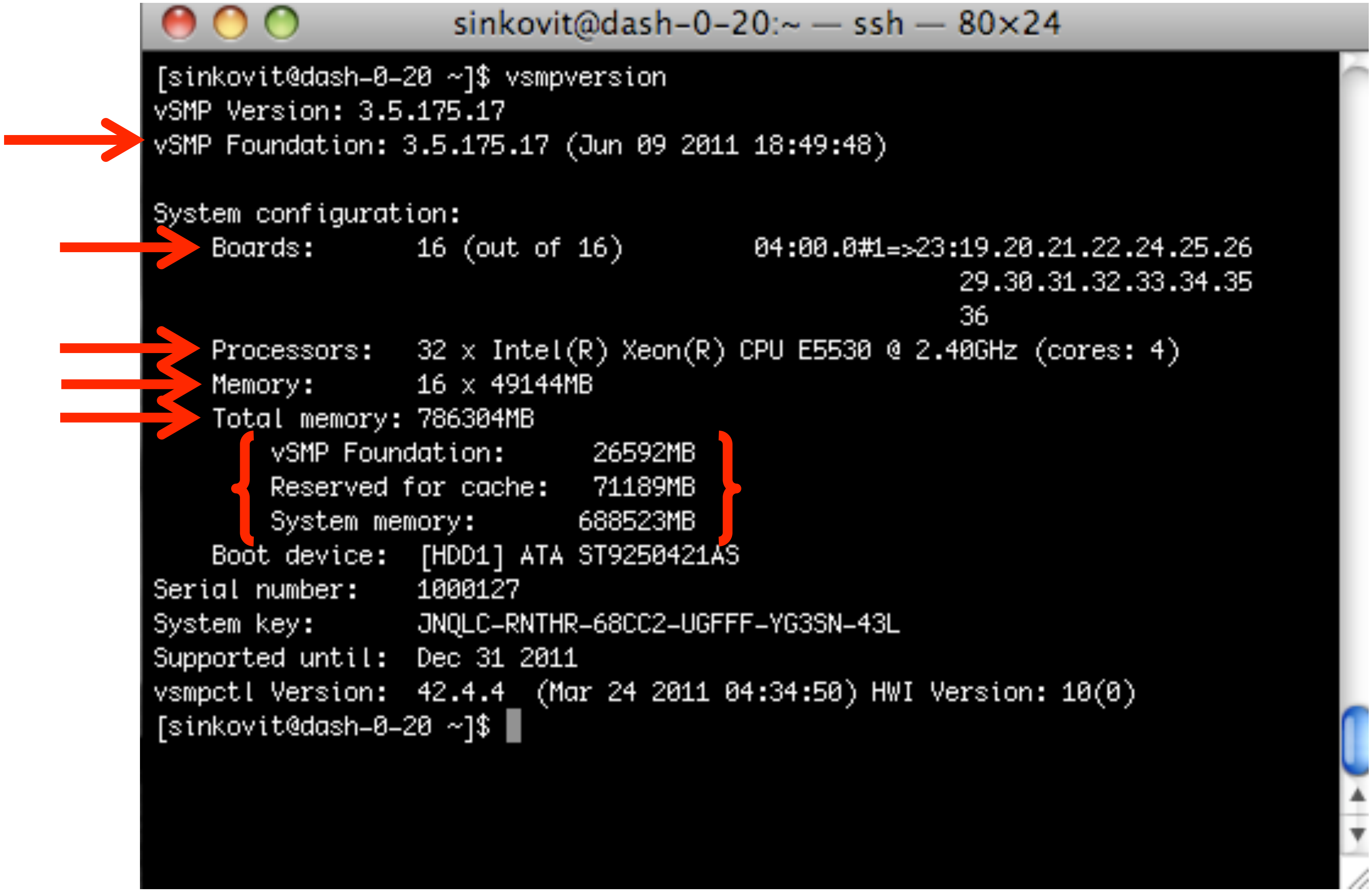
To user, logically appears as a single, large SMP node vSMP node configured from 8 compute nodes and one I/O node



The vSMP foundation software provides flexibility in configuring the system. Compute nodes 8-15 will be available for non-vSMP jobs

Investigating use of cpusets to run multiple jobs within a 16-way vSMP nodes, so may not pursue this option

Overview of a vSMP node



```
[sinkovit@dash-0-20 ~]$ vsmpversion
vSMP Version: 3.5.175.17
vSMP Foundation: 3.5.175.17 (Jun 09 2011 18:49:48)

System configuration:
    Boards:       16 (out of 16)          04:00.0#1=>23:19.20.21.22.24.25.26
                                                     29.30.31.32.33.34.35
                                                     36
    Processors:   32 x Intel(R) Xeon(R) CPU E5530 @ 2.40GHz (cores: 4)
    Memory:       16 x 49144MB
    Total memory: 786304MB
        vSMP Foundation:       26592MB
        Reserved for cache:    71189MB
        System memory:        688523MB
    Boot device:  [HDD1] ATA ST9250421AS
Serial number:    1000127
System key:       JNQLC-RNTHR-68CC2-UGFFF-YG3SN-43L
Supported until:  Dec 31 2011
vsmpctl Version:  42.4.4  (Mar 24 2011 04:34:50) HWI Version: 10(0)
[sinkovit@dash-0-20 ~]$
```

Overview of a vSMP node

```
[sinkovit@dash-0-20 ~]$ grep processor /proc/cpuinfo | tail -5
processor       : 123
processor       : 124
processor       : 125
processor       : 126
processor       : 127
[sinkovit@dash-0-20 ~]$
```

/proc/cpuinfo indicates 128 processors
(16 nodes x 8 cores/node = 128)

```
Tasks: 1893 total,   1 running, 1892 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.1%sy,  0.0%ni, 99.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  695301784k total,  2475088k used, 692826696k free,    21824k buffers
Swap:        0k total,        0k used,        0k free,   202432k cached
```

Top shows 663 GB memory (16 nodes x 48 GB/node = 768 GB)
Difference due to vSMP overhead

Making effective use of vSMP

While vSMP does provide a flexible, cost-effective solution for hardware aggregation. Care must be taken to get the best performance

- Control placement of threads to compute cores
- Link optimized versions of MPICH2 library
- Use libhoard for dynamic memory management
- Follow application specific guidelines from ScaleMP
- Performance depends heavily on memory access patterns

In many cases, little or no modifications at the source code level are required to run applications effectively on vSMP nodes

Making effective use of vSMP

The Hoard memory allocator is a fast, scalable, and memory-efficient memory allocator for Linux, Solaris, Mac OS X, and Windows. Hoard is a drop-in replacement for malloc that can **dramatically improve application performance, especially for multithreaded programs running on multiprocessors and multicore CPUs.** No source code changes necessary: just link it in or set one environment variable (from www.hoard.org)

export LD_PRELOAD="/usr/lib/libhoard.so"

threads	w/ libhoard	w/o libhoard
1	607	625
2	310	328
4	173	199
8	119	121

Timing results for MOPS run under vSMP (3.5.175.17).

With older versions of vSMP, impact of libhoard was much greater.

Continuing to see vSMP improvements as we work closely with ScaleMP numabind evaluates all possible contiguous sets of compute cores and determines set with best placement cost

- cores span minimum number or nodes
- cores chosen with lowest load averages

KMP_AFFINITY specifies preferred assignment of threads to the selected set of cores

```
export KMP_AFFINITY=compact,verbose,0,`numabind --offset 8`
```

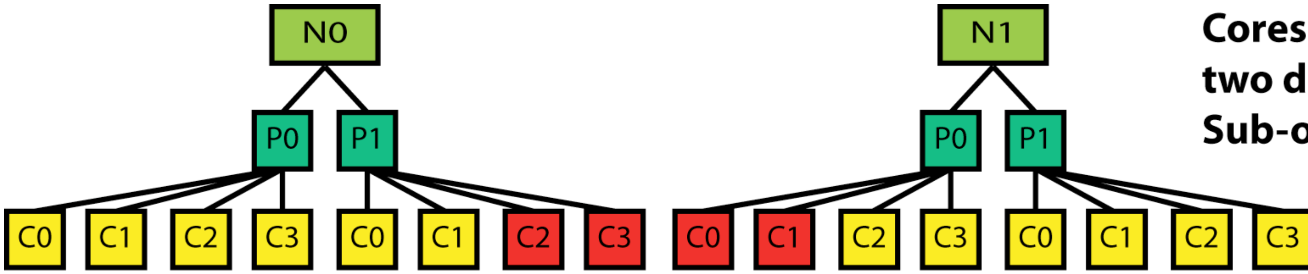
- Place threads as compactly as possible
- Be verbose
- Do not permute assignment of threads to cores
- Use this set of core (note back quotes)

