Modern High Performance Computing Platforms

Rob Kunz
Head, Computational Mechanics Division
Penn State Applied Research Laboratory
rfk102@psu.edu
Contents

- What Do HPC Systems Do and Who Uses Them
- Overview of “Fastest” Systems
- Some History
- The Real Basics
- Forms of Parallelism
- System Classification
- HPC Centers
- HPC at PSU
- US Government Policy Issues
- Summary
Acknowledgements

- This lecture draws on:
  - top500.org, an excellent resource for HPC information
  - An annually released Technical paper entitled “Overview of Recent Supercomputers” by van der Steen (Utrecht University, Netherlands) and Dongarra (University of Tennessee, Knoxville)
  - Prof. Chau-Wen Tseng of University of Maryland course notes: amber.cs.umd.edu/class/838-s04/lec7.pdf
Disclaimer

- Your lecturer today is NOT a HPC expert, but is an experienced HPC user
- Numerous people on campus have significantly more expertise in HPC hardware:
  - Lyle Long, Vijay Agarwala, Abdul Aziz, Jason Holmes, Jeff Nucciarone, Phil Sorber, ……?
- Some of what I’ll be presenting includes opinions based on 20 years of Computational Fluid Dynamics research using HPC platforms
- This material is a rapidly moving target and I hope that this first lecture will serve as a starting point for an evolving component of the ICS course
- Please submit comments to ICS web site
What Do HPC Systems Do

- HPC systems are powerful computers or networks of computers that are used to:
  - Simulate systems that are difficult (or illegal!) to study experimentally
  - Test researchers’ understanding of physical systems
  - Simulate how new products will behave in different environments
  - Solve massive mathematical computations
  - Manage massive amounts of data
Who Uses HPC Systems

- **Scientific Applications:**
  - Climate modeling
  - Biology and biomedical sciences
  - Astrophysics
  - Particle physics
  - Geology
  - Earthquake predictions

- **National Defense Applications:**
  - National Security Agency (NSA) – code breaking
  - DOE National Nuclear Security Agency (NNSA) for modeling nuclear weapons
  - Data mining
Who Uses HPC Systems

- National Defense Applications (continued):
  - WMD dispersion
  - Engineering for weapons systems

- Industrial Applications:
  - Automotive engineering
  - Semiconductor design
  - Pharmaceutical development
  - Satellite/map generation
  - Telecommunications
  - Financial predictions
  - Entertainment
Who Uses HPC Systems

• TOP500 applications breakdown
  • #systems in TOP 500/performance

- Telecomm (8.0%)
- Weather and Climate Research (12.3%)
- Semiconductor (14.9%)
- Geophysics (21.1%)
- Others (30.9%)
- Digital Content Creation (6.9%)
- Database (6.0%)
Overview of Fastest Systems: TOP 500

- **TOP500** project:
  - Started in 1993 for tracking HPC trends
  - 2x per year, a list of the 500 most powerful computer systems is assembled and released.
  - **Linpack** benchmark is used

- Just last week (Nov 9, 2004), latest list was released at **SC2004** in Pittsburgh

- **IBM BlueGene**: 70.72 TFLOPS (70.72x10^{12} floating point operations per second)

- Supplanted Japanese **Earth simulator**
Overview of Fastest Systems: TOP 500; Nov 9, 2004

1. BlueGene/L
   DOE/IBM
   Rochester, USA
   BlueGene/L DD2
   Rmax: 70.72 TFlops

2. Columbia
   NASA/Ames
   Mountain View, USA
   SGI Altix/Voltaire
   Rmax: 51.87 TFlops

3. Earth Simulator
   Earth Simulator Center
   Yokohama
   NEC
   Rmax: 35.86 TFlops

4. MareNostrum
   Barcelona Supercomputer Center
   Barcelona, Spain
   eServer BladeCenter JS20/Myrinet
   Rmax: 20.53 TFlops

5. Thunder
   Lawrence Livermore National Lab
   Livermore, USA
   Intel Itanium2 Tiger4/Quadrics
   Rmax: 19.94 TFlops
Overview of Fastest Systems: TOP 500

- **LINPACK Benchmark:**
  - Performance measure introduced by Jack Dongarra at University of Tennessee
  - Solution of a dense system of linear equations using an LU factorization with partial pivoting
  - Does not reflect the overall performance of a given system ⇒ no single number ever can
  - Reflects the performance of a dedicated system for solving a dense system of linear equations
  - TOP500 contenders measure performance for different problem sizes \( n \) and find a maximal achieved performance \( R_{\text{max}} \) for problem size \( N_{\text{max}} \)
Overview of Fastest Systems: TOP 500

