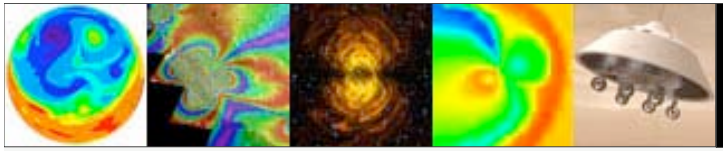
Computational Modeling Technology Trends

Tom Clune

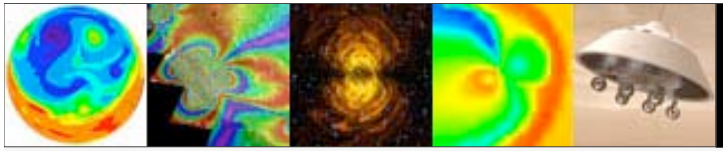
NASA SCIENCE MISSION DIRECTORATE

COMPUTATIONAL MODELING CAPABILITIES WORKSHOP, JULY 29-30, 2008

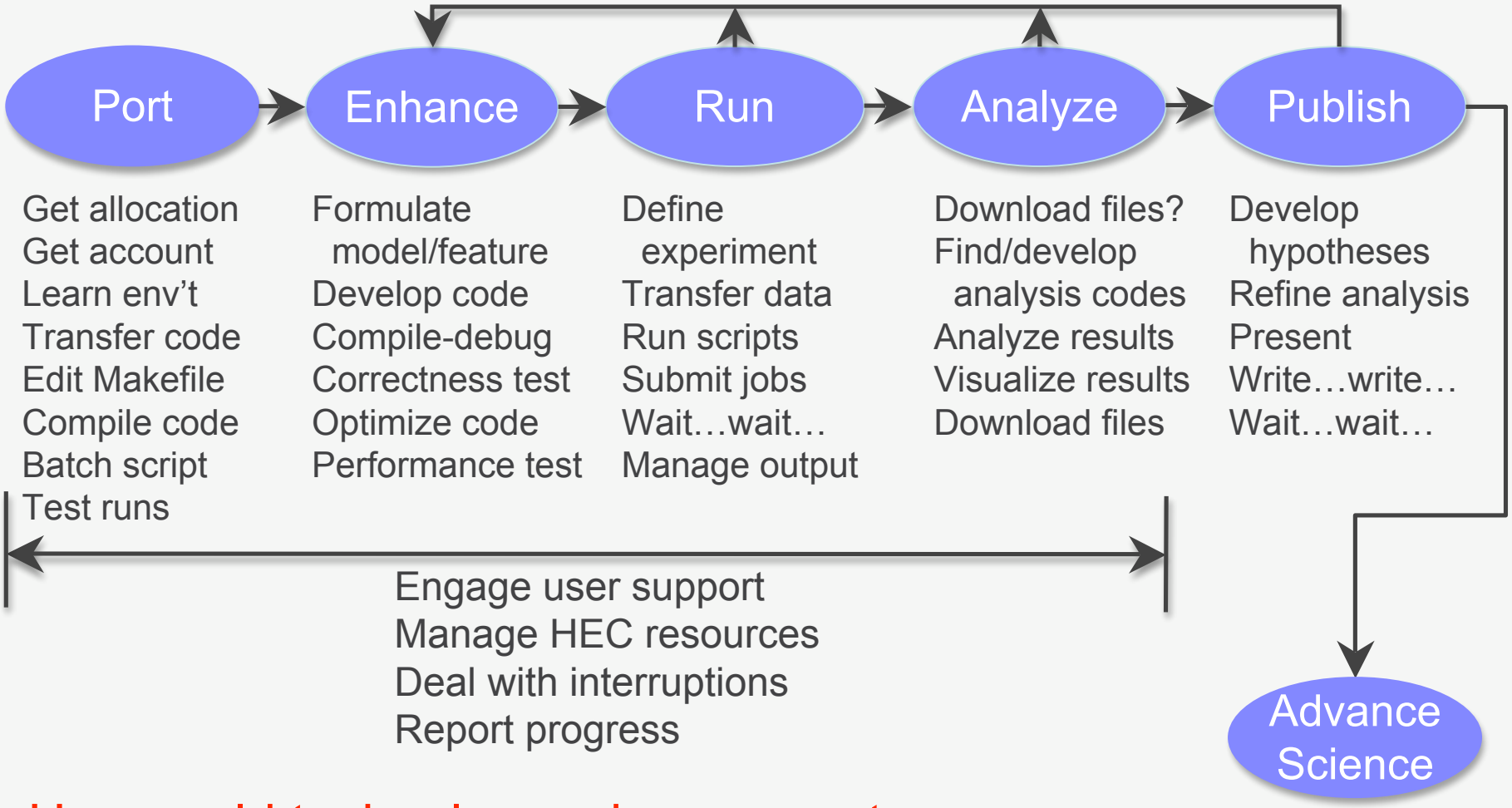


Outline

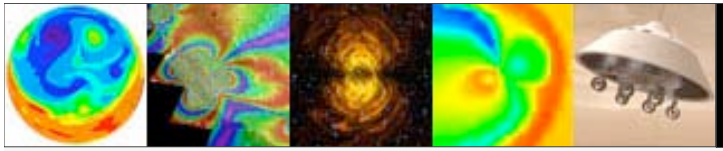
- SMD's Current Computational Modeling Environment
- Technology Trends
  - Applications
  - Computational Architectures
  - Storage and I/O
  - Software Development Tools
  - User Interface/Environment