export KMP_AFFINITY=compact,verbose,0,`numabind --offset 8`

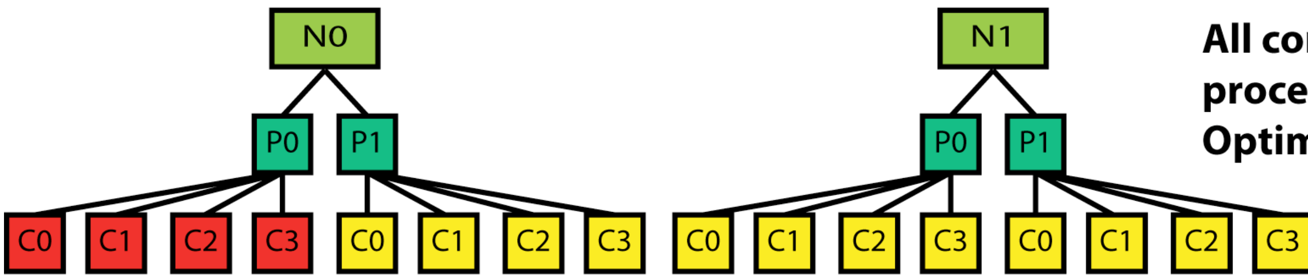
```
Placement cost for {0,1,2,3,4,5,6,7} is 1170000 (oversub 0)
Placement {1,2,3,4,5,6,7,8} is not acceptable, uses more boards than the minimum
Placement {2,3,4,5,6,7,8,9} is not acceptable, uses more boards than the minimum

[lines not shown]

Placement {118,119,120,121,122,123,124,125} is not acceptable, uses more boards than the minimum
Placement {119,120,121,122,123,124,125,126} is not acceptable, uses more boards than the minimum
Placement cost for {120,121,122,123,124,125,126,127} is 0 (oversub 0)
Best placement is {120,121,122,123,124,125,126,127}

[lines not shown]

OMP: Info #168: KMP_AFFINITY: OS proc 120 maps to package 30 core 0 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 121 maps to package 30 core 1 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 122 maps to package 30 core 2 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 123 maps to package 30 core 3 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 124 maps to package 31 core 0 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 125 maps to package 31 core 1 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 126 maps to package 31 core 2 [thread 0]
OMP: Info #168: KMP_AFFINITY: OS proc 127 maps to package 31 core 3 [thread 0]
OMP: Info #147: KMP_AFFINITY: Internal thread 0 bound to OS proc set {120}
OMP: Info #147: KMP_AFFINITY: Internal thread 4 bound to OS proc set {124}
OMP: Info #147: KMP_AFFINITY: Internal thread 5 bound to OS proc set {125}
OMP: Info #147: KMP_AFFINITY: Internal thread 6 bound to OS proc set {126}
OMP: Info #147: KMP_AFFINITY: Internal thread 3 bound to OS proc set {123}
OMP: Info #147: KMP_AFFINITY: Internal thread 7 bound to OS proc set {127}
OMP: Info #147: KMP_AFFINITY: Internal thread 2 bound to OS proc set {122}
OMP: Info #147: KMP_AFFINITY: Internal thread 1 bound to OS proc set {121}
```