- **LINPACK Benchmark** (continued):
  - Theoretical peak performance $R_{\text{peak}}$ also reported
  - Paper computation to determine the theoretical peak rate of execution of floating point operations for the machine
  - Determined by counting the number of floating-point additions and multiplications that can be completed during a period of time, usually the cycle time of the machine.
  - E.g.: Intel Itanium 2 at 1.5 GHz can complete 4 floating point operations per cycle $\Rightarrow R_{\text{peak}} = 6$ GFlops

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site Country/Year</th>
<th>Computer / Processors Manufacturer</th>
<th>$R_{\text{max}}$</th>
<th>$R_{\text{peak}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM/DOE United States/2004</td>
<td>BlueGene/L beta-System BlueGene/L DD2 beta-System (0.7 GHz PowerPC 440) / 32768 IBM</td>
<td>70720</td>
<td>91750</td>
</tr>
</tbody>
</table>
Overview of Fastest Systems: TOP 500

- Systems are disappearing from the TOP 500 list faster than new ones are appearing
- Due largely to DOE’s Accelerated Strategic Computing Initiative (ASCI) program
  - Impetus to HPC industry
  - Heavy emphasis on COTS components
  - Large vendors favored – amortized costs
- Less emphasis on developing new architectures, however, the Japanese Earth Simulator “surprise” may help to reverse this trend
- More on this in last section: US Government Policy Issues
Overview of Fastest Systems: IBM BlueGene

- Distributed memory multiprocessor (MPP) architecture
- 32768 processors
- 0.7 GHz PowerPC 440 processor
- Theoretical performance:
  \[
  \left( \frac{0.7 \times 10^9 \text{ cycles}}{\text{sec}} \right) \left( \frac{4 \text{ FLOPS}}{\text{cycle}} \right) \left( 32768 \text{ processors} \right) = 91 \times 10^{12} \text{ FLOPS}
  \]

- LINPACK performance: 70.72\times10^{12} \text{ FLOPS}
- Designed for target $2^{16}=65,536$ compute node scaling
- Cellular architecture $\Rightarrow$ basic building block of the system can be replicated in a regular pattern, with no introduction of bottlenecks as the system is scaled up
- Each node is a “system on a chip” (processors, network interfaces, fast memory)
Overview of Fastest Systems: IBM BlueGene

- **Compute Chip**: 2 processors 2.8/5.6 GF/s, 4 MiB* eDRAM
  - (compare this with a 1988 Cray YMP/8 at 2.7 GF/s)

- **Node Card**: 16 compute cards 0-2 I/O cards, 32 nodes (64 CPUs) (4x4x2), 90/180 GF/s, 16 GIB* DDR

- **Cabinet**: 2 midplanes, 1024 nodes (2,048 CPUs) (8x8x16), 2.9/5.7 TF/s, 512 GIB* DDR, 15-20 kW

- **System**: 64 cabinets, 65,536 nodes (131,072 CPUs) (32x32x64), 180/360 TF/s, 32 TiB*, 1.2 MW, 2,500 sq.ft., MTBF 6.16 Days

Overview of Fastest Systems: SGI Columbia

- Distributed memory multiprocessor (MPP) architecture
- 10160 processors
- 1.5 GHz Intel Itanium 2 processor
- Theoretical performance:
  \[
  \left( 1.5 \times 10^9 \frac{\text{cycles}}{\text{sec}} \right) \left( 4 \frac{\text{FLOPS}}{\text{cycle}} \right) (10160 \text{ processors}) = 61 \times 10^{12} \text{ FLOPS}
  \]
- LINPACK performance: 51.87x10^{12} FLOPS
- Integrated cluster: 20 SGI Altix 512 processor systems
- 20 TB of memory
Overview of Fastest Systems: NEC Earth Simulator

- Distributed memory multiprocessor (MPP) architecture
- 5120 processors
- NEC vector processor
- Theoretical performance:
  \[
  \left( \frac{8 \times 10^9 \text{ FLOPS}}{\text{processor}} \right) (5120 \text{ processors}) = 41 \times 10^{12} \text{ FLOPS}
  \]
- LINPACK performance: $35.86 \times 10^{12} \text{ FLOPS}$
- 2GB per CPU, shared memory within 8-CPU nodes
- Multi-level parallelism $\Rightarrow$ optimized code:
  - MPI across nodes
  - OPEN-MP within node
  - Vectorization on processor
Overview of Fastest Systems: NEC Earth Simulator
Some History ⇒ Moore’s Law