  - User Services



Typical SMD HEC User Workflow



How could technology advancements accelerate your HEC workflow?



Trends in Computational Architecture Components

General Trends

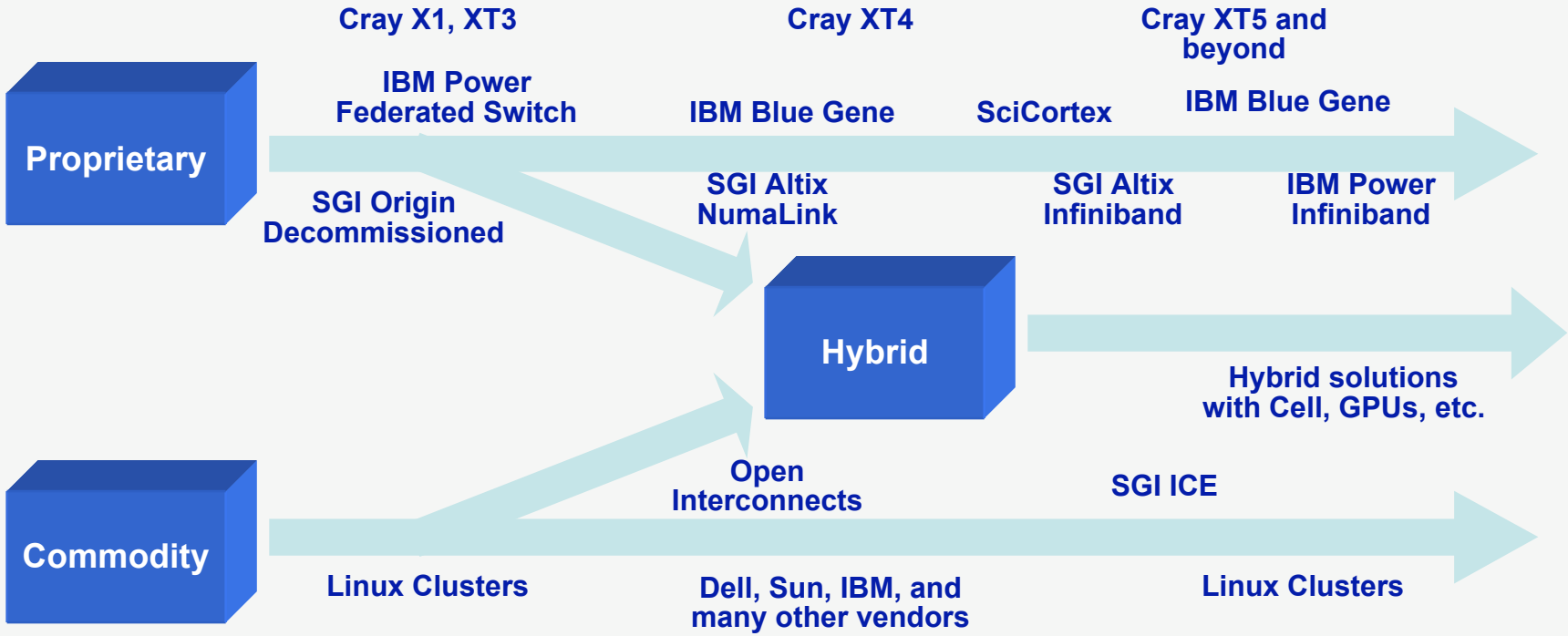
- Multi-core is here to stay
- Very few clock speed bumps for the future
- Multi-processing element paradigm necessary
- Continued imbalance of processing power and data bandwidth
- Commodity interconnects overtaking proprietary interconnects
- Low power
- Virtualization