numabind output

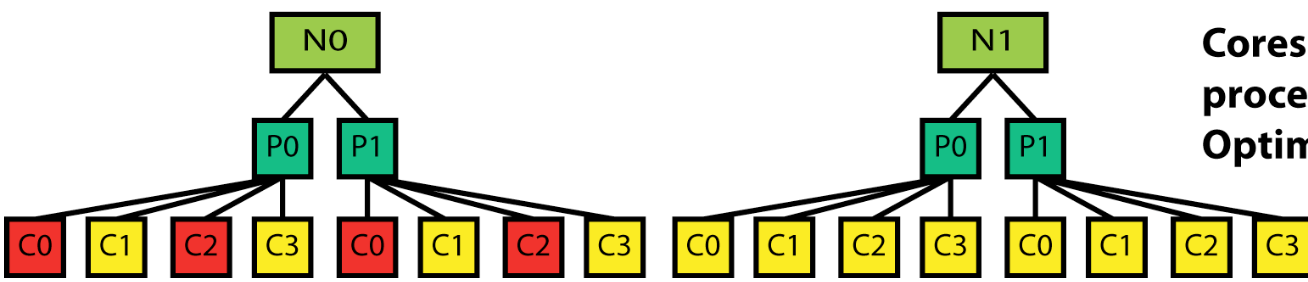
KMP_AFFINITY output



**Cores belong to
two different nodes
Sub-optimal placement**

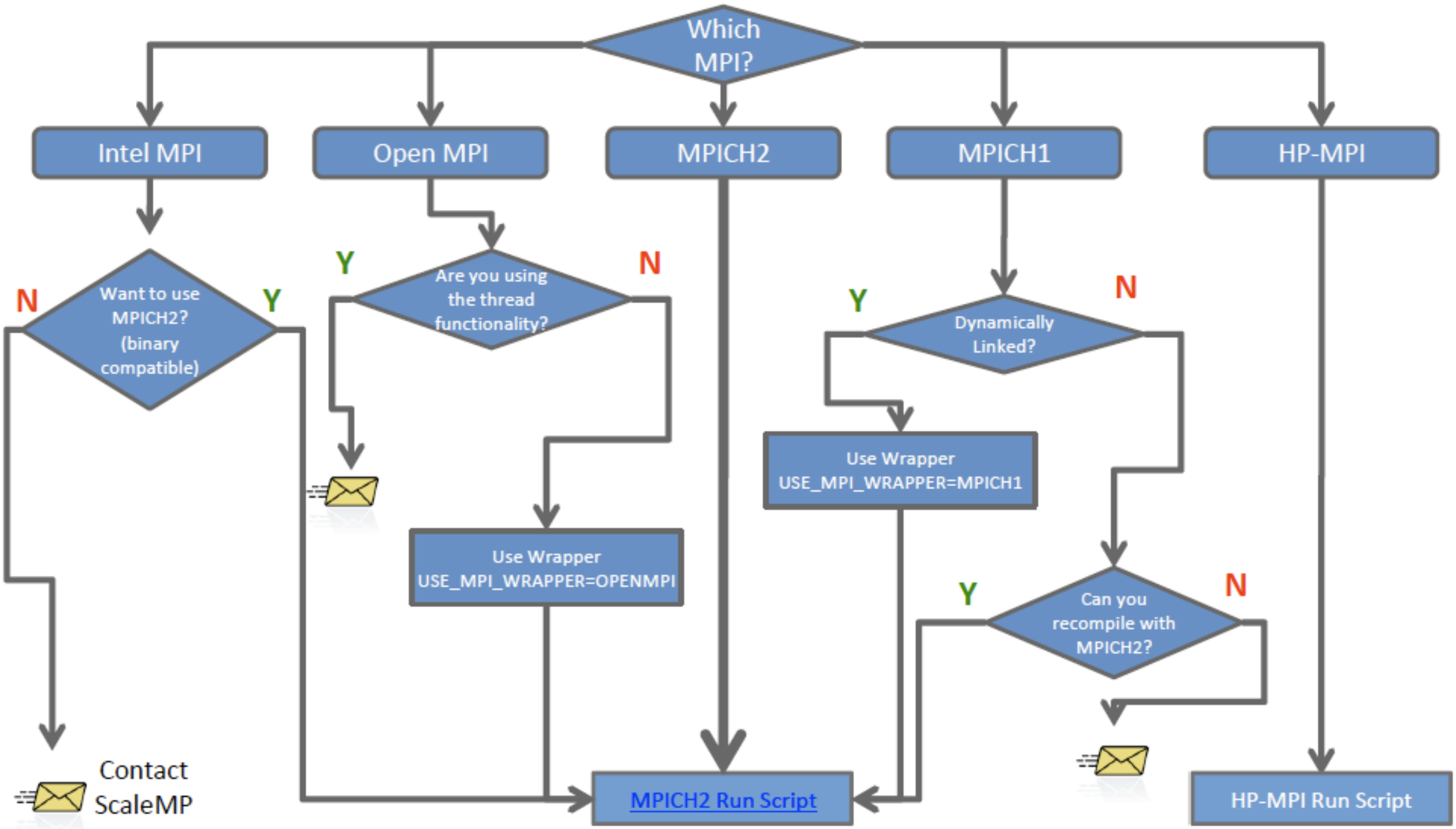


**All cores belong to same
processor on same node
Optimal compact placement**

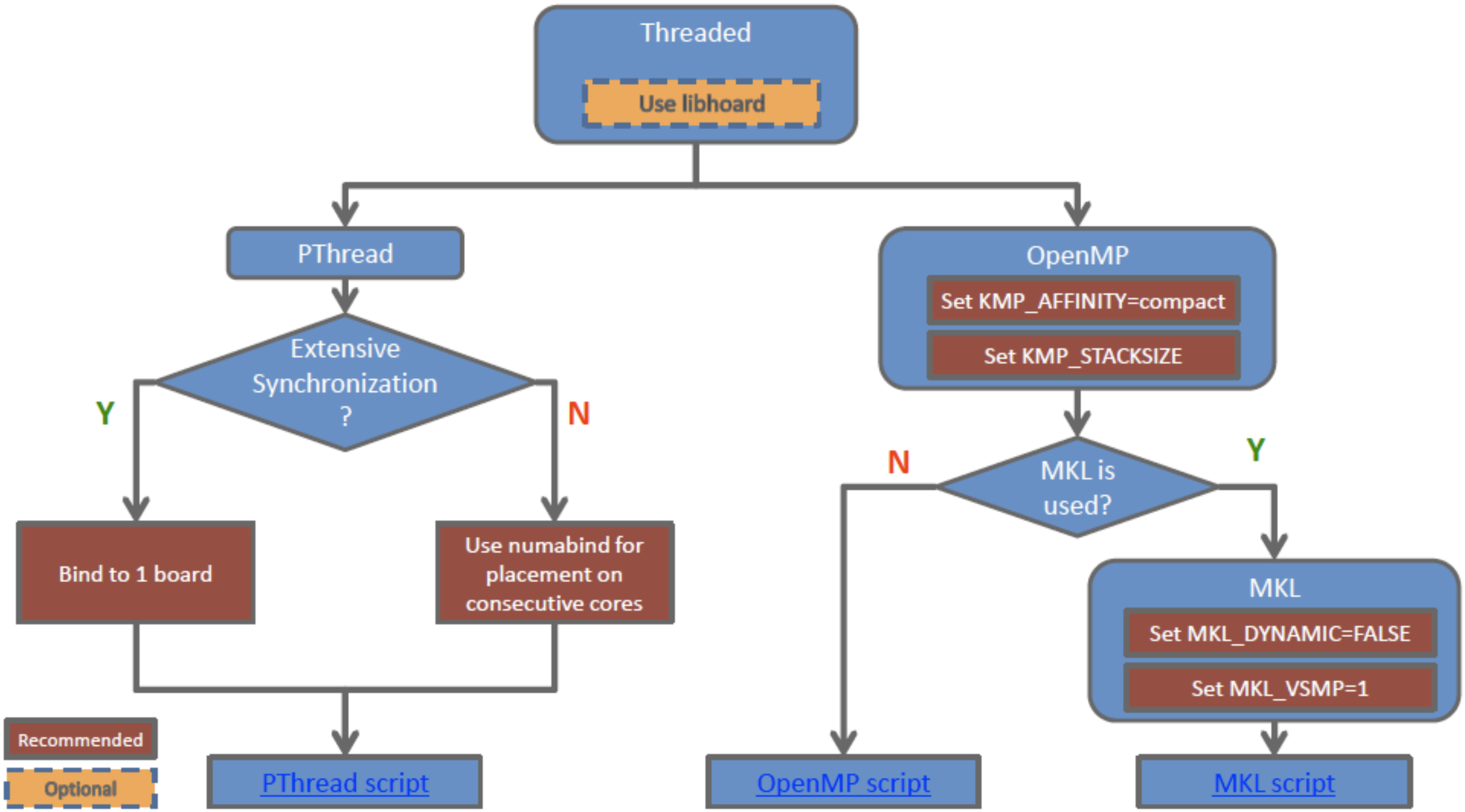


**Cores spread across
processors on same node
Optimal scatter placement**

General guidelines – MPI with vSMP



General guidelines – Threaded codes with vSMP



ScaleMP vSMP Foundation: Application Execution Guidelines

Abaqus Explicit 6.8 – Execution Guidelines for running applications in aggregated environment using ScaleMP's vSMP Foundation

Overview

Abaqus Explicit is a multi-process application that uses MPI for inter-process communication. HP-MPI has been set as the default MPI for the Abaqus Explicit application. In addition Abaqus Explicit supports MPICH2 as well by using a dmp library.

While it is possible to run Abaqus Explicit on the aggregation platform with the HP-MPI implementation, using MPICH2 tuned for vSMP Foundation may yield a performance improvement of 5-15%.

Running Abaqus-Explicit with HP-MPI

HP-MPI has a built-in mechanism for assigning MPI processes to specific CPUs. Process placement is controlled by environment variables named **MPI_BIND_MAP** and **MPIRUN_OPTIONS**. When these variables are not set, process placement will not be performed.

Environment variables – HP-MPI

If you are running with HPMPI, you should set the following environment variables prior to running Abaqus to yield the optimal performance:

```
export MPI_BIND_MAP=0,1,2,3,4,5,6,7   (For example)

export MPIRUN_OPTIONS="-cpu_bind=map_cpu,v"

export HPMP_FRAGSIZE=131072

export MPI_SHMEMCNTL=16,24000000,4000000
```

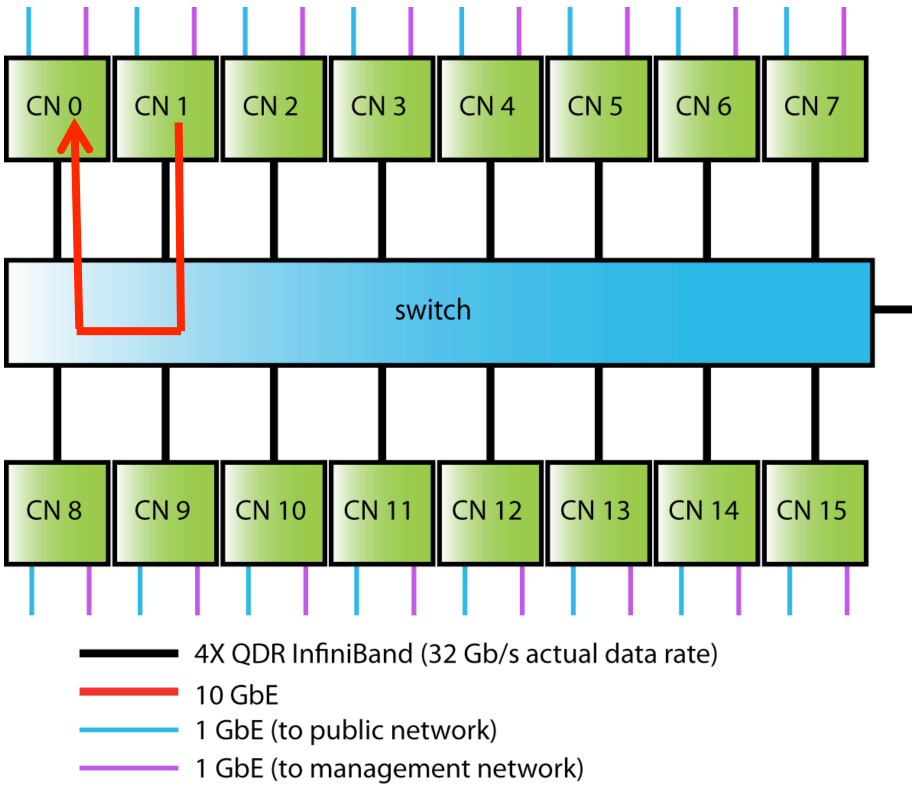
MPI_BIND_MAP specifies a list of CPUs to which MPI ranks will be bound. You should replace the list above with a list of integers, zero to #cpus-1.
For more information on HP-MPI CPU affinity settings, refer to the HP-MPI user's guide available from "http://docs.hp.com/en/B6060-96022/B6060-96022.pdf".

Running Abaqus-Explicit with MPICH2 tuned for vSMP

ScaleMP provides detailed instructions for running many applications under vSMP

- CFD
- structural mechanics
- chemistry
- MATLAB

logical shared memory – ccNUMA under the hood

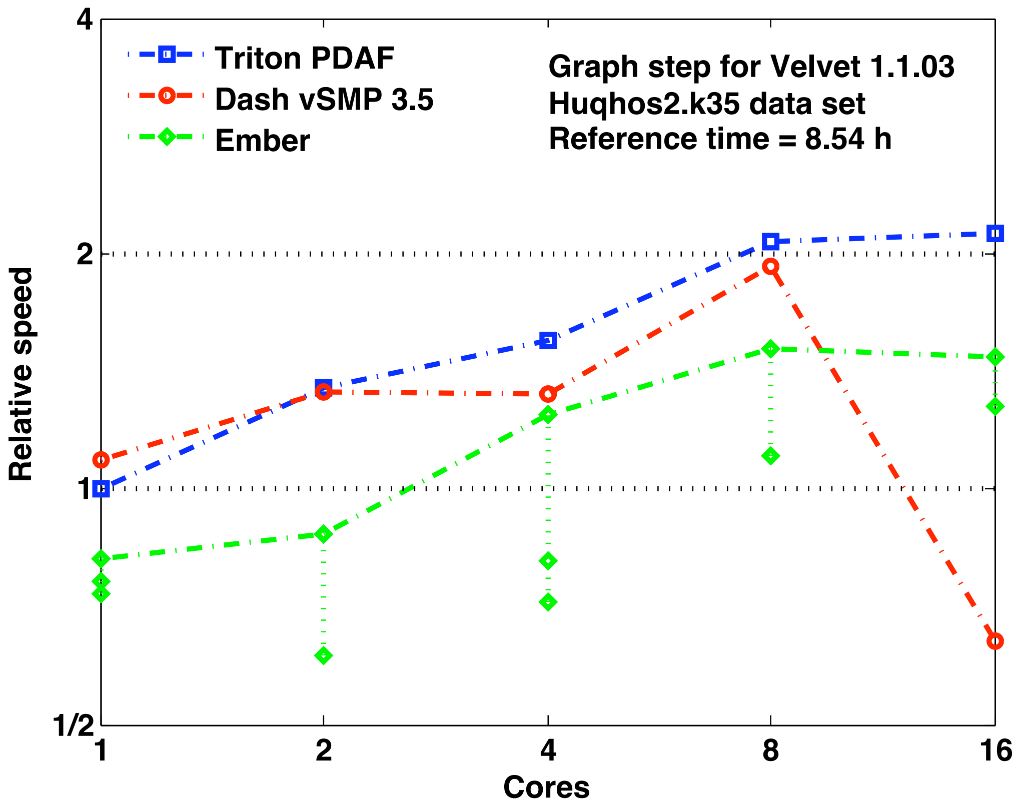


When cores executing on CN 0 require memory that resides on CN 1, page must be transferred over the network.

Usual rules for optimizing for cache still apply – take advantage of temporal and spatial data locality.

