High Performance Computing

Dheeraj Bhardwaj
Department of Computer Science & Engineering
Indian Institute of Technology, Delhi –110 016 India
http://www.cse.iitd.ac.in/~dheerajb

Introduction

- **Traditional Science**
  - **Observation**
  - **Theory**
  - **Experiment -- Most expensive**
- **Experiment can be replaced with Computers Simulation - Third Pillar of Science**

What do we want to Achieve ?

- **Develop High Performance Computing Applications (HPC) which are**
  - **Portable (Laptop ➡ Supercomputers ➡ Grid)**
  - **Future Proof**
  - **Grid Ready**

- **Develop HPC Infrastructure (Parallel & Grid Systems) which is**
  - **User Friendly**
  - **Based on Open Source**
  - **Efficient in Problem Solving**
  - **Able to Achieve High Performance**
  - **Able to Handle Large Data Volumes**

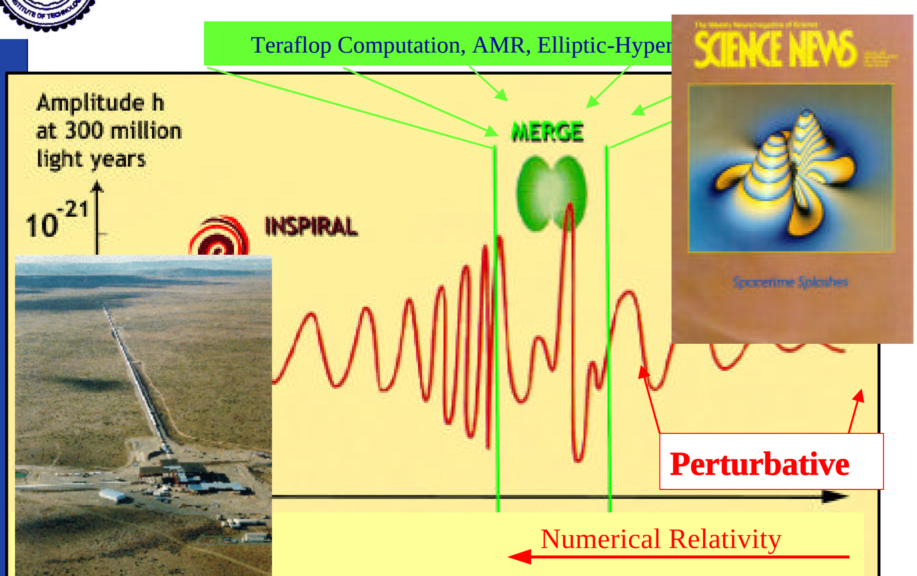
Where HPC is being Used?

- **Scientific & Engineering Applications**
  - **Simulation of physical phenomena**
  - **Virtual Prototyping (Modeling)**
  - **Data analysis**
- **Business / Industry Applications**
  - **Data warehousing for financial sectors**
  - **E-governance**
  - **Medical Imaging**
  - **Web servers, Digital libraries**
  - **Financial Markets**
    - **Modeling currencies**
    - **Share Market Forecast**
  - **Airlines, insurance companies**
  - **Transaction, data, etc**

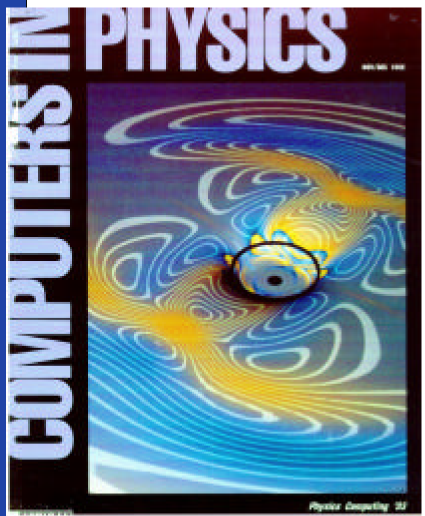
HPC Applications

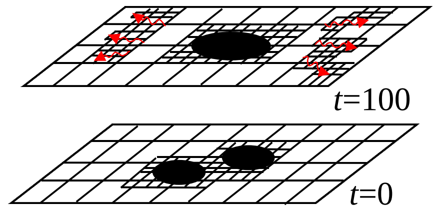
- All face similar problems
  - Not enough computational resources
  - Remote facilities – Network becomes the bottleneck
  - Heterogeneous and fast changing systems
- Three Types
  - **High-Capacity** – Grand Challenge Applications
  - **Throughput** – Running hundreds/thousands of job, doing parameter studies, statistical analysis etc…
  - **Data** – Genome analysis, Particle Physics, Astronomical observations, Seismic data processing etc
- We are seeing a Fundamental Change in HPC Applications
  - They have become multidisciplinary
  - Require incredible mix of varies technologies and expertise

High Capacity Computing: Want to Compute What Happens in Nature!



