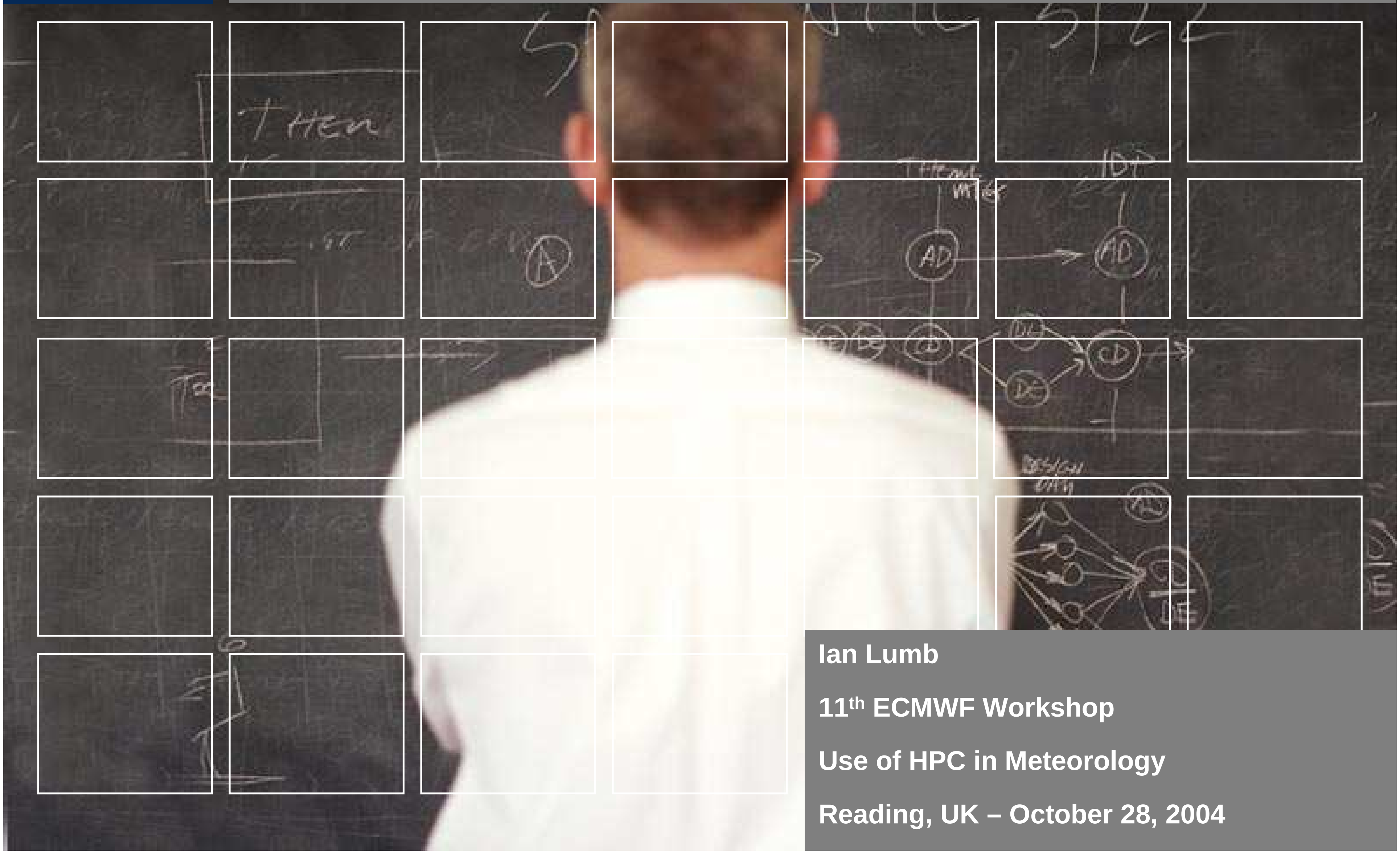
Platform™

Production HPC on Commodity Linux Clusters: The Role of Infrastructural Software



Ian Lumb

11th ECMWF Workshop

Use of HPC in Meteorology