Usual ccNUMA issues – e.g. avoid false sharing

vSMP case study – Velvet (genome assembly)

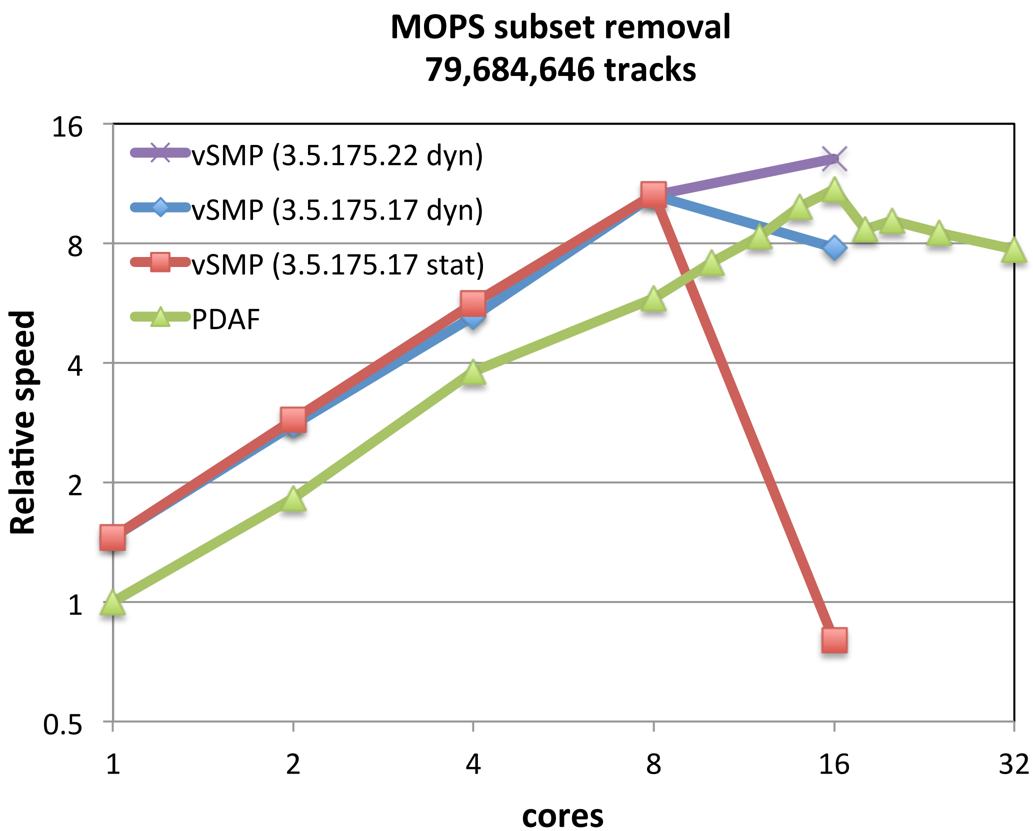


Total memory usage ~ 116 GB (3 boards)

De novo assembly of short DNA reads using the de Bruijn graph algorithm. Code parallelized using OpenMP directives.

Benchmark problem: Daphnia genome assembly from 44-bp and 75-bp reads using 35-mer

vSMP case study – MOPS (subset removal)



Total memory usage ~ 100 GB (3 boards)

Sets of detections collected using the Large Synoptic Survey Telescope are grouped into tracks representing potential asteroid orbits

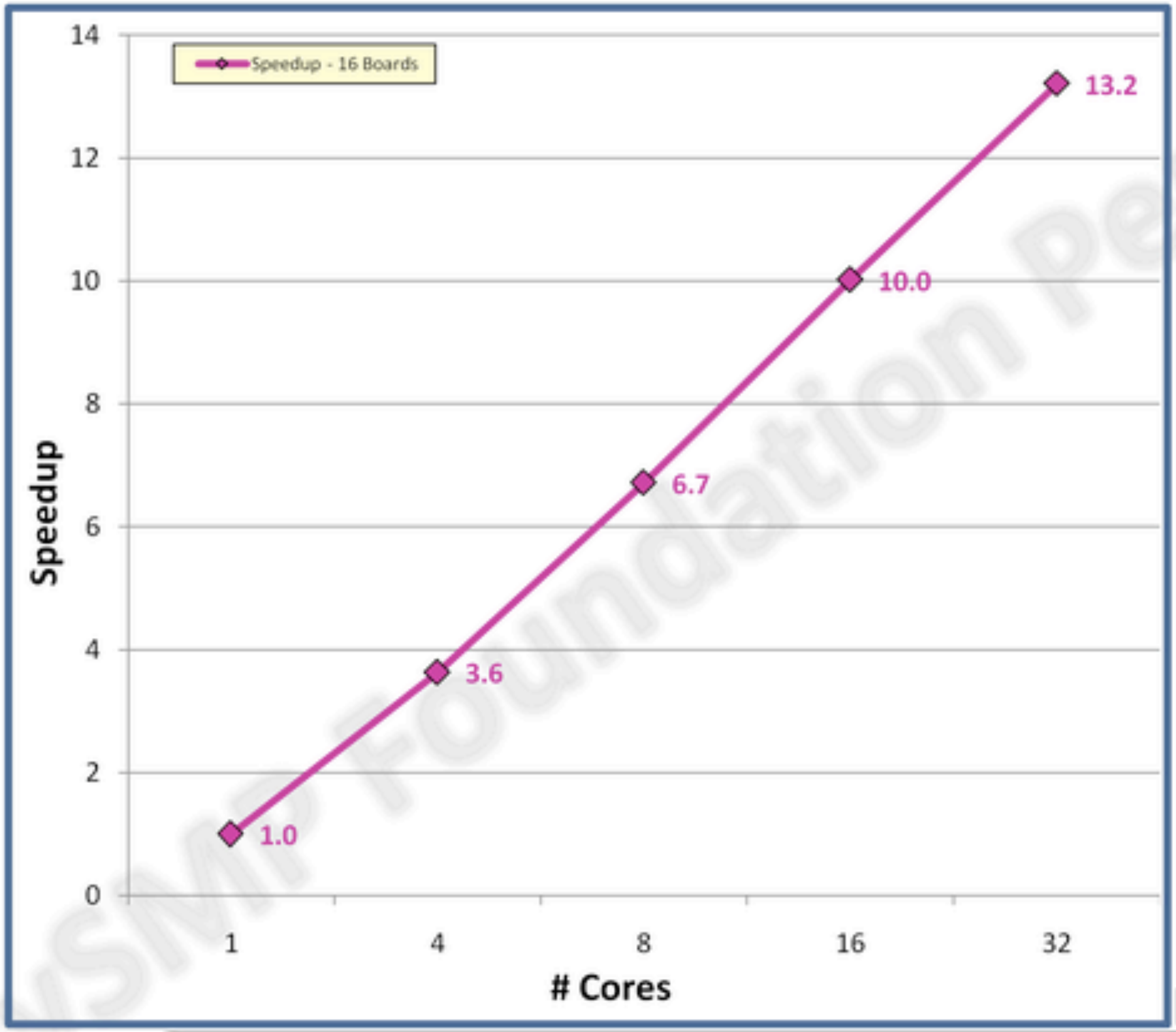
Subset removal algorithm used to identify and eliminate those tracks that are wholly contained within other tracks

7.3x speedup on 8 cores is better than that obtained on large shared memory node. Dynamic thread scheduling mitigates impact of using CPUs off board.

GAUSSIAN

Last Update:
1/4/2010

397 BENCHMARK



- vSMP Foundation scales up to 32 cores
- System configuration:
  - vSMP Foundation: 16 X Dual-socket servers (Intel Xeon X5570, 2.93 GHz, 48 GB RAM)

Threaded

Gordon Software

chemistry
adf
amber
gamess
gaussian
gromacs
lammps
namd
nwchem

visualization
idl
NCL
paraview
tecplot
visit
VTK

genomics
abyss
blast
hmmer
soapdenovo
velvet

data mining
IntelligentMiner
RapidMiner
RATTLE
Weka

libraries
ATLAS
BLACS
fftw
HDF5
Hypre
SPRNG
superLU

distributed computing
globus
Hadoop
MapReduce

compilers/languages
gcc, intel, pgi
MATLAB, Octave, R
PGAS (UPC)
DB2, PostgreSQL

* Partial list of software to be installed, open to user requests

vSMP Tools - vsmpstat

Board Basic Counters:

bbc:Bd	Time	%VMM	%Brd	%Sys	#Brd	#Sys	#TLB Flush	#PTW	%PTEm	#PTf	#4kCL	%Use
bbc:00	11120	6.5	2.7	3.9	2142548	609714	250551	10019	99.2	1049	12067568	91.1
bbc:01	11121	0.8	0.4	0.5	445861	29177	14107	486	90.3	41394	12083493	0.6
bbc:02	11122	1.3	0.8	0.5	452153	22378	10882	598	84.3	42866	12083497	0.6
bbc:03	11121	1.3	0.8	0.4	399664	17905	8500	0	0.0	44117	12091753	0.1
bbc:04	11119	0.8	0.4	0.5	392337	22196	10321	0	0.0	44461	12091732	0.1
bbc:05	11120	0.8	0.3	0.5	396604	25324	11938	0	0.0	44509	12074476	0.1
bbc:06	11119	0.9	0.5	0.4	394401	20608	9678	0	0.0	44556	12083480	0.1
bbc:07	11119	0.8	0.4	0.4	383972	21400	10057	0	0.0	42552	12083476	1.1