- 1965 observation by Gordon Moore, co-founder of Intel:
  - Density of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented
  - In recent years data density has doubled approximately every 18 months, and this is the current definition of Moore's Law
  - Most experts expect Moore's Law to hold for at least another two decades

- The most common extension of Moore’s law is “fixed cost computer speed doubles every 18 months”
Some History ⇒ Moore’s Law

- Transistor density (scaling) remains only one of numerous contributors to overall platform speed:
  - Clock frequency not a main driver since 1980s
  - Processor parallelism
  - Compiler technology
  - Heterogeneous processors
  - Platform parallelism
  - Networks
Some History ⇒ Moore’s Law

≈ 2x every 14 months
Some History ⇒ Moore’s Law

- Another common extension of Moore’s law is “fixed compute speed halves in cost every 18 months”

Example: 1994; Cray model C90:
- Peak performance: 16 GFLOPS (Giga Floating Point Operations Per Second)
- $40,000,000 ⇒ 2.5 \times 10^{-3} \$/FLOP
- Extensive code development labor involved in optimizing for vector processor based platform

Example: 2004; 46 processor 2.2 GHz Xeon:
- Peak performance: 202 GFLOPS
- $65,000 ⇒ 3.2 \times 10^{-7} \$/FLOP (8000 times cheaper!)
- Virtually no code development labor involved in optimizing for platform since codes already conform to domain decomposition-message passing paradigm
The Real Basics

- All HPCs today are parallel, wherein numerous processors are simultaneously working your problem.

- Parallel computer = a collection of processing elements that can communicate and cooperate to solve large problems fast
  - Virtually all scientific or engineering problems of interest today lend themselves to scaling up one’s problem size interest to where 1 processor is inadequate
  - Platform must be fast enough to solve your problem fast enough to be of interest (hours ⇒ months)

- More cost effective than custom uniprocessor
  - Processor design / fabrication cost >> $100 million
  - Commercial microprocessor costs can be amortized
  - Architects running out of uses for ># transistors on 1 chip
Forms of Parallelism

- **Granularity:**
  - Refers to % of work that can be executed independently
  - Coarse grain parallelism $\Rightarrow$ infrequent synchronization
  - Fine grain parallelism $\Rightarrow$ frequent synchronization

Instruction level, vector operation, loop level, task level

Decreasing granularity
**Forms of Parallelism**

- **Pipeline parallelism:**
  - Refers to when a number of tasks can be overlapped.
  - Can be fine grain (instruction level) or coarse grain (task level).
  - Instruction pipelining – assembly line:

<table>
<thead>
<tr>
<th>Non-pipelined  (sequential)</th>
<th>Non-pipelined  (sequential)</th>
<th>Pipelined</th>
<th>Pipelined</th>
</tr>
</thead>
<tbody>
<tr>
<td>data element a</td>
<td>data element a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction fetch</td>
<td>Instruction decode</td>
<td>Instruction fetch</td>
<td>Instruction decode</td>
</tr>
<tr>
<td>Instruction decode</td>
<td>Execution</td>
<td>Instruction decode</td>
<td>Execution</td>
</tr>
<tr>
<td>Execution</td>
<td>Memory Access</td>
<td>Instruction decode</td>
<td>Memory Access</td>
</tr>
<tr>
<td>Write back</td>
<td></td>
<td>Execution</td>
<td>Write back</td>
</tr>
<tr>
<td>data element b</td>
<td>data element b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction fetch</td>
<td>Instruction decode</td>
<td>Instruction fetch</td>
<td>Instruction decode</td>
</tr>
<tr>
<td>Instruction decode</td>
<td>Execution</td>
<td>Instruction decode</td>
<td>Execution</td>
</tr>
<tr>
<td>Execution</td>
<td>Memory Access</td>
<td>Instruction decode</td>
<td>Memory Access</td>
</tr>
<tr>
<td>Write back</td>
<td></td>
<td>Execution</td>
<td>Write back</td>
</tr>
<tr>
<td>data element c</td>
<td>data element c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction fetch</td>
<td>Instruction decode</td>
<td>Instruction fetch</td>
<td>Instruction decode</td>
</tr>
<tr>
<td>Instruction decode</td>
<td>Execution</td>
<td>Instruction decode</td>
<td>Execution</td>
</tr>
<tr>
<td>Execution</td>
<td>Memory Access</td>
<td>Instruction decode</td>
<td>Memory Access</td>
</tr>
<tr>
<td>Write back</td>
<td></td>
<td>Execution</td>
<td>Write back</td>
</tr>
</tbody>
</table>
Vector Processing:
- Takes instruction pipelining one step further:
- In addition to pipelining instructions, vector processors also pipeline the data itself
- Arrays A, B, C; C=A+B can be handled by vector processor by processing chunks of arrays at once.
- Fewer fetch and decode instructions
- Vector processors are much more expensive than scalar processors
SIMD: All processors perform the same task on different data simultaneously (in lock step)
- Single instruction stream for all processors
- Limited flexibility
- Suitable for vector computations
**Forms of Parallelism**