Reading, UK – October 28, 2004

Outline

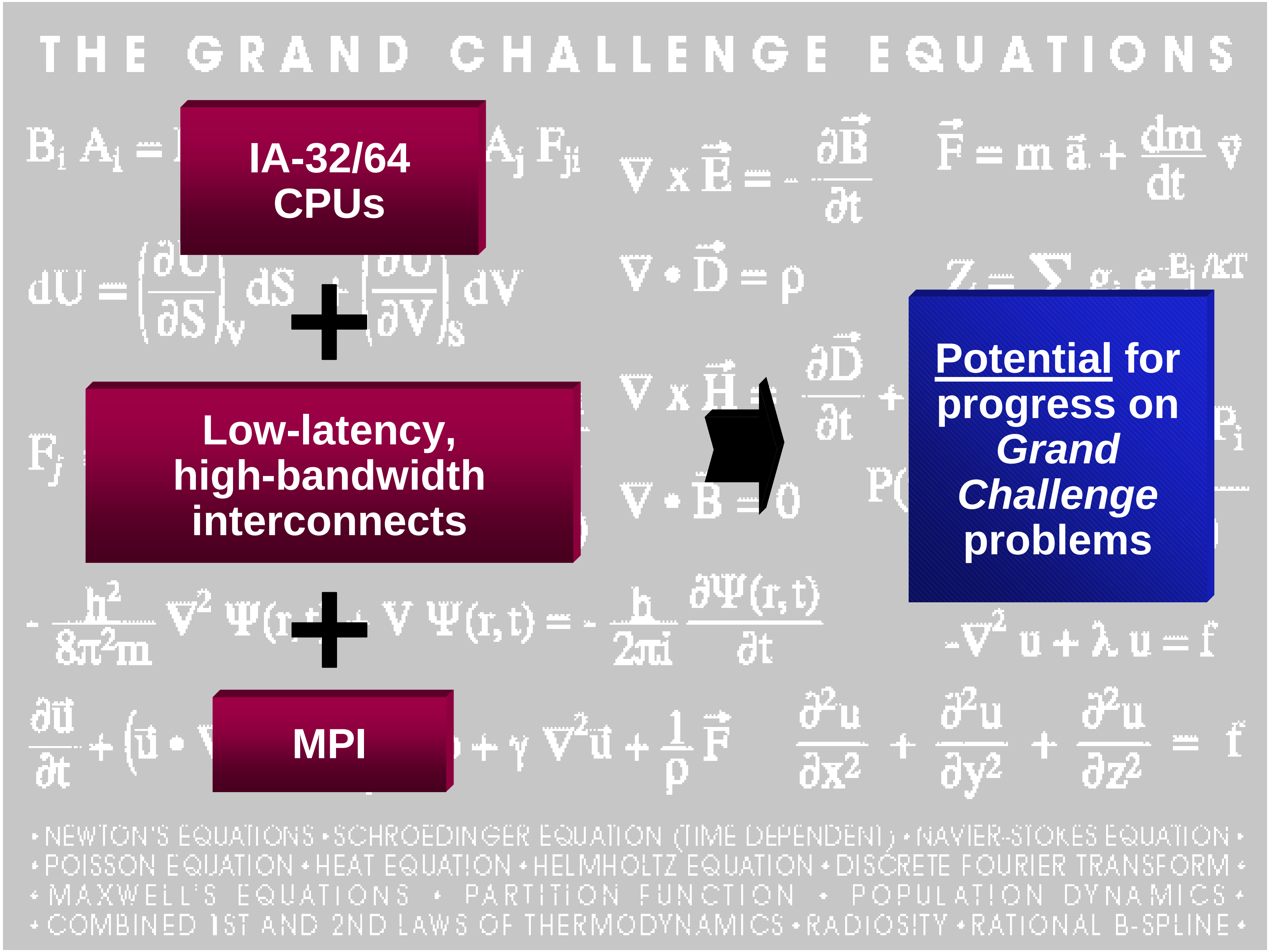
Introduction

Real-World Examples

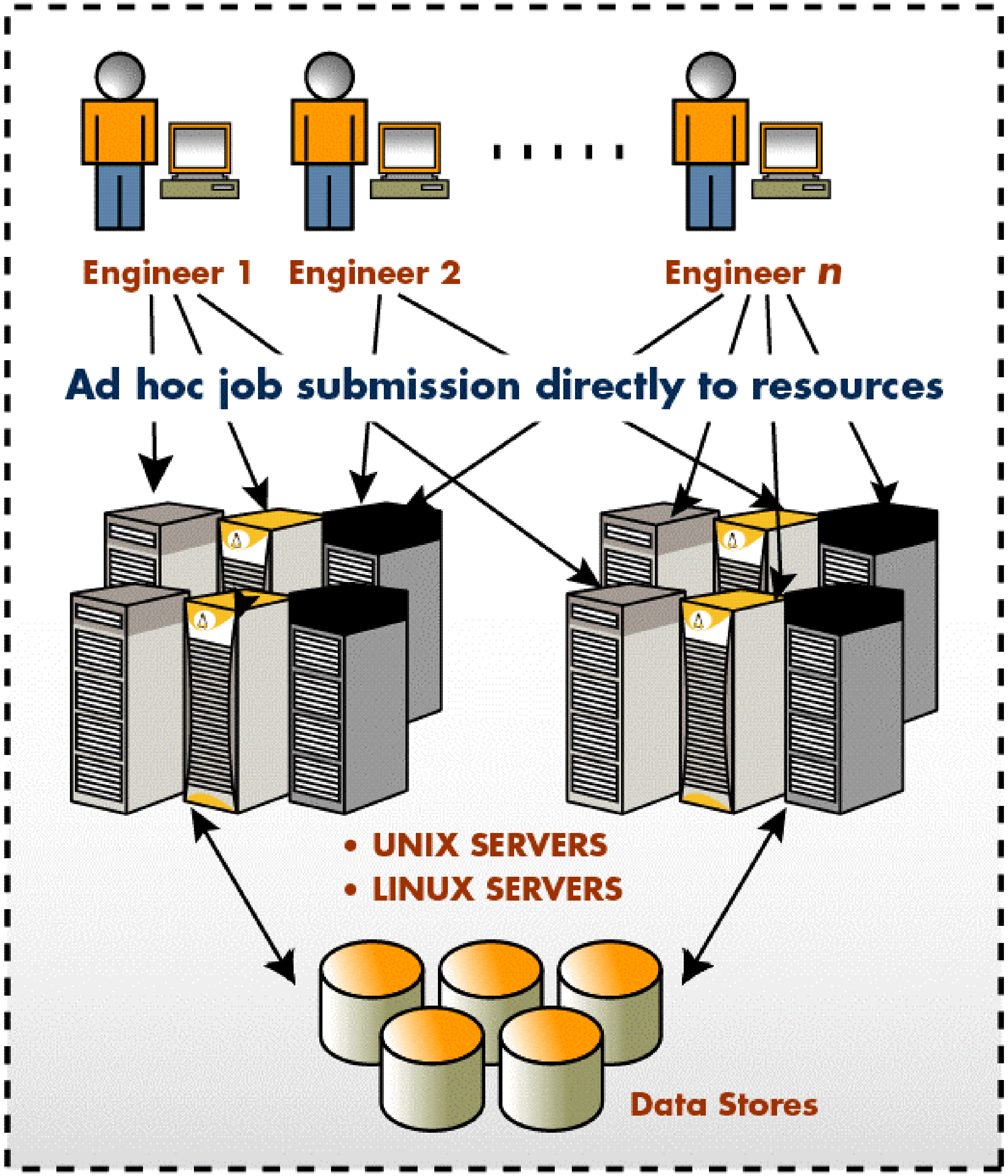
- Topology Awareness
- Task Geometry