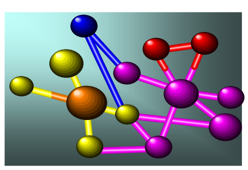
Computation Needs: 3D Numerical Relativity



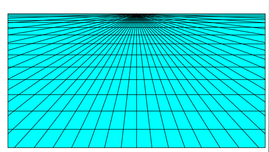


- Get physicists + CS people together
- Find Resource (TByte, TFlop crucial)
- Initial Data: 4 coupled nonlin. elliptics
- Choose Gauge (elliptic/hyperbolic…)
- Evolution
  - "hyperbolic" evolution
  - coupled with elliptic eqs.
- Find Resource ….
- Analysis: Interpret, Find AH, etc

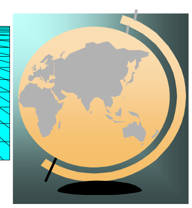
Need of more Computing Power: Grand Challenge Applications



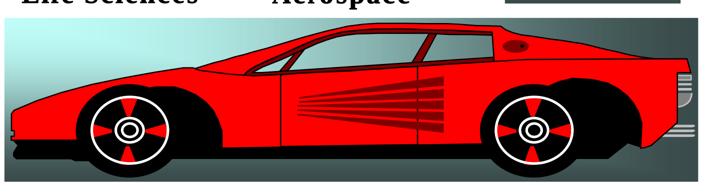
Life Sciences



Aerospace



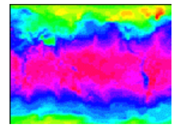
Geographic Information Systems



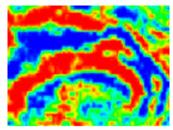
Mechanical Design & Analysis (CAD/CAM)

Need of more Computing Power: Grand Challenge Applications

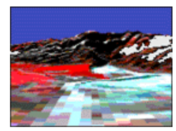
- **Weather Forecasting**



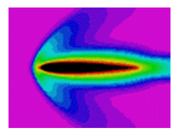
- **Seismic Data Processing**



- **Remote Sensing, Image Processing & Geomatics**



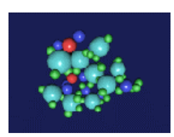
- **Computational Fluid Dynamics**
- **Astrophysical Calculations**



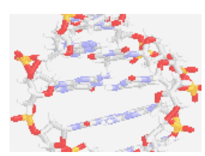
Grand Challenge Applications

Scientific & Engineering Applications

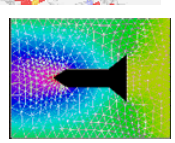
- **Computational Chemistry**



- **Molecular Modelling**

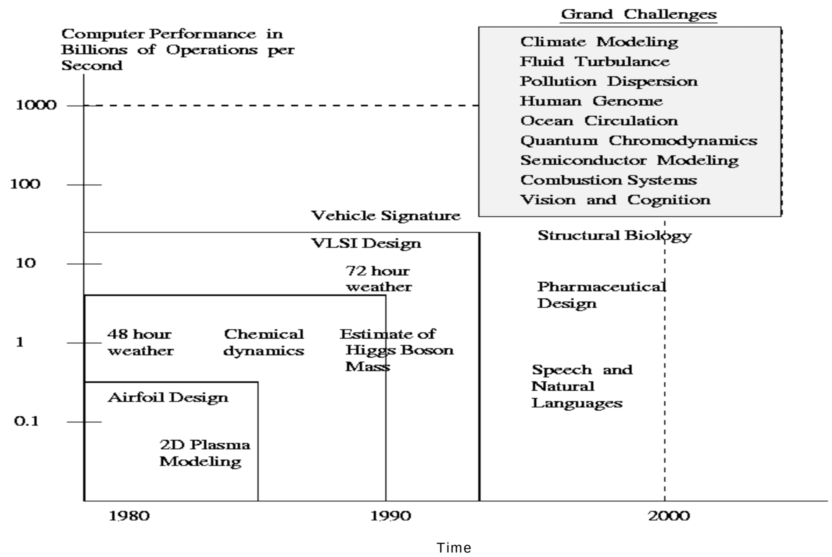


- **Molecular Dynamics**
- **Bio-Molecular Structure Modelling**



- **Structural Mechanics**

Requirements for Applications



HPC Applications Requirements

- **Scientific and Engineering Components**
  - **Physics, Biology, CFD, Astrophysics, Computer Science and Engineering ...etc..**
- **Numerical Algorithm Components**
  - **Finite Difference/ Finite Volume /Finite Elements etc...**
  - **Dense Matrix Algorithms**
  - **Solving linear system of equations**
  - **Solving Sparse system of equations**
  - **Fast Fourier Transformations**