System Event Time:

set:Bd	%Sys	DLIN	DPT	DDMA	EVAC	MMIO	PIO	IPI	PIN	LLIN	LDMA
set:00	3.9	97.1	-	0.0	-	-	-	0.0	-	2.9	-
set:01	0.5	39.8	0.0	-	56.7	3.4	0.0	0.0	-	-	-
set:02	0.5	34.9	-	0.0	59.4	5.5	0.1	0.0	-	-	-
set:03	0.4	29.7	-	-	64.5	5.8	-	-	-	-	-
set:04	0.5	36.7	-	-	59.7	3.7	-	-	-	-	-
set:05	0.5	38.8	-	-	57.6	3.6	-	-	-	-	-
set:06	0.4	31.7	-	-	63.3	5.0	-	-	-	-	-
set:07	0.4	31.2	-	-	63.9	4.8	-	-	-	-	-

System Event Count:

sec:Bd	#Brd	DLIN	DPT	DDMA	EVAC	MMIO	PIO	IPI	PIN	LLIN	LDMA
sec:00	609714	99.3	-	0.0	-	-	-	0.0	-	0.7	-
sec:01	29177	90.4	0.0	-	5.5	4.0	0.1	0.0	-	-	-
sec:02	22378	85.3	-	0.0	7.2	7.3	0.2	0.0	-	-	-
sec:03	17905	83.9	-	-	9.1	7.0	-	-	-	-	-
sec:04	22196	87.4	-	-	7.3	5.3	-	-	-	-	-
sec:05	25324	88.9	-	-	6.4	4.7	-	-	-	-	-
sec:06	20608	86.0	-	-	7.9	6.1	-	-	-	-	-
sec:07	21400	86.5	-	-	7.6	5.9	-	-	-	-	-