- **MIMD**: Processors independently perform the same or different tasks on different data
  - Different instruction stream for each processor
  - Very flexible
  - Suitable for any “coarse grain” computation
  - Usually higher synchronization/communication costs
Forms of Parallelism

More MIMD:

- Multiple Program Multiple Data (MPMD) – each processor has a different program to execute
  - Complicated programming

- Single Program Multiple Data (SPMD) – each processor has same program to execute
  - Simpler programming

- e.g., CFD based on domain decomp coupled to NASTRAN (FEM) with parallelism for matrix inversion only

- e.g., CFD based on domain decomp
System Classification - Categories

- How do processors exchange data
  - Shared memory vs. distributed memory w/ message passing
- How are processors interconnected
  - Bus vs. switching network vs. loosely coupled network
- How do processors synchronize their execution
  - Shared memory: locks, barriers, monitors
  - Message passing: blocking msgs. / collective communications
- How frequently do processors coordinate execution
  - Granularity
System Classification – Architectures

- Flynn’s Taxonomy:
  - Shared memory SIMD
  - Distributed memory SIMD systems
  - Shared memory MIMD systems
  - Distributed memory MIMD systems

S/M=single/multiple processor, I=instruction, D=data

- Becoming a bit dated for HPC because:
  - Emergence of clusters
  - Disappearance of SIMD from truly high end systems
287/500 = clusters!
System Classification – Architectures

So for this top level overview we’ll go with:

- Shared memory multiprocessors (SMP)
- Distributed memory multiprocessors (MPP)
- Clusters (focus of next lecture)
- Constellations
- Grid computers
Shared Memory Multiprocessors (SMP)

- Tightly-coupled identical processors, shared memory
- Parallel nature hidden from the user ⇒ OS managing allocation of processor time
- Timesharing, multi-tasking OS that has >1 processor to choose from when scheduling programs to run
Shared Memory Multiprocessors (SMP)

- Processors usually low-cost RISC chips
- Memory can be physically co-located (Symmetric Multiprocessor – also SMP) or distributed (Nonuniform memory access - NUMA) – shared memory address space in either case
- Popular for workstations and servers due to relative ease of use; the single memory pool and the fact that the multiprocessing nature is hidden from the user
- But SMPs are not very scalable so they have all but disappeared from TOP500 since peaking at nearly 50% of systems ≅ 1997

SMP machines:
- Cray T3E
- SGI Challenge
- Sun Sunfire, Enterprise server
- HP SuperDome
Distributed Memory Multiprocessors (MPP)

- MPP architecture can be viewed as a "super-cluster" of relatively inexpensive processors, connected together with custom designed fast interconnects and supporting hardware.
- Fully distributed memory; each processor is self-contained, with its own cache and memory chips.
- MPP = Massively Parallel Processors.
Distributed Memory Multiprocessors (MPP)

- In TOP500, all 9 Cray, all 6 Hitachi, all 20 SGI, 55/216 IBMs and most NEC systems (including Earth Simulator) are MPP. Top 3 and 10 of Top 20.
- Typically off-the-shelf scalar RISC and CISC chips of the kind found in desktop workstations and PCs.
- Some Japanese vendors use vector chips including Earth Simulator.
- Since MPP's memory is distributed to each processor, connecting large numbers of processors together more straightforward than in the SMP model – better scalability.
- No complex bus and controller architecture to allow processors to access the same memory area.
- Key design feature is fast interconnecting network.
Distributed Memory Multiprocessors (MPP)