HPC Applications Requirements

- Non - Numerical Algorithm Components
  - Graph Algorithms
  - Sorting algorithms
  - Search algorithms for discrete Optimization
  - Dynamic Programming
- Different Computational Components
  - Parallelism (MPI, PVM, OpenMP, Pthreads, F90, etc..)
  - Architecture Efficiency (SMP, Clusters, Vector, DSM, etc..)
  - I/O Bottlenecks (generate gigabytes per simulation)
  - High Performance Storage (High I/O throughput from Disks)
  - Visualization of all that comes out
- Scientist/eng. wants to focus on top, but *all* required for results...
- Such work cuts across many disciplines, areas of CS...
- And now how we do it ??!!

High Throughput Computing: Task farming

- Running hundreds - millions ++ of jobs as quickly as possible
- Collecting statistics, doing ensemble calculations, surveying large parameter space, etc
- Typical Characteristics
  - Many small, independent jobs: must be managed!
  - Usually not much data transfer
  - Sometimes jobs can be moved from site to site
- Example Problems: climatemodeling.com, NUG30
- Later: examples that combine "capacity" and "throughput"

Large Data Computing

- Data: more and more the "killer app" for the Grid
  - Data mining:
    - Looking for patterns in huge databases distributed over the world
    - E.g. Genome analysis
  - Data analysis:
    - Large astronomical observatories
    - Particle physics experiments
    - Huge amounts of data from different locations to be correlated, studied
  - Data generation
  - Resources Grow: Huge simulations will each generate TB-PB to be studied
- Visualization
  - How to visualize such large data, here, at a distance, distributed
- Soon: Dynamic combinations of all types of computing, data & on grids
- Our Goal is to give strategies for dealing with all types of computing

HPC Applications Issues

- Architectures and Programming Models
  - Distributed Memory Systems MPP, Clusters – Message Passing
  - Shared Memory Systems SMP – Shared Memory Programming
  - Specialized Architectures – Vector Processing, Data Parallel Programming
  - The Computational Grid – Grid Programming
- Applications I/O
  - Parallel I/O
  - Need for high performance I/O systems and techniques, scientific data libraries, and standard data representation
- Checkpointing and Recovery
- Monitoring and Steering
- Visualization (Remote Visualization)
- Programming Frameworks

Future of Scientific Computing

- Require Large Scale Simulations, beyond reach of any machine
- Require Large Geo-distributed Cross Disciplinary Collaborations
- Systems getting larger by 2- 3- 4x per year !!
  - Increasing parallelism: add more and more processors
- New Kind of Parallelism: GRID
  - Harness the power of Computing Resources which are growing

High Performance Computing: A brief History

- Mainframe Era (1960 – early 1970)
  - FORTAN became important
- Vector Processors Era (Mid 1970 – Mid 1980)
  - Pipe line floating point processor
  - Attach special devices for I/O
  - FFT was major focus

High Performance Computing: A brief History

- **Supercomputer Era (Mid 1980 – early 1990)**
  - **Most dramatic period in the advancement of High Performance Computers.**
  - **CRAY and CONVEX were the leaders**
  - **FORTRAN and special purpose libraries**
  - **I/O performance improved substantially**

- **SMP/MMP Era (Early 1990 – late 1990)**
  - **Unix popularity**
  - **Potential of highly parallel computations**
  - **Gained popularity due to easy programming paradigm**