	Production: NOW	Cutting Edge	Bleeding Edge	Research
Specialized Units	Accelerator Cards, e.g. Clearspeed; Cell	GPU 128-core NVidia; GPU >128-core	HyperTransport Enabled; Virtualization	FPGA
Processors	32-bit Vector Scalar; 64-bit Vector (Cray) and Scalar (Power, Xeon, Itanium, Opteron, Sparc); Single-core; Dual-core; HyperTransport	Quad-core; Intel CSI	Multi-core SMP on a chip; TF Chips	Beyond CMOS
Interconnect	Proprietary • Cray • IBM • SGI (NUMA); Standards Based; 3 µsec	• Quadrics • Myrinet • Infiniband • 10 GbE • Federation; 1 µsec	Integrated Networks; Proprietary • Cray •SciCortex	Optical; <1 µsec
File Systems	100 MB/s; Parallel FS • PVFS • GPFS • GFS; SAN FS • CXFS • SAM • StorNext; Object Oriented FS • Lustre • Panasas	4 Gb FC; IB; NFS Enhancements • Parallel NFS • NFS v4 • NFS RDMA	8 Gb FC; 100 GB/s; Network Attached • NetApp • Isilon	TB/s

Increasing IT Capability Over Time

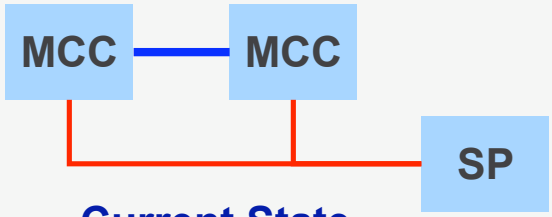
Computational Architectures

- Representative Technologies: cluster, single system image, multi-core, GPU, FPGA, cell, etc.; memory, memory bandwidth; Interconnect bandwidth and latency
  - Adoption of commodity components (processors, interconnect, memory, etc.)
  - Trend toward hybrid computing systems for computational systems
- Leading Question: What computing system characteristics are required by (or best for) my application?



Chip Architectures

- Multi-core is here to stay; soon to be many-core and many type of cores in a socket
- Push to embed specialized units on the same mother board and eventually the same chip as generalized compute units
- Power envelopes per socket are staying constant; regulation of power at the chip level
- Leading Question: What chip characteristics are required by (or best for) my application?



Current State to Near Term

On board many core chips connected via a high speed interconnect like Hyper-transport. Specialized units are connected through standardized I/O like PCI.

MCC = Many core chip

SP = Specialized Unit (cell, GPU, etc.)

Mid Term

On board many core chips connected via a high speed interconnect like Hyper-transport to not only other many core chips but also specialized units integrated into the board.