board counters

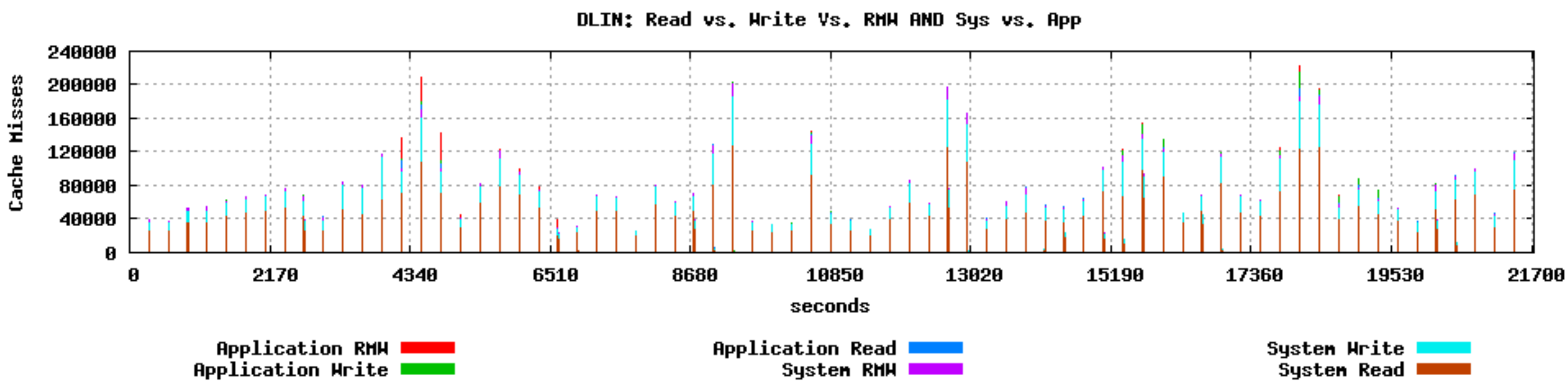
board event counts

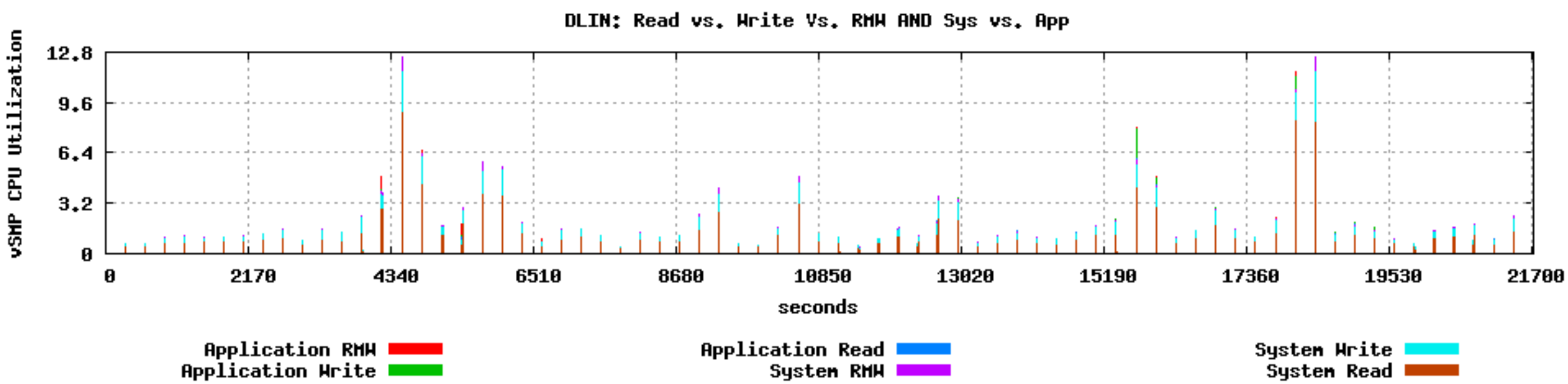
board event timing

system event counts

system event timers

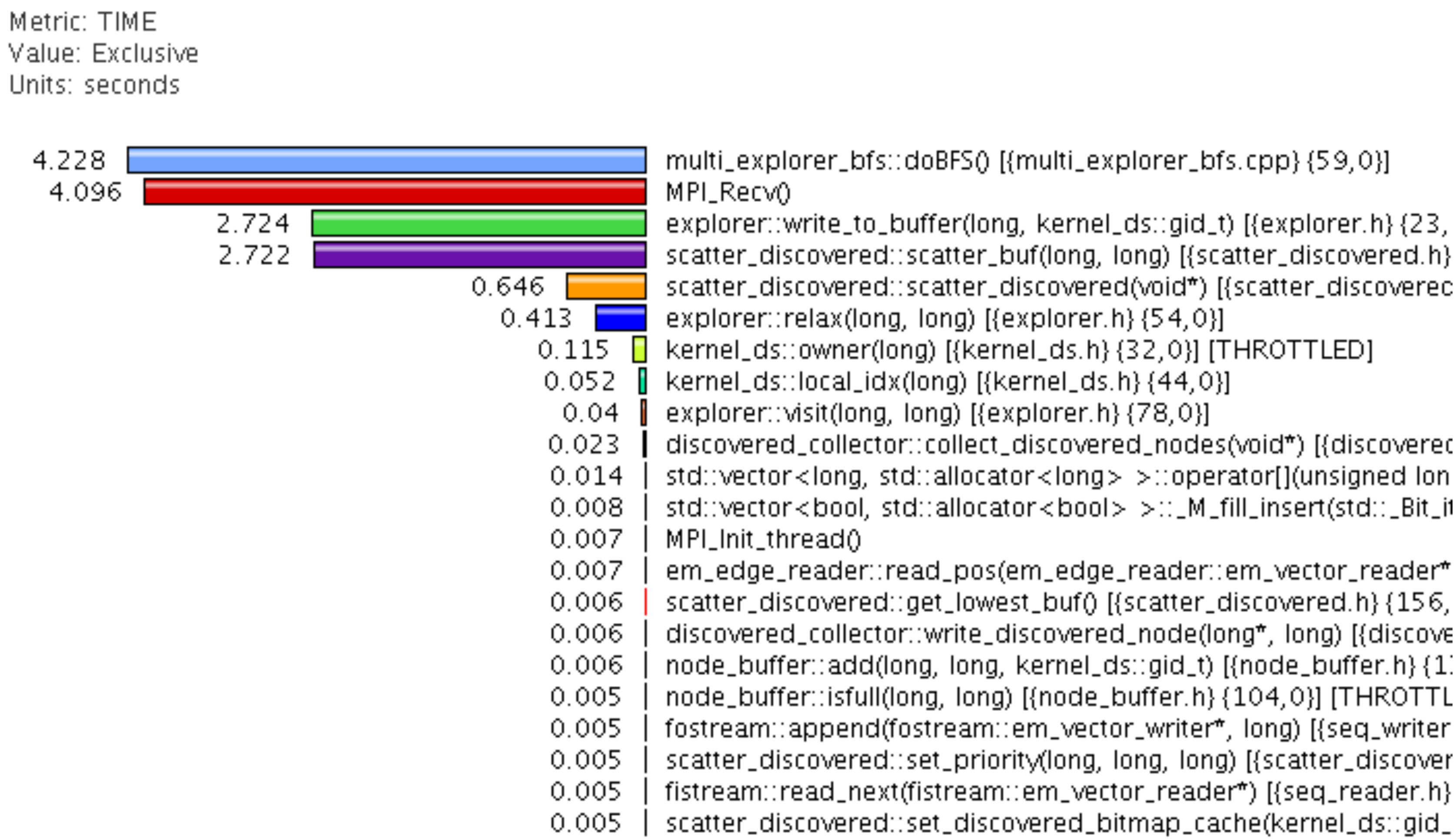
vSMP tools – vsmpprof / logpar





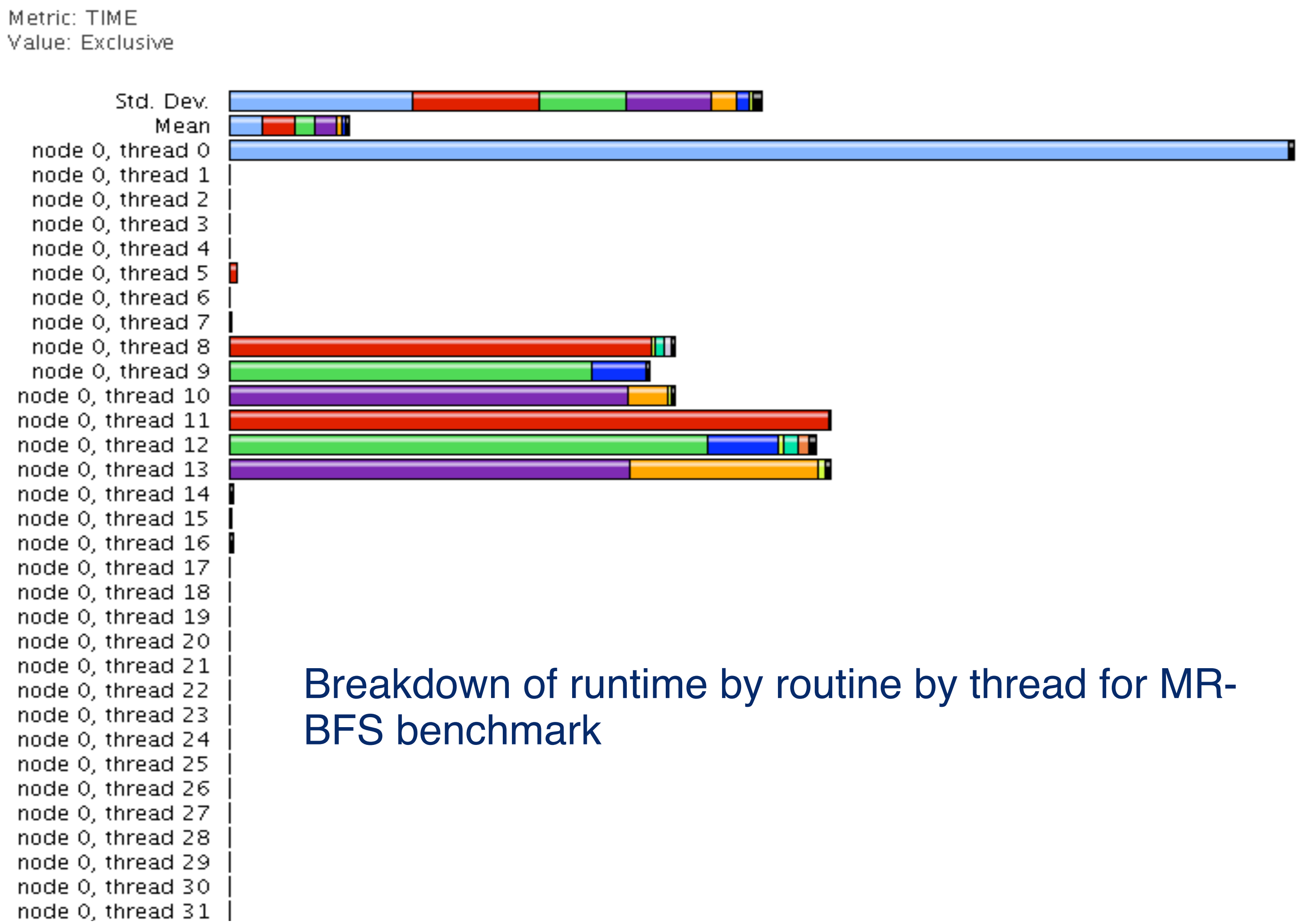
Profiling results obtained for Velvet run on Dash vSMP node

General purpose tools - TAU



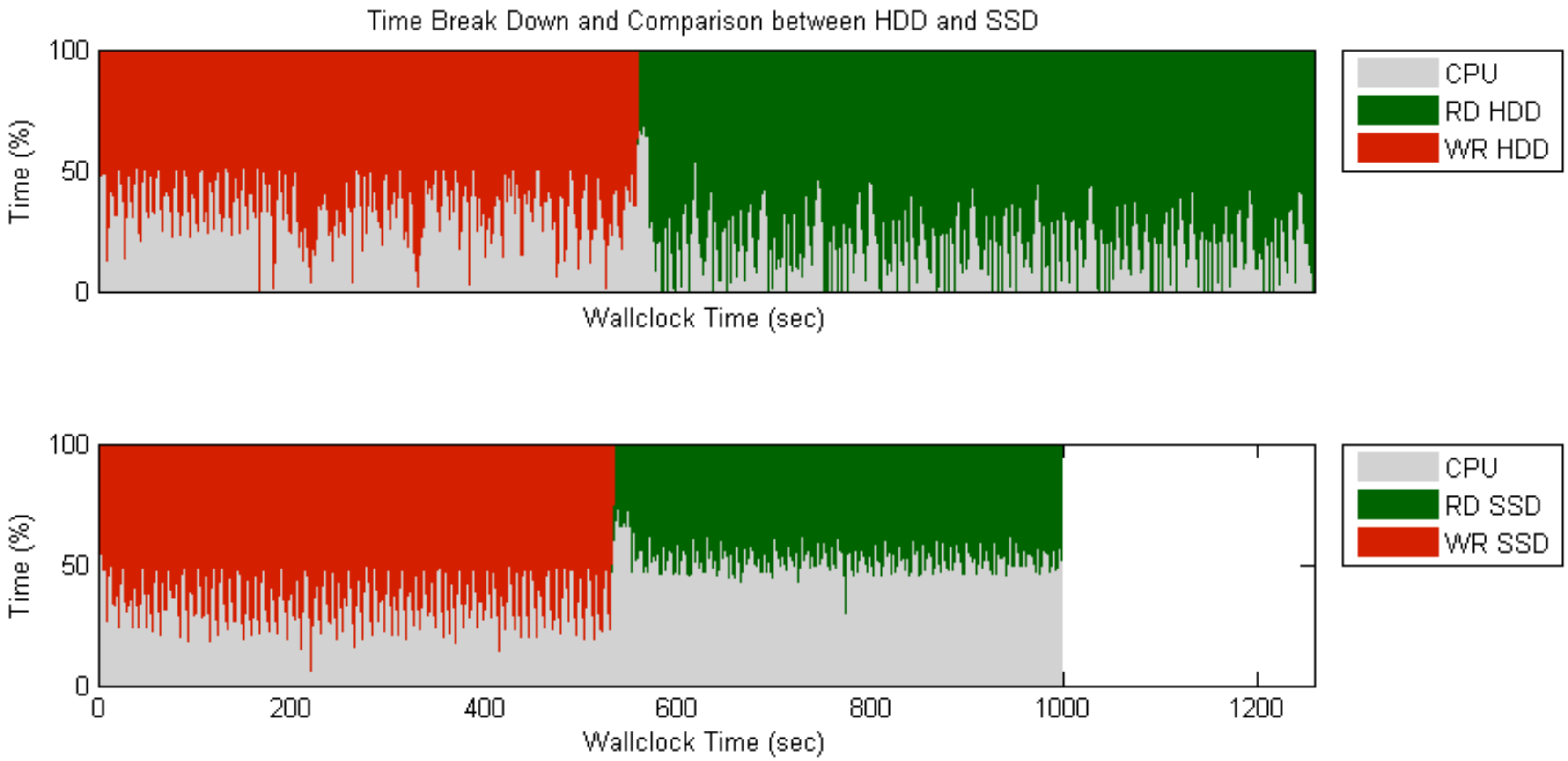
Breakdown of runtime by routine for MR-BFS benchmark