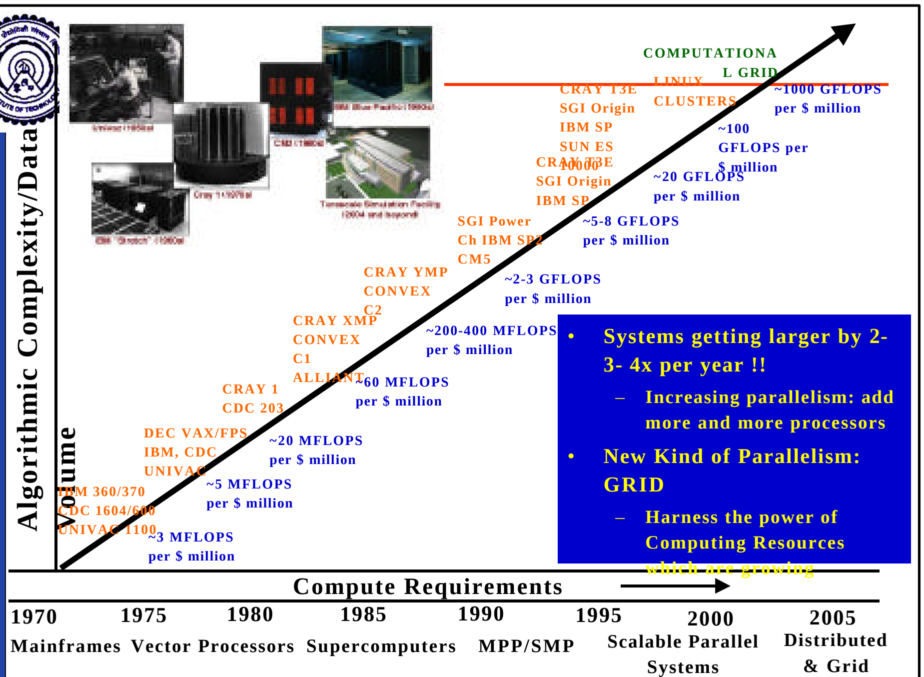
High Performance Computing: A brief History

- **Scalable Parallel System Era (Late 1990 –2001)**
  - **Golden period in the advancement of microprocessors and network technology**
  - **Processors were available at much cheaper price**
  - **Clustering of microprocessors became the technology for high performance computing**

High Performance Computing: A brief History

- **Distributed & GRID Computing Era (Present - Future)**
  - **Become very popular along with World Wide Web.**
  - **Good price/performance**

- **GRID: Computational Grid is an emerging infrastructure that enables the integrated use of remote high-end computers, databases, scientific instruments, networks and other resources.**



Why Parallel Computing ?

- If your Application requires more computing power than a sequential computer can provide ? !!!!!
  - You might suggest to improve the operating speed of processor and other components
  - We do not disagree with your suggestion BUT how long you can go ?
- We always have desire and prospects for greater performance

Performance

Three ways to improve the performance

- Work harder - Using faster hardware
- Work smarter - - doing things more efficiently (algorithms and computational techniques)
- Get help - Using multiple computers to solve a particular task.

Parallel Computing is the right answer

Serial and Parallel Computing

A parallel computer is a "Collection of processing elements that communicate and co-operate to solve large problems fast".

SERIAL COMPUTING

- Fetch/Store
- Compute

PARALLEL COMPUTING

- Fetch/Store
- Compute/communicate
- Cooperative game

Weather Modeling and Forecasting

Consider 3000 X 3000 miles, and height of 11 miles.

For modeling partition into segments of 0.1X0.1X0.1 cubic miles = ~10^{11} segments.

Lets take 2-day period and parameters need to be computed every 30 min. Assume the computations take 100 instrs. A single update takes 10^{13} instrs. For two days we have total instrs. of 10^{15} . For serial computer with 10^{10} instrs./sec, this takes 280 hrs to predict next 48 hrs !!

Lets take 1000 processors capable of 10^{8} instrs/sec. Each processor will do 10^{8} segments. For 2 days we have 10^{12} instrs. Calculation done in 3 hrs !!

Currently all major weather forecast centers (US, Europe, Asia) have supercomputers with 1000s of processors.

Issues in Parallel Computing

- **Design of parallel computers :** Design so that it scales to a large # of processor and are capable of supporting fast comunication and data sharing among processors.
- **Design of Efficient Algorithms :** Designing parallel algorithms are different from designing serial algorithms. Significant amount of work is being done for numerical and non-numerical parallel algorithms
- **Methods of Evaluating Parallel Algorithms :** Given a parallel computer and a parallel algorithm we need to evaluate the performance of the resulting system. How fast problem is solved and how efficiently the processors are used.

Issues in Parallel Computing

- **Parallel Computer Languages :** Parallel algorithms are implemented using a programming language. This language must be flexible enough to allow efficient implementation and must be easy to program in. Must efficiently use the hardware.
- **Parallel Proramming Tools :** Tools (compilers, libraries, debuggers, other monitoring or performance evaluation tools) must shield users from low level machine characteristics.
- **Portable Parallel Programs:** This is one of the main problems with current parallel computers. Program written for one parallel computer require extensive work to port to another parallel computer.