- User needs to be much more aware of the nature of the machine in order to reap the benefits
- However, given the large # of processors, large available memory, high throughout and scalability, well-designed and well-implemented parallel programs perform very well ⇒ MPPs remain strong on TOP500, including current top 3
- Examples of MPP systems:
  - NEC Vector SX6
  - Cray T3E, X1 Systems
  - SGI Altix Systems
  - IBM SP/2-4, BlueGene Systems
  - Thinking Machines CM-2/5 Systems (ancient history)
Clusters

- Large collections of loosely connected PC/SMP nodes
- Beowulf (LINUX PC), HP AlphaServer most common
- Message passing for interprocessor communication
- Range of slower to very fast network interconnections
- Did not exist 10 years ago
- Explosively popular due to:
  - Scalability
  - Large memory
- Significant presence in TOP500:
- Focus of next lecture
Constellations

- Small/medium collections of fast vector nodes
- Large memory and moderate scalability
- High performance processor nodes
- Hybrid memory model
- TOP500 classify only the Fujitsu PRIMEPOWER and HP Superdome architectures as constellations
- Some consider IBM BlueGene and NEC Earth Simulator constellations by relaxing the definition of vector nodes to “processors that incorporate some degree of pipelining”
Grid Computers

- Parallel systems of geographically distributed "autonomous" resources
- Component processors dynamically allocated at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements
- Workstations networks are the most well known examples
- Clusters vs. grids: how resources are managed:
  - Clusters: resource allocation is performed by a centralized resource manager (head node) and all nodes cooperatively work together as a single unified resource
  - Grids: each node has its own resource manager and don't aim for providing a single system view
- Not in TOP500 but offer very cost effective HPC for many enterprises
Parallel Architecture Trends

- Faster interconnects
  - Not keeping up with node performance in clusters
- Larger memories per processor
- Deeper cache hierarchies per processor
- Processor-memory integration
HPC Centers

- DoD’s High Performance Computing Modernization Program – **HPCMP**
  - HPC resources, networks and software for DoD labs and test centers
  - 4 major centers:
    - US Army Research Lab (ARL); MD
    - Aeronautical Systems Center (ASC); OH
    - US Army Engineer Research and Development Center (ERDC); MS
    - Naval Oceanographic Office (NAVO); MS; (2)
  - Numerous “distributed” centers
  - “Challenge” resource grants for solving state-of-the-art problems (of military significance)
HPC Centers

- DoE currently supports 3 large supercomputer centers, 4,000 users (military (most), industry and academia):
  - Los Alamos National Laboratory, Los Alamos, NM
  - National Energy Research Supercomputer Center, Lawrence Livermore National Laboratory
  - Oak Ridge National Laboratory, Oak Ridge, TN
HPC Centers

- NSF currently supports 3 large supercomputer centers, 3,000 users (most academic):
  - San Diego Supercomputing Center at the University of California-San Diego
  - National Center for Supercomputing Applications at the University of Illinois Urbana-Champaign
  - Pittsburgh Supercomputing Center, run by Carnegie Mellon University and the University of Pittsburgh
HPC at PSU

- **HPC group at PSU:**
  - Mission to empower PSU scholars in their ability to compute and manage data by developing and maintaining several state-of-the-art computational clusters
  - Software support
  - Code optimization and parallelization support for PSU systems and systems at major national centers
- **Staff**
  - Vijay Agarwala, Director HPC & Viz Group, vijay@psu.edu
  - Abdul Aziz, Research Programmer, aha103@psu.edu
  - Jason Holmes, Research Programmer, jholmes@psu.edu
  - Jeff Nucciarone, Senior Research Programmer, nucci@psu.edu
  - Phil Sorber, Computer Support Specialist, aafes@psu.edu
HPC at PSU

- Large systems:
  - LION-XL: 352 proc, 2.4-2.8 GHz P4, Quadrics network
  - LION-XM: 336 proc, 3.06-3.2 GHz Xeon, Myrianet network
  - LION-XO: 160 proc, 2.4 GHz Opteron, Infinicon Network

- Smaller systems for interactive, debug code development, 64-bit large applications

- Accounts available for PSU students and faculty
## HPC at PSU: TOP 500 presence