General purpose tools - TAU

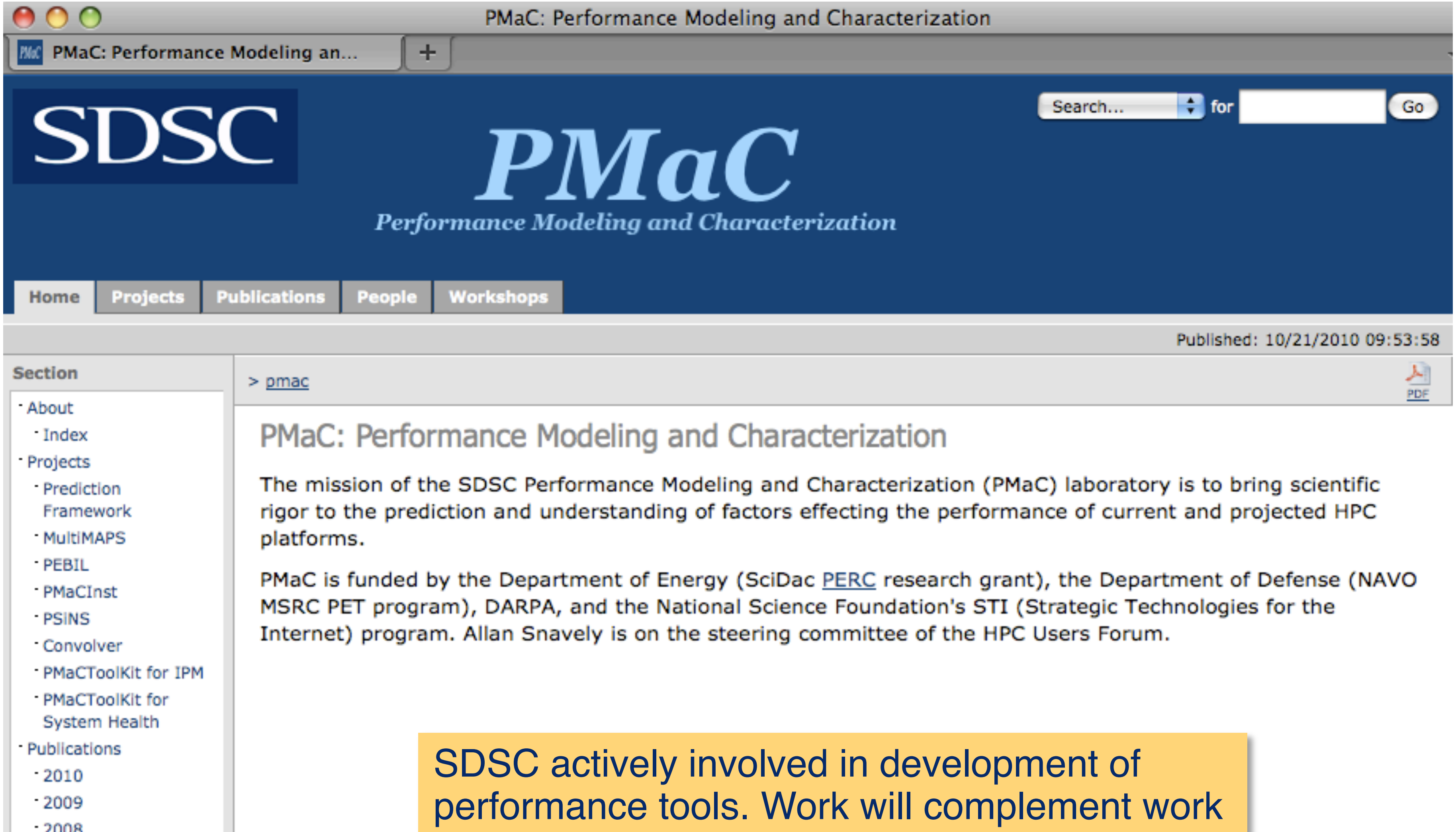


Breakdown of runtime by routine by thread for MR-BFS benchmark

General purpose tools - PEBIL



Division of time between computation and I/O for acoustic imaging application. Comparison between flash and hard disks



SDSC actively involved in development of performance tools. Work will complement work done to deploy applications on Gordon

Cluster management and Job scheduling

Cluster management and job scheduling will be handled using the Simple Linux Utility for Resource Management (SLURM)

- Open source, highly scalable
- Deployed on many of the world's largest systems, including Tianhe-1A and Tera-100
- Advanced reservations
- Backfill scheduling
- Topology aware

Job submission

SLURM batch script syntax is different from Torque/PBS. A translator does exist, but we will strongly encourage users to use the new syntax

Access to different types of resources (vSMP, I/O, and regular compute nodes) will be determined from queue name

Scheduler will handle optimal placement of jobs

- N < # cores/node: all cores belong to single node
- N <= 16 nodes: all nodes connected to same switch
- N > 16 nodes: neighboring switches in 3D torus

Obtaining allocations on Gordon

Gordon will be allocated through the same process as other TeraGrid (XSEDE) resources (reviewed by TRAC)

But… some things will be different

- Must make a strong case for using Gordon, justifying use of flash memory and/or vSMP nodes. Wanting access to Sandy Bridge processors is not sufficient
- Can request compute nodes and/or I/O nodes
- The allocations committee will be authorized to grant dedicated access to I/O nodes

https://www.teragrid.org/web/user-support/allocations

Essential - Make the case for Gordon

- **vSMP**
- Threaded codes requiring large shared memory (> 64 GB)
- MPI applications with limited scalability, where each process has large memory footprint

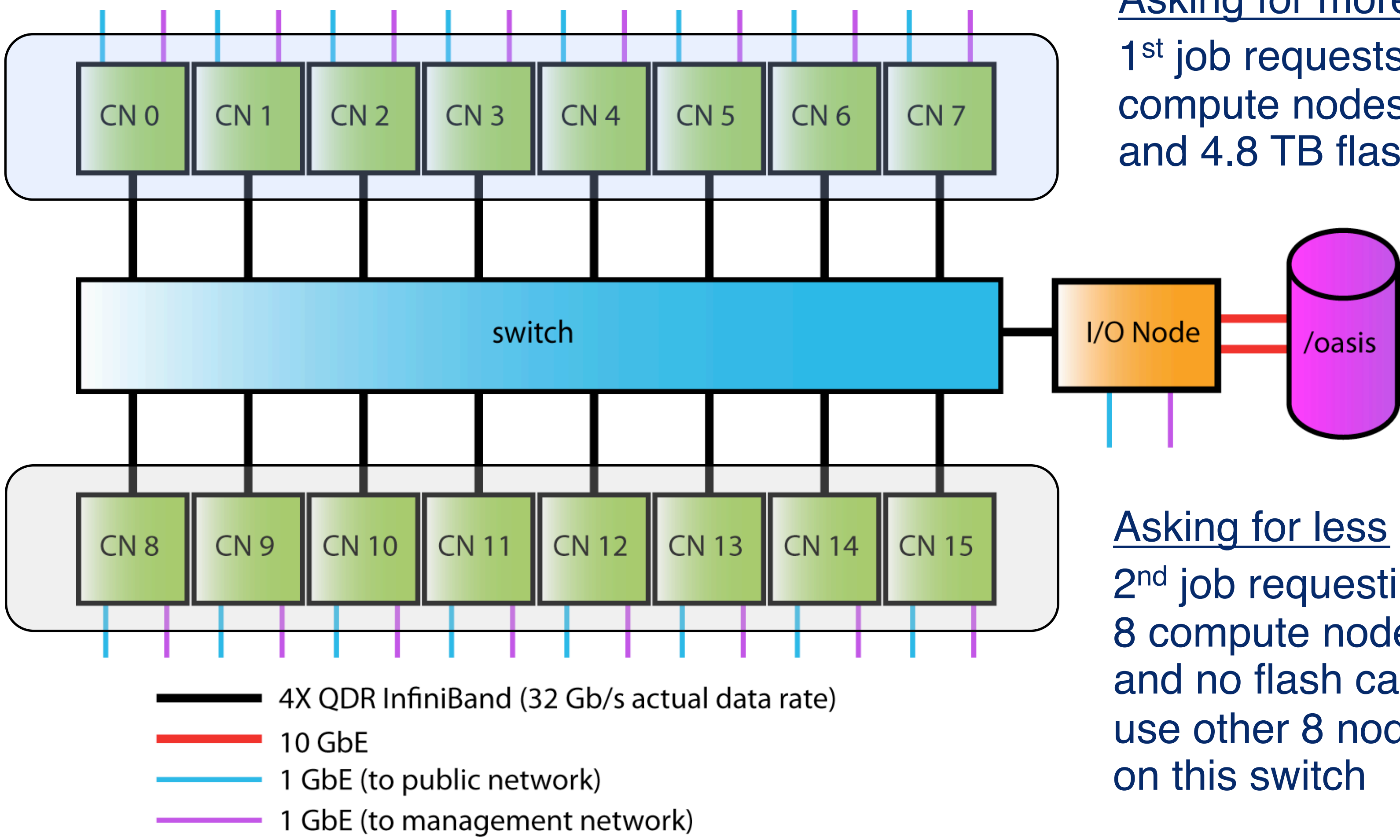
- **Flash**
- Apps that will run much faster when data set resides in flash (keep in mind time to populate flash)
- Flash used as level in memory hierarchy
- Scratch files written to flash

- **MPI apps with limited scalability, but potential for hybrid parallelization**

Gordon compute nodes allocations and usage (proposed)

- Awards made in the usual way (1 core hour = 1 SU)
- vSMP nodes
  - Jobs should request cores in proportion to amount of memory required
- Flash
  - Default: flash made available in proportion to nodes requested (for both vSMP and non-vSMP)
  - Jobs can request more flash memory
  - Jobs can request less flash memory

Advantage of specifying flash requirements



<u>Asking for more</u>

1st job requests 8 compute nodes and 4.8 TB flash

<u>Asking for less</u>

2nd job requesting 8 compute nodes and no flash can use other 8 nodes on this switch

Gordon dedicated I/O nodes allocations and usage (proposed)

- Can request long-term dedicated use of one or (in exceptional cases) two I/O nodes
- Four dedicated compute nodes will be awarded for each compute node unless strong justification is made for more
- Usage scenarios
  - Hosting/analysis of community data sets
  - Very large data sets with “hot” results
  - Science Gateways: www.teragrid.org/web/science-gateways
  - Other special cases that we haven’t even thought of, but maybe you have

How will Gordon be deployed?