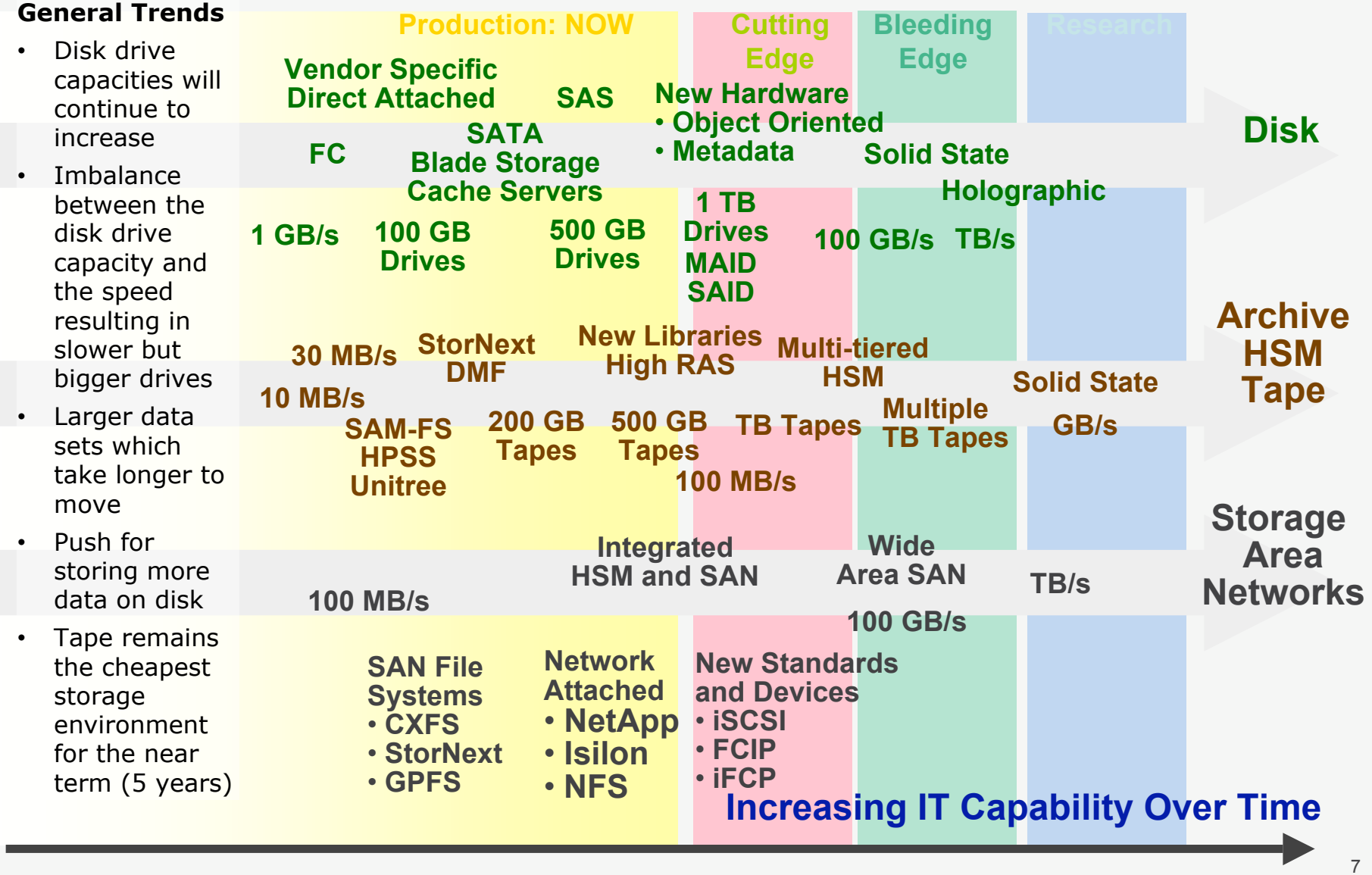
Long Term

Finally, specialized units will be integrated onto the many core chip itself.

Storage and Archiving Technology Trends

General Trends

- Disk drive capacities will continue to increase
- Imbalance between the disk drive capacity and the speed resulting in slower but bigger drives
- Larger data sets which take longer to move
- Push for storing more data on disk
- Tape remains the cheapest storage environment for the near term (5 years)



Data and I/O: Hardware

- Representative Technologies: on-line storage (e.g., spinning disks; volume, stagging, lifetime), archive I/O, archival hardware (e.g., tape silos)
- On-line Storage
  - Disk speeds have not kept up with the throughput requirements for HPC
  - The only way to get the throughput needed is to have lots of disks
  - New disk technologies and capabilities have emerged to provide a much better price performance for near line disk
    - Beyond RAID: MAID, SAID
    - Beyond FC: SATA, SAS
  - De-duplication and compression technologies may allow for better disk usage, but may come at a performance cost
- Archive
  - Given the volume of data handled by the HPC centers, archive technologies will remain
    - Tape will remain the most cost effective long term solution
  - Can de-duplication help within an archive system?
- Leading Question: What hardware storage system characteristics are required by (or best for) my application?