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site Country/Year</th>
<th>Computer / Processors Manufacturer</th>
<th>$R_{\text{max}}$</th>
<th>$R_{\text{peak}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>222</td>
<td>Pennsylvania State University United States/2004</td>
<td><strong>LION-XM PowerEdge 1750, P4 Xeon 3.06 GHz, Myrinet</strong> / 336 Dell</td>
<td>1319.67</td>
<td>2056.32</td>
</tr>
<tr>
<td>480</td>
<td>Pennsylvania State University United States/2003</td>
<td><strong>LION-XL PowerEdge 2650 Cluster P4 Xeon 2.4 GHz - Quadrics</strong> / 256 Dell</td>
<td>877.9</td>
<td>1228</td>
</tr>
<tr>
<td>484</td>
<td>Pennsylvania State University United States/2003</td>
<td><strong>Fire V60 Cluster, 2.8 GHz, Gig Ethernet</strong> / 320 Sun</td>
<td>872.269</td>
<td>1792</td>
</tr>
</tbody>
</table>
HPC at PSU : TOP 500 presence

- Sun Fire v60 cluster:
  - Run for Dr. Sam Finn, Director of the Center for Gravitational Wave Physics.
  - Funded by IVDGL/LIGO project (a multi-university project to prove or disprove the existence of gravity waves)
  - Not available to the general university.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>484</td>
<td>11/2004</td>
</tr>
<tr>
<td>297</td>
<td>06/2004</td>
</tr>
<tr>
<td>156</td>
<td>11/2003</td>
</tr>
</tbody>
</table>
HPC at PSU: TOP 500 presence

Pennsylvania State University

Diagram showing the presence of different computing systems at PSU over various years, with details such as model, processor type, and network configuration.
**US Government Policy Issues**

- Japanese Earth Simulator was a “surprise”
- US military establishment has driven and led HPC since WWII
- US government has responded
- US industry has responded
- In May 2004, House of Reps met to examine federal HPC R&D and to consider legislation to help re-establish US leadership in HPC
- Timely since several federal agencies are in the process of reformulating their HPC programs in part in response to the challenge posed by Japan
US Government Policy Issues

- Important questions that need to be addressed:
  - How does HPC affect international competitiveness of US scientific enterprise?
  - What are/must DOE, DoD, NSF doing to assure US leadership in HPC?
  - Where should HPC research expenditures be targeted?
- The depth and strength of U.S. capability stems in part from the sustained research and development program carried out by federal science agencies under an interagency program codified by the High-Performance Computing Act of 1991. Widely credited with reinvigorating U.S. HPC capabilities after a period of relative decline during the late 1980s.
However, in recent years, federal agency efforts have lost momentum as federal activities began focusing less on high-performance computing and more on less specialized computing and networking technologies.

Earth Simulator cost $350M for Japan to build from custom-made components, something US programs stopped doing in the 1990s.

US programs chose to favor commercially available components for HPC systems – more cost effective.

Many now believe that such an approach has left US researchers in some fields, like climate modeling, at a disadvantage, since the algorithms for these models are not well suited to commodity clusters.
US Government Policy Issues

- White House Office of Science and Technology Policy (OSTP) responding to this flagging of US efforts to develop/deploy HPCs created the inter-agency High-End Computing Revitalization Task Force (HEC-RTF)

- In April 2004, Rep Judy Biggert (IL) introduced High Performance Computing Revitalization Act of 2004:
  - Updates HPC Act of 1991
  - Provides law and $$ to ensure sustained access by research community in US to highest end HPC systems

- So what's happening:
  - DOE and NSF (lead agencies responsible for providing HPC resources for US civilian research) are moving in several new directions
So what's happening (continued):

- NSF placing greater emphasis on grid computing
- NSF continues support of 3 HPC centers @ $70M/yr
- DOE expanding access to large single location machines
- DOE office of Science initiated new effort in development of next-generation computer architectures (NGA). $38M/yr. < emphasis on commercial components & needs
- DOE’s National Energy Research Scientific Computing Center (NESRC) at Lawrence Berkley Lab provides HPC to > 2000 scientists annually ⇒ budget *2 since 2003
- DARPA funding a Phase II to Cray, IBM and Sun to develop next generation of HPCs for National Security and industrial applications by 2010
So in summary:

- Our federal leadership has recognized and acted on the belief that world leadership by the US in HPC is critical to our economic competitiveness and national security.
- HPC users in the US will have more access to ever more rapidly improving state-of-the-art HPC systems whether in the civilian research, military or industrial arenas.
Summary

- HPCs play a critical role in the scientific, military and industrial competitiveness of the US
- The super-Moore’s law rate of HPC power increase is contributing dramatically to analysis capability and strategic thrusts ⇒ need to more than keep up, need to plan for how to use PFLOP systems by 2010!
- Clusters play an ever increasing role in our lives due principally to their net cost but recent US strategy shifts may lean us back towards use of high end specialty systems
- All PSU researchers have excellent and ever improving access to local and off-site HPC resources