- Fraction of machine deployed as vSMP nodes
- Size of vSMP nodes
- Number of I/O nodes allocated as dedicated
- Fraction of machine available for interactive jobs
- Fraction of I/O nodes used for visualization
- Size and length of queues

Answers to all of these questions depends heavily on the mix of allocations requests approved by committee, demand by user, and scheduling decisions to balance needs of users

Advanced User Support

Advanced Support for TeraGrid Applications (ASTA)

Home > User Support > ASTA

Advanced Support for TeraGrid Applications (ASTA) provides collaboration between Advanced User Support (AUS) staff and users of TeraGrid resources. The objective of the program is to enhance the effectiveness and productivity of scientists and engineers. As a part of the ASTA program, guided by the allocation process, one or multiple AUS staff will join the Principle Investigator's team to collaborate for up to a year, working with users' applications.

On this page

- How to Apply
- ASTA Selection Process

Related Links

- ASTA Project List
- Allocations Information

Gordon has a number of features that are totally new to most TeraGrid users. We strongly suggest that you request ASTA support as part of your allocation if you require special assistance in adapting your application to make use of Gordon.

https://www.teragrid.org/web/user-support/asta
https://www.xsede.org/auss

Education, Outreach and Training

TeraGrid 2010

- Tutorial and Hands-on Demo: *Using vSMP and Flash Technologies for Data Intensive Applications*. Presented by Mahidhar Tatineni and Jerry Greenberg, SDSC User Services
- Invited Talk: *Accelerating Data Intensive Science with Gordon and Dash*. Michael Norman and Allan Snavely (Norman presenting)
- Technical Paper: *DASH-IO: An Empirical Study of Flash-Based IO for HPC*. Jiahua He, Jeffrey Bennett, Allan Snavely (He presenting)
- Birds of a Feather: *New Compute Systems in the TeraGrid Pipeline*. Richard Moore, Chair. Michael Norman presenting on the Gordon system.

Grand Challenges in Data Intensive Discovery Conference (GCDID) – October 26-28, 2010

Education, Outreach and Training

<u>Grand Challenges in Data Intensive Discovery Conference (GCDID) – October 26-28, 2010 (~90 attendees)</u>

- ***Visual Arts*** - Lev Manovich, UC San Diego
- ***Needs and Opportunities in Observational Astronomy*** - Alex Szalay, Johns Hopkins University
- ***Transient Sky Surveys*** - Dovi Poznanski, Lawrence Berkeley National Laboratory
- ***Large Data-Intensive Graph Problems*** – John Gilbert, UC Santa Barbara
- ***Algorithms for Massive Data Sets*** – Michael Mahoney, Stanford University
- ***Needs and Opportunities in Seismic Modeling and Earthquake Preparedness*** - Tom Jordan, University of Southern California
- ***Economics and Econometrics*** - James Hamilton, UC San Diego

plus many other topics

<u>http://www.sdsc.edu/Events/gcdid2010/docs/GCDID_Conference_Program.pdf</u>

Education, Outreach and Training

Supercomputing 2010

- *Understanding the Impact of Emerging Non-Volatile Memories on High-Performance, IO-Intensive Computing*, Adrian M. Caulfield, Joel Coburn, Todor Mollov, Arup De, Ameen Akel, Jiahua He, Arun Jagatheesan, Rajesh K. Gupta, Allan Snavely, and Steven Swanson, Supercomputing, 2010. (*Nominated for best technical paper and best student paper*).

- *DASH: a Recipe for a Flash-based Data Intensive Supercomputer,* Jiahua He, Arun Jagatheesan, Sandeep Gupta, Jeffrey Bennett, Allan Snavely. Supercomputing, 2010.

- Live demo 4x4x2 torus (Appro, Mellanox, SDSC)

Education, Outreach and Training

Biennial Richard Tapia Celebration of Diversity in Computing (San Francisco, CA)

vSMP Workshop (May 10-11, 2011)

Early-Users Track 2D Workshop at the Open Grid Forum (July 15 - 17, 2011)

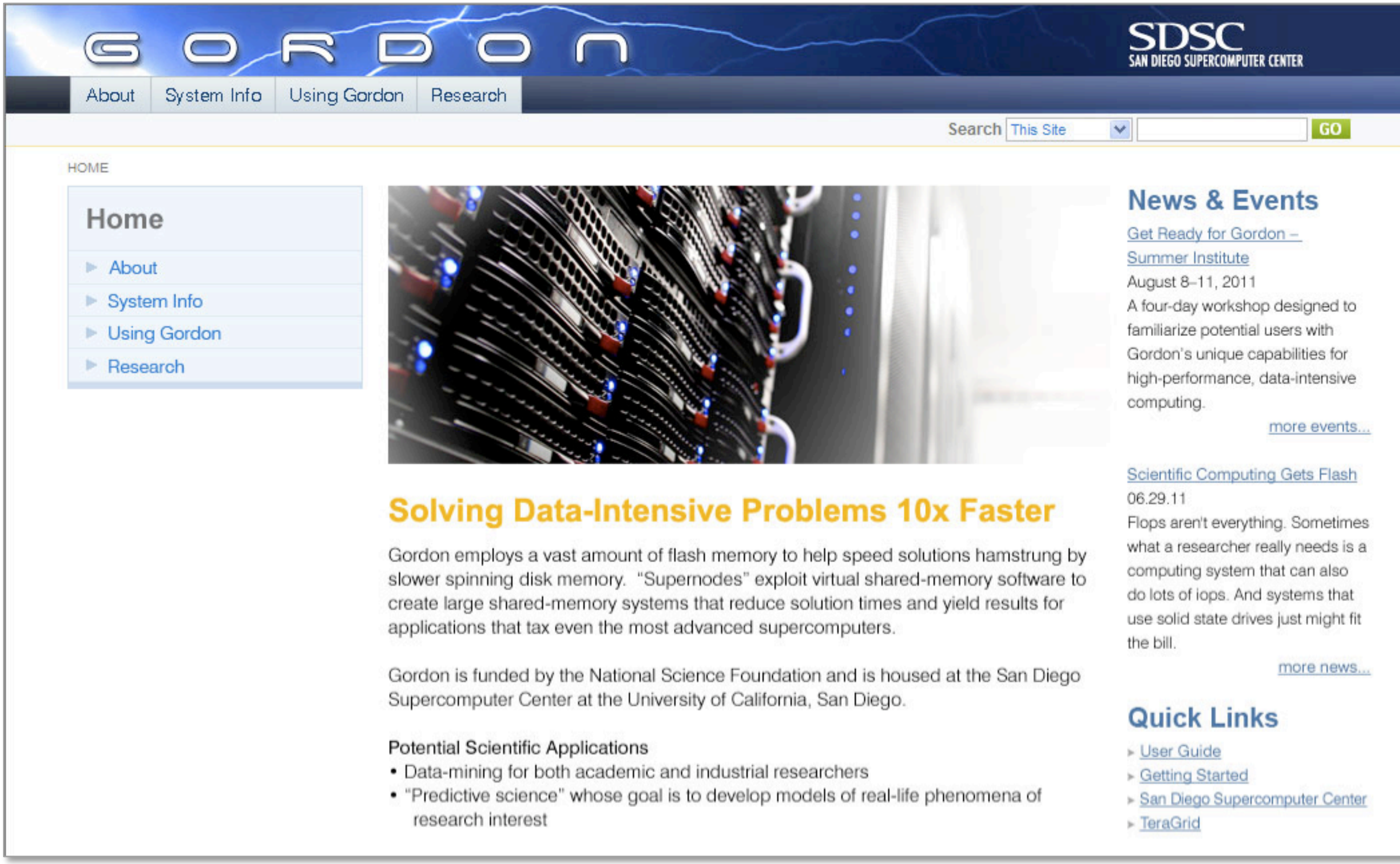
TeraGrid 2011 (July 17-22, 2011)

- **Tutorial:** *An Introduction to the TG Track 2D Systems: FutureGrid, Gordon, & Keeneland*. Tutorial Abstract:
- **Paper:** *Subset Removal on Massive Data with Dash* (Myers, Sinkovits, Tatineni). Paper abstract:

Get Ready for Gordon: Summer Institute (GSI) (August 8-11, 2011)

KDD11 - Data Intensive Analysis on the Gordon High Performance Data and Compute System (August 21-24, 2011)

Coming soon … one stop site for Gordon http://gordon.sdsc.edu



Gordon Team

SDSC

Mike Norman – PI
Allan Snavely – co-PI
Shawn Strande – Project Manager
Bob Sinkovits – Applications Lead
Mahidhar Tatineni – User support / applications
Jerry Greenberg – Applications (chem, MATLAB)
Pietro Cicotti – Applications & benchmarking
Wayne Pfeiffer – Applications (genomics)
Jeffrey Bennett – Storage Engineer
Eva Hocks – Systems Administration
William Young - Systems
Chaitan Baru – Database applications
Kenneth Yoshimoto – Scheduling/SLURM
Susan Rathbun – Project Coordinator
Diane Baxter - EOT
Jim Ballew – acceptance testing and design
Amit Majumdar – ASTA
Nancy Wilkins – Science Portals

UCSD

Steve Swanson
Adrian Caulfield
Jiahua He (now at Amazon)
Meenakshi Bhaskaran

ScaleMP

Nir Paikowsky
(and many others)

Appro

Steve Lyness
Greg Faussette
Adrian Wu
Roland Wong