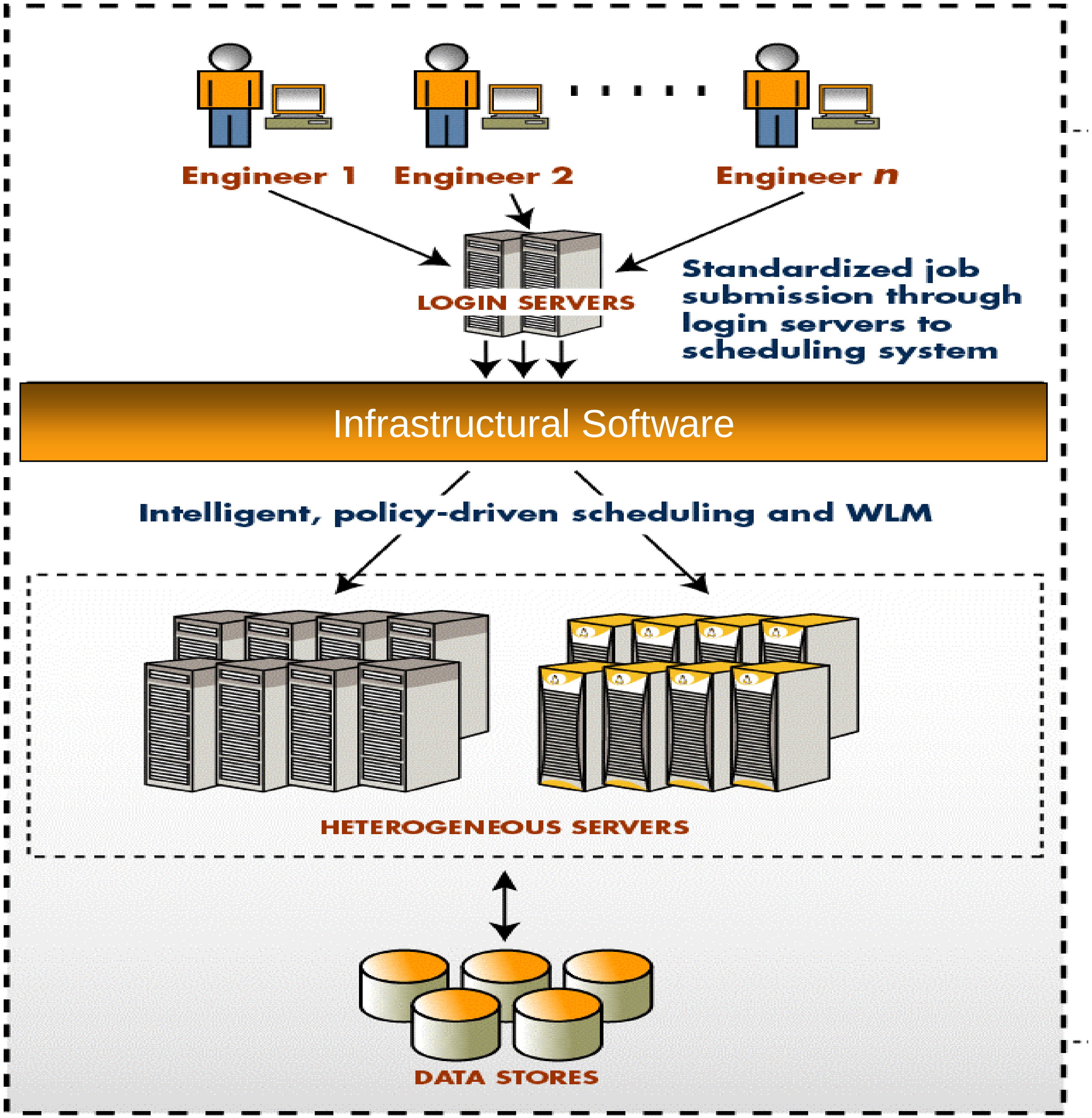
Summary

Introduction

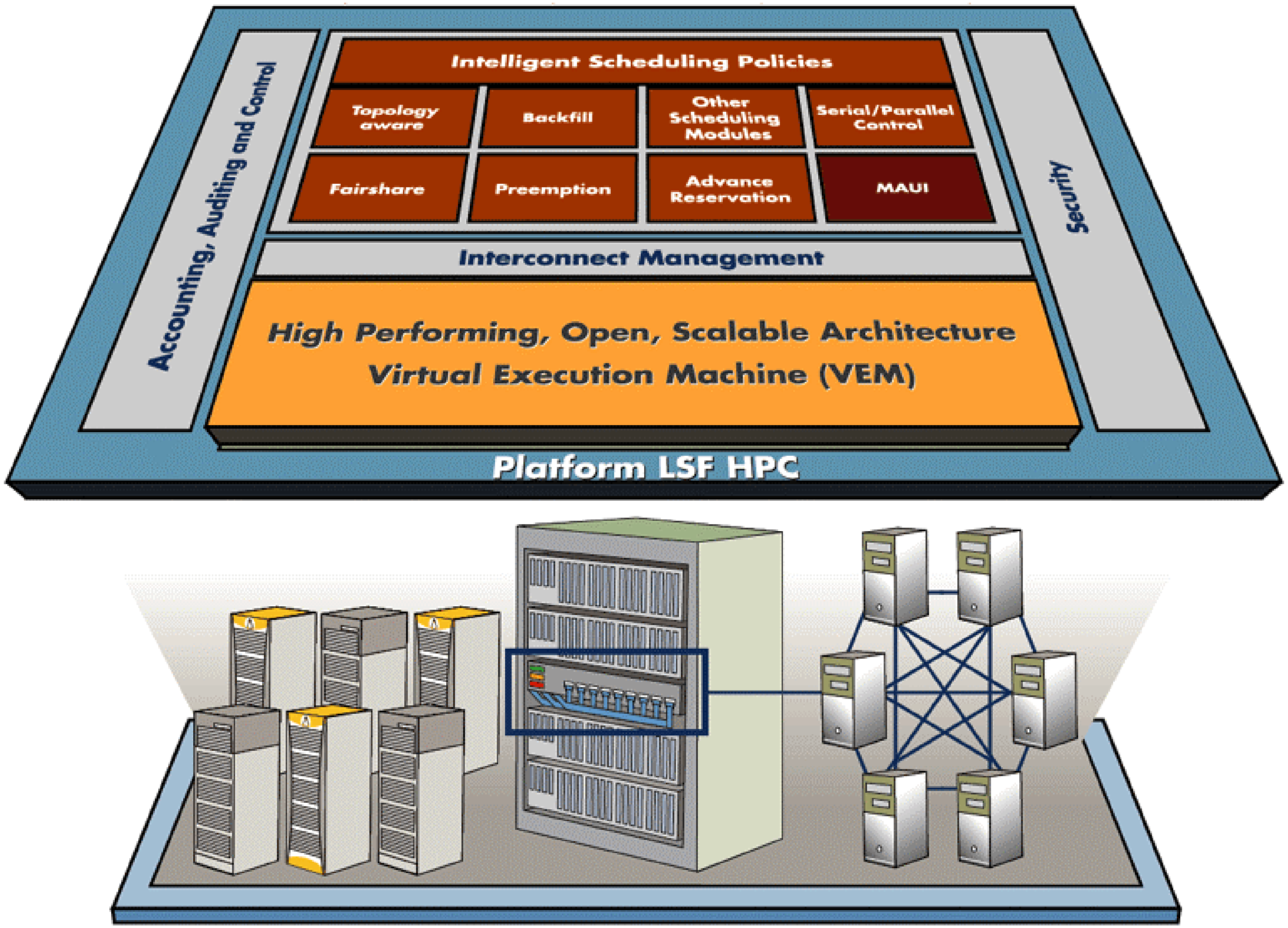

THE GRAND CHALLENGE EQUATIONS
IA-32/64 CPUs
+
Low-latency, high-bandwidth interconnects
+
MPI
Potential for progress on *Grand Challenge* problems
• NEWTON'S EQUATIONS • SCHROEDINGER EQUATION (TIME DEPENDENT) • NAVIER-STOKES EQUATION •
• POISSON EQUATION • HEAT EQUATION • HELMHOLTZ EQUATION • DISCRETE FOURIER TRANSFORM •
• MAXWELL'S EQUATIONS • PARTITION FUNCTION • POPULATION DYNAMICS •
• COMBINED 1ST AND 2ND LAWS OF THERMODYNAMICS • RADIOSITY • RATIONAL B-SPLINE •