Issues in Parallel Computing

- **Automatic Programming of Parallel Computers :** This is about design of parallelizing compilers which extract implicit parallelism from programs that have not been explicitly parallelized. But this approach has limited potential for exploiting the power of large parallel machines.

Serial and Parallel Algorithms - Evaluation

- **Serial Algorithm**
  - **Execution time as a function of size of input**
- **Parallel Algorithm**
  - **Execution time as a function of input size, parallel architecture and number of processors used**

Parallel System

A parallel system is the combination of an algorithm and the parallel architecture on which its implemented

Applications – Commercial computing

Commercial Computing

- The database is much too large to fit into the computer's memory
- Opportunities for fairly high degrees of parallelism exist at several stages of the operation of a data base management system.
- Millions of databases have been used in business management, government administration, Scientific and Engineering data management, and many other applications.
- This explosive growth in data and databases has generated an urgent need for new techniques and tools.

Applications – Commercial computing

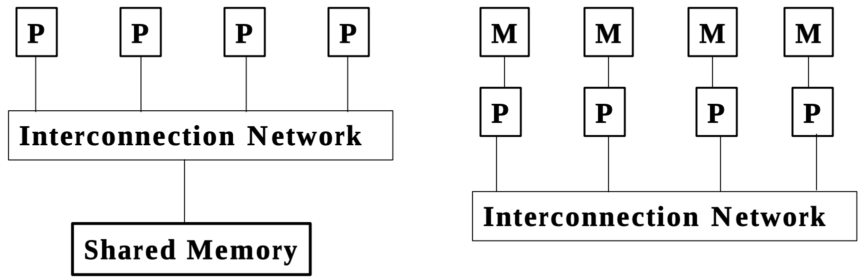
Sources of Parallelism in Query Processing

- Parallelism within Transactions (on line transaction processing)
- Parallelism within a single complex transactions.
- Transactions of a commercial database require processing large complex queries.

Parallelizing Relational Databases Operations

- Parallelism comes from breaking a relational operations (Ex : JOIN)
- Parallelism comes from the way these operations are implemented.

General Purpose Parallel Computer



Shared Memory Architecture

Distributed Memory Architecture

Issues in Parallel Computing

- Design of Parallel Computers
  - Different Models of Architecture
  - Node Configuration in each model
  - Interconnection Network
  - System Software
- Engineering Design Considerations
- Design of efficient parallel algorithms

Issues in Parallel Computing

- Performance Metrics used
- Performance and Scalability : System and Application Benchmarks
- Parallel Algorithmic Paradigms and Programming models
    - Parallel computer languages
    - Automatic programming of parallel computers

Issues in Parallel Computing

(Contd...)

- Performance
- Cost
- Productivity
- Reliability
- Availability

Important Issues in Parallel Programming

- ❖ Partitioning of data
- ❖ Mapping of data onto the processors
- ❖ Reproducibility of results
- ❖ Synchronization
- ❖ Scalability and Predictability of performance

Designing Parallel Algorithm

- ❖ Detect and exploit any inherent parallelism in an existing sequential Algorithm
- ❖ Invent a new parallel algorithm
- ❖ Adopt another parallel algorithm that solves a similar problem

Requirements for Applications

- Parallel I/O
- Optimized libraries
- Low latency and High bandwidth networks
- Scalability of a parallel system

Success depends on the combination of

- Architecture, Compiler, Choice of Right Algorithm, Programming Language
- Design of software, Principles of Design of algorithm, Portability, Maintainability, Performance analysis measures, and Efficient implementation

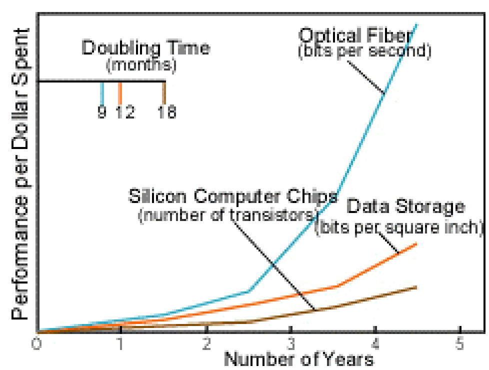
Computational Grid

A Computational Grid is an emerging infrastructure that enables the integrated use of remote high-end computers, databases, scientific instruments, networks and other resources

- **Computational Resources Scattered Across the World**
  - **Computer servers (double each 18 months)**
  - **Handhelds**
  - **File servers**
  - **Networks (Double each 9 months)**
  - **Databases etc....**
- **How take advantage of this for Scientific/Engineering Simulations?**
  - **Harness multiple sites and devices**
  - **Simulation at new level of complexity and scale, interacting with data**