Data and I/O: Software

- Representative Technologies: file systems (e.g., global), system I/O (volume, frequency)
- File Systems
  - Global parallel file systems: IBM GPFS, Sun Lustre, PVFS, and more
  - Perhaps one of the most difficult sub-system to support for an HPC center
    - File systems tend to cause more problems on an HPC system than any other issue
    - User applications exercise the file systems in ways that the file system developers did not imagine
  - Interoperability between file systems remains lacking
    - Users and centers are forced to employ slower technologies, like NFS, scp, bbftp, etc.
    - NFSv4 shows some promise
- Data Management
  - There is a need to capture metadata to catalogue, search, query, find, and read data sets for long periods of time
  - This goes beyond knowing where the data files are stored on disk or tape to supporting the dataflow from job setup through the entire workflow process
- Leading Question: What software storage system characteristics are required by (or best for) my application?

Program Development

- Architectural trends:
  - Many-core chips leading to larger shared memory nodes
  - Deeper memory hierarchies
  - Complex interconnection networks
  - Hyper-threading (fast thread switching) to hide memory latency
  - Accelerators/Coprocessors (GPGPU, Cell, FPGA)

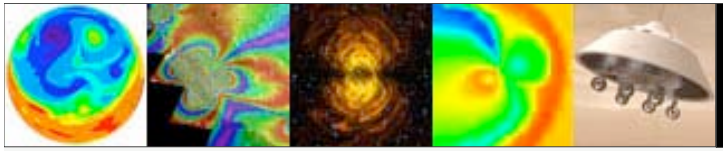
- Challenges for models/languages/tools:
  - A high level of abstraction for the user to express the application (as close to the domain as possible)
  - Enough information for the compiler/runtime system to effectively exploit the target architectures
  - Trade-off between portability/maintainability versus performance
  - Feedback on performance implications of code segments

Program Development: Models/Languages

- Current:
  - *MPI:*
    - standard approach for distributed memory systems;
    - provides no support for other architectural issues
  - *OpenMP:*
    - focus on shared memory and loop parallelism; recent support for task parallelism;
    - does not scale in most cases
  - *Hybrid (MPI + OpenMP):*
    - works well in some cases
    - adds complexity to the program
  - *Higher level approaches – Matlab/StarP; optimized libraries – MKL, LAPACK*
    - applicable to codes if bulk of computation can utilize optimized third party code;
    - can provide portability
- Emerging Technologies:
  - *Partitioned Global Array Space (PGAS) - UPC, CAF, Titanium*:
    - for distributed memory system -> explicit data "movement" using a global index space;
    - solves one issue, no support for other challenges
  - *Specialized - CUDA, Brook, RapidMind … :*
    - some are non-portable due to focus on specialized hardware,
    - others are not proven for large applications
  - *High Productivity Computing Systems (HPCS) - X10, Chapel, Fortress:*
    - focus on productivity and programming-in-the-large;
    - relying on compiler technology to enhance performance

Program Development: Tools

- Computer-aided parallelization:
  - *CAPO:* OpenMP
  - *ParaWise:* domain decomposition/MPI
- Debugging Tools:
  - *TotalView, DDT:* general purpose, MPI, OpenMP, many platforms
  - *ThreadChecker:* race detection on Intel platforms
  - *Valgrind, Electric Fence, Purify:* memory debugging
- Performance Tools:
  - *Tau, OpenSpeedShop, Vampir, Paraver:* general purpose
  - *Vtune:* Intel platform
  - *Perfsuite:* from NCSA, Intel only
  - *MPInside:* SGI only
  - *PAPI:* library for access to hardware counters



User Interface/Environment (1)

- Standard user interface to HEC resources: command line
  - account/resource/file management; help requests
  - develop/debug your own codes using C or Fortran
  - run jobs using arcane batch and script files
  - wait for job(s) to complete; if they fail, manually restart
  - manage and post-process output files
  - copy results to local computer to view or print
  - repeat…
- What would be a more productive interface to HEC systems and services than SSH and a command line?
  - web interface to resources, queues, and help system
  - prototype codes in higher level or mathematical language
  - application frameworks (e.g., ESMF) to incorporate algorithms, link models, compare/validate results, etc.
  - Ensemble and workflow management tools; intelligent environment
  - Real-time (every timestep) rendering and streaming of simulations