Platform™

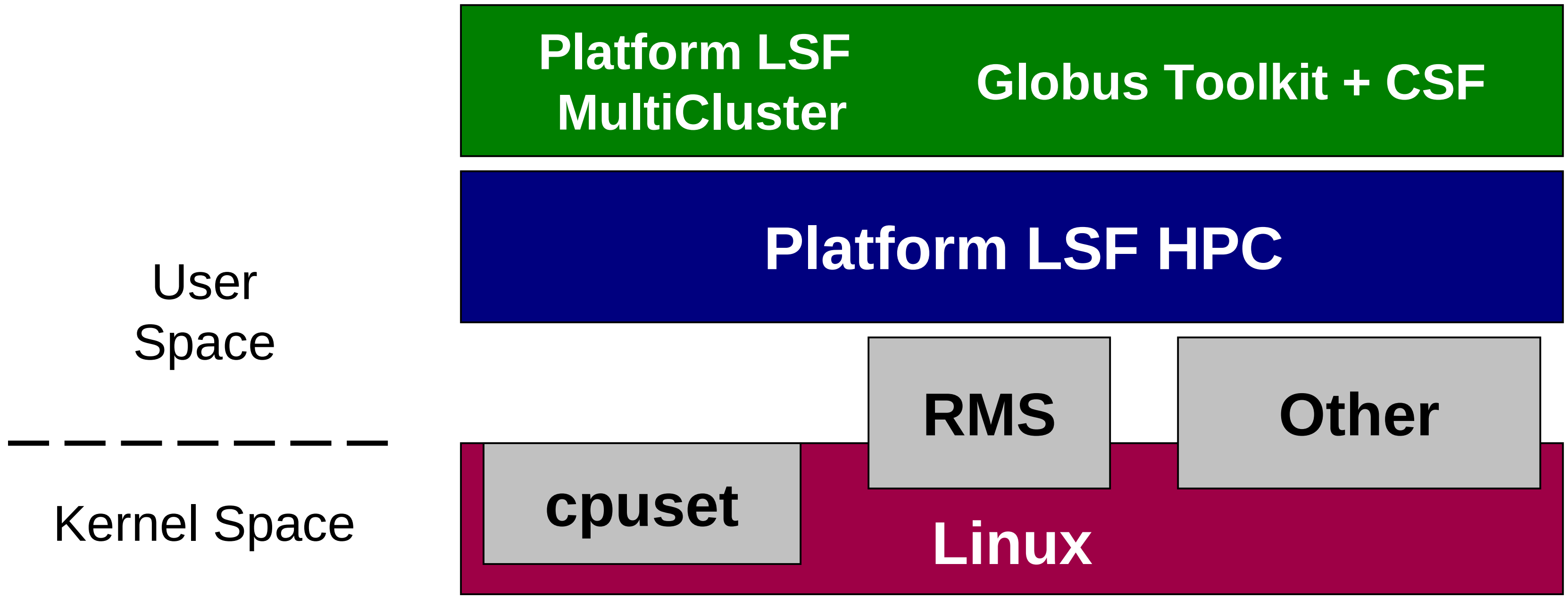

Engineer 1
Engineer 2
Engineer n
Ad hoc job submission directly to resources
• UNIX SERVERS
• LINUX SERVERS
Data Stores


Engineer 1
Engineer 2
Engineer n
LOGIN SERVERS
Standardized job submission through login servers to scheduling system
Infrastructural Software
Intelligent, policy-driven scheduling and WLM
HETEROGENEOUS SERVERS
DATA STORES