Network Exponentials

- **Network vs. computer performance**
  - **Computer speed doubles every 18 months**
  - **Network speed doubles every 9 months**
  - **Difference = order of magnitude per 5 years**
- **1986 to 2000**
  - **Computers: x 500**
  - **Networks: x 340,000**
- **2001 to 2010**
  - **Computers: x 60**
  - **Networks: x 4000**



Moore's Law vs. storage improvements vs. optical improvements. Graph from Scientific American (Jan-2001) by Cleo Vilett, source Vined Khoslan, Kleiner, Caufield and Perkins.

Why Grid ?

We are seeing a Fundamental Change in Applications

- They have become multidisciplinary
- Require incredible mix of varies technologies and expertise

"Many problems require tightly coupled computers, with low latencies and high communication bandwidths; Grid computing may well increase … demand for such systems by making access easier" - Foster, Kesselman, Tuecke *The Anatomy of the Grid*

Motivation: When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances.

Glider Technology Report, June 2002

The GRID: Definition

Hardware and software infrastructure which provides Dependable, Consistent, Pervasive and inexpensive access to [high-end] resources

- ✓ **Dependable**: The hardware and software that supports Grid must be highly available with some level of "assurance".
- ✓ **Consistent:** Standard interface to the Grid must be available .
- ✓ **Pervasive:** Grid must be available in what ever environment potential Grid users wish to run.

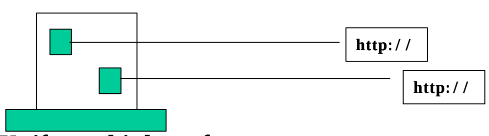
Distributed Computing vs. GRID

- Grid is an evolution of distributed computing
  - Dynamic
  - Geographically independent
  - Built around standards
  - Internet backbone

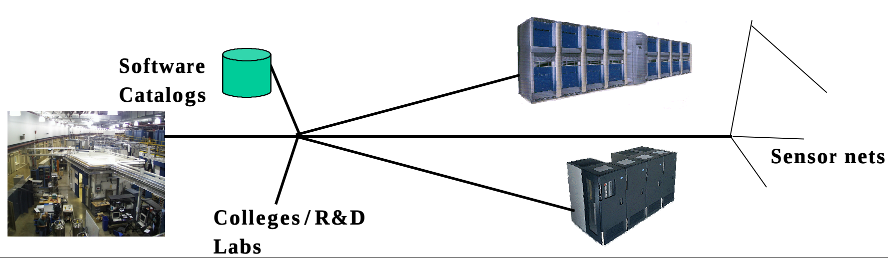
- Distributed computing is an "older term"
  - Typically built around proprietary software and network
  - Tightly couples systems/organization

Web vs. GRID

- **Web**
  - Uniform naming access to documents



- **Grid - Uniform, high performance access to computational resources**



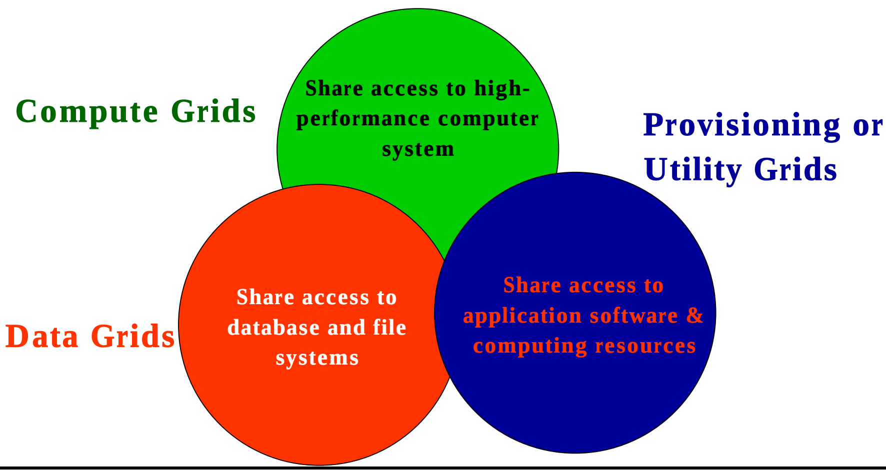
Is the World Wide Web a Grid ?

- Seamless naming? Yes
- Uniform security and Authentication? No
- Information Service? Yes or No
- Co-Scheduling? No
- Accounting & Authorization ? No
- User Services? No
- Event Services? No
- Is the Browser a Global Shell ? No

What does the World Wide Web bring to the Grid ?