User Interface/Environment (2)

My Supercomputing Home Page

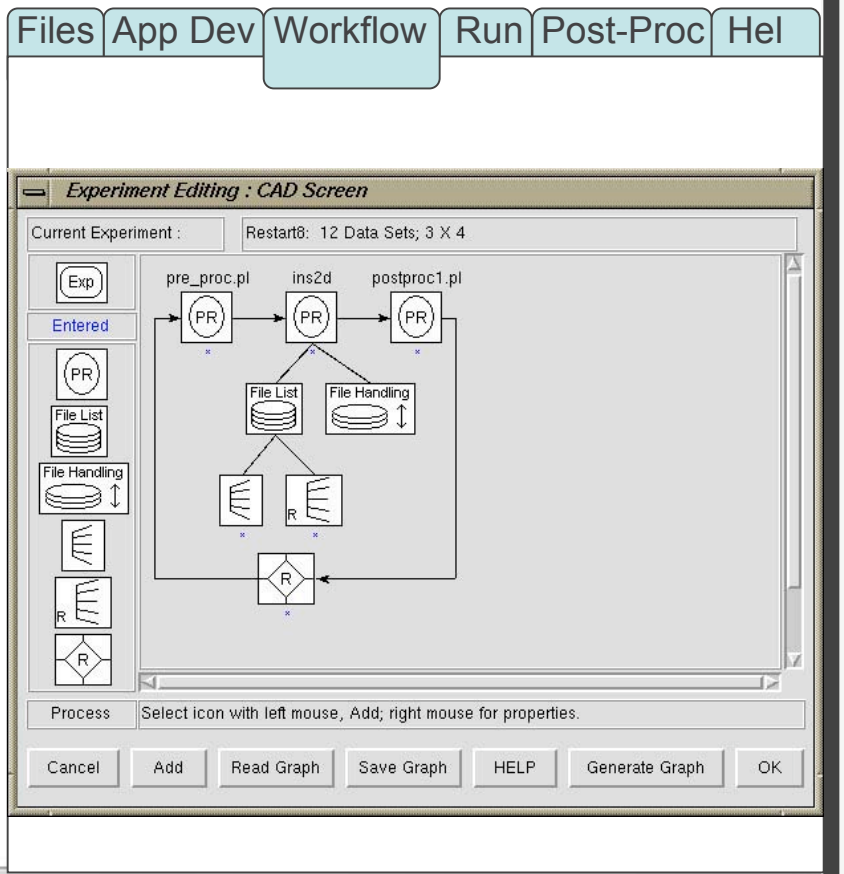
http://myhec.nasa.gov/home.html

Search

Account			
UID: bbiegel	GID: 22978	Expires: 3/31/2009	
Resource	**Allocation**	**Remaining**	**Est. Use Date**
CPU-Hours	1,000,000	49.30%	12/13/2008
Disk (GB)	10,000	0.93%	8/19/2008
Tape (TB)	100	48.60%	3/15/2009

Queues			
Domain:	Science	Status:	Nominal
Queue	**CPUs**	**Available**	**Avg. Wait (Hrs)**
high_priority	2,000	10.20%	0.36
normal	4,000	0.98%	2.77
low_priority	0	N/A	19.05

Workflows and Job Sets		
Status	**Number**	**Est. Completion (Hrs)**
In Prep.	2	N/A
Queued	6	69.75
Running	3	4.98
Completed	139	N/A




COMPUTATIONAL MODELING
TECHNOLOGY TRENDS
NASA

Trends in User Services

- Support Funding reduced by scientists and technologists
  - Increasing rate of consolidation and remote support
  - Increasing Use of unattended support
    - Online documentation: how-to, faq, wiki
    - Good search tools becoming more important
    - User access to Help Desk ticket systems: R/T, Remedy, etc.
  - Specialized, local help by defacto local experts
    - End users instead of support professionals
- Complexity and interconnection of computing environments increasing demands for sophisticated users
  - Tools, compilers, access models
  - Need for code migration from platform to platform