- Uniform Naming
- A seamless, scalable information service
- A powerful new meta-data language: XML
  - XML will be standard language for describing information in the grid
  - SOAP – simple object access protocol
    - Uses XML for encoding. HTML for protocol
  - SOAP may become a standard RPC mechanism for Grid services
    - Uses XML for encoding. HTML for protocol
- Portal Ideas

- **Grid types are defined by shared resources**



Data Grid for High Energy Physics

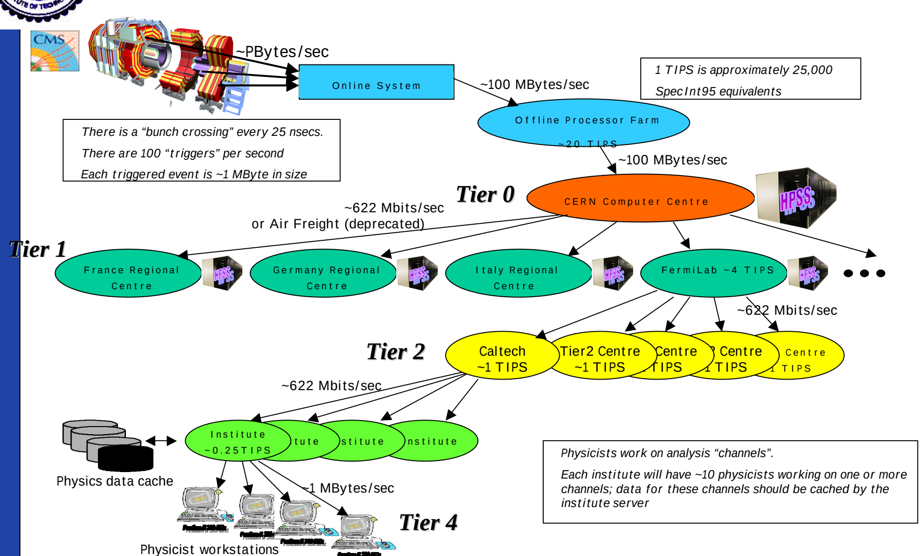
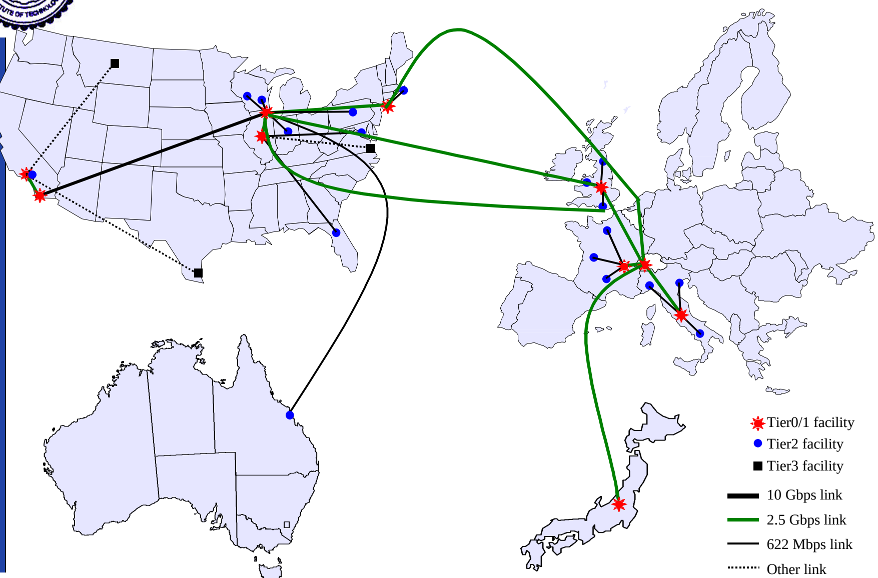


Image courtesy Harvey Newman, Caltech

iVDGL: International Virtual Data Grid Laboratory



U.S. PIs: Avery, Foster, Gardner, Newman, Szalay www.ivdgl.org

The Grid

An emerging infrastructures that enables the integrated use of remote high-end computers, databases, scientific instruments, networks and other resources.

The GRID is all about

- **The Coordinated, Transparent, Secure and Effective Utilization of Geographically distributed heterogeneous resources (both hardware & Software) for Applications**

To be Successful

- **The Grid has to support applications in the same way that the power utilities support the use of household appliances**

The Metaphor

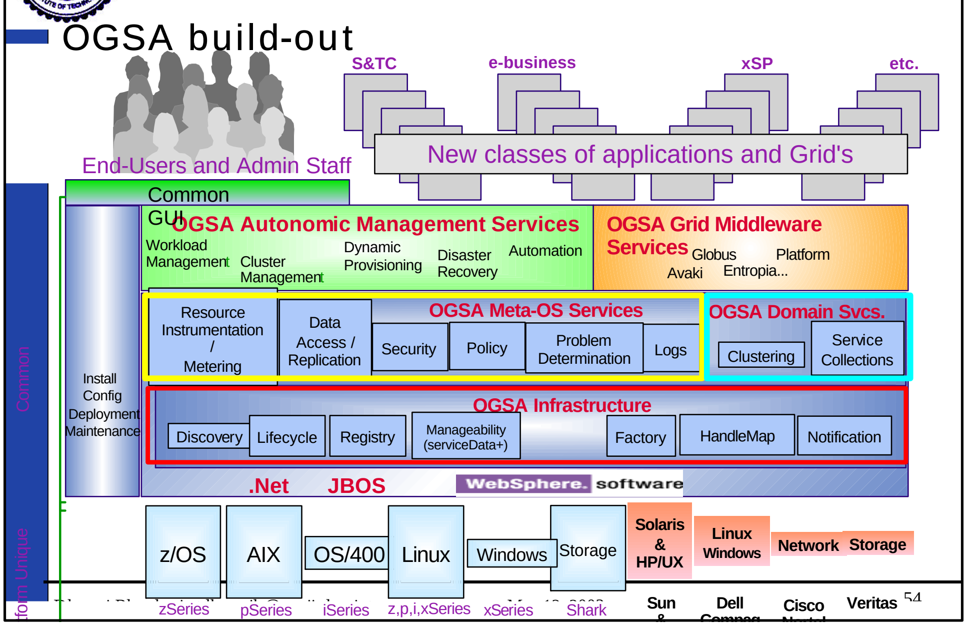
- **Computers to act as generators of computational "power", for applications to become computational appliances**
- **The software infrastructure to act as the utility responsible for managing the interaction between them**

Whom Does Grid Computing Serve ?

- **The users and Their Applications**
- **Large Complex Applications which need resources beyond the traditional**
  - **Parallel / Distributed processing in a box**
  - **Put-it-yourself together Clusters**
- **Applications that describe multiple aspects of a system**
- **Applications consisting of multiple modules**
- **Applications with multi-source data**
- **Applications interfacing with measurement systems and visualization systems**

Application Programmers will be able to write applications that leverage *TeraFlops* computations amd *PetaBytes* storage

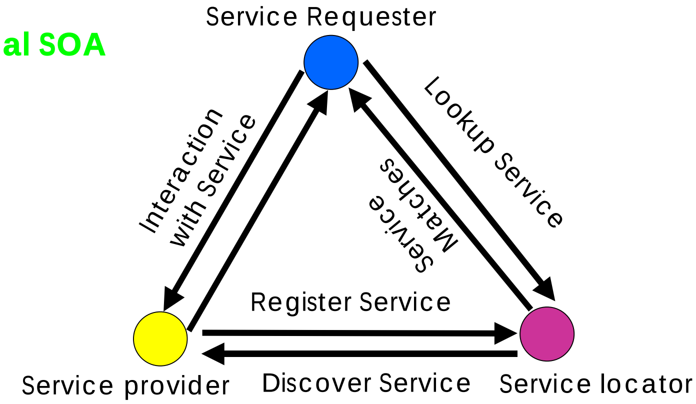
Open Grid Service Architecture



Grid Architecture

- Development of service oriented grid middleware using different technologies (such as Java/Jini, web services) to instantiate the service architecture.

A typical SOA



The Globus Project™

Making Grid computing a reality

- **Close collaboration with real Grid projects in science and industry**
- **Development and promotion of standard Grid protocols to enable interoperability and shared infrastructure**
- **Development and promotion of standard Grid software APIs and SDKs to enable portability and code sharing**
- **The Globus Toolkit™: Open source, reference software base for building grid infrastructure and applications**
- **Global Grid Forum: Development of standard protocols and APIs for Grid computing**

Convergence

- Grid technologies are currently distinct from other major technology trends, such as internet, enterprise, distributed, and peer-to-peer computing.

- All these technologies are going to merge in GRID technologies by making use of them.

From “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”

PARAM Padma

CONCLUSION

High Performance Computing is changing dimensions rapidly.