Infrastructural Software



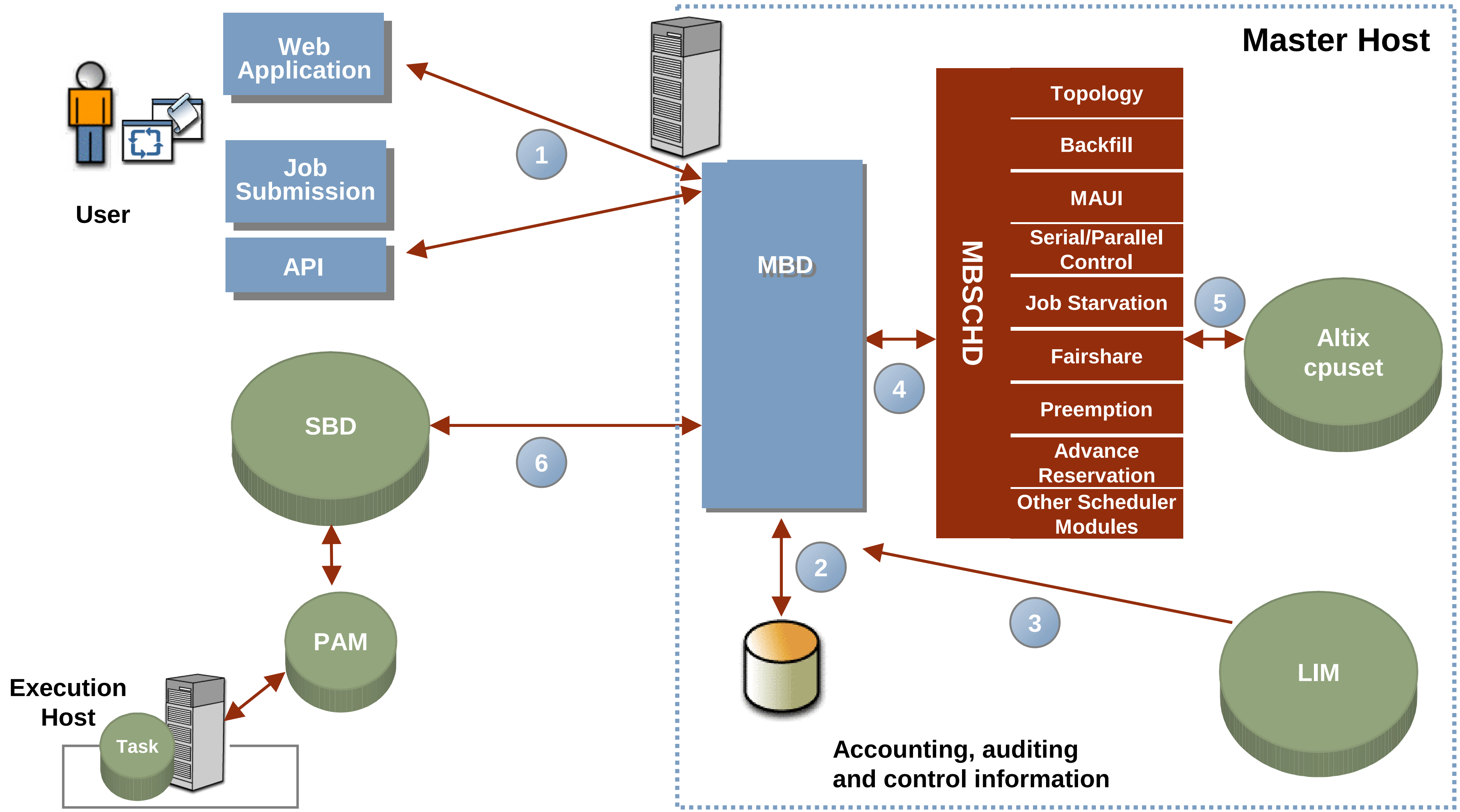
Infrastructural Software



i.e., ∃ an ecosystem of resource management

Topology-Aware Scheduling

Topology-Aware, Bound-Processor Scheduling on Linux/NUMA



Topology-Aware, Bound-Processor Scheduling on Linux/NUMA

```
bsub <bsub options> -n <number of processors> -R
     -ext "SGI_CPUSET[cpuset_options]" pam -mpi -auto_place a.out
```

```
CPUSET_TYPE=static; CPUSET_NAME=<static cpuset name>
```

or

```
CPUSET_TYPE=dynamic; [MAX_RADIUS=<radius>;] [RESUME_OPTION=ORIG_CPUS];
[CPU_LIST=<cpu_ID_list>;] [CPUSET_OPTIONS=<SGI cpuset option>;]
[MAX_CPU_PER_NODE=max_num_cpus]
```

or

```
CPUSET_TYPE=none
```

RSL = Resource Specification Language

Topology-Aware Scheduling on Linux/RMS

```
bsub –n # -ext "alloc_type[; nodes=# | \ ptile=cpus_per_node |
  base=base_node_name] \
 [; rails=# | railmask=bitmask]"
```

where ***alloc_type*** is:

- **RMS_SNODE** – sorted node order as returned by RMS, gaps are allowed in the allocation
- **RMS_MCONT** – contiguous allocation from ‘left to right’ taking into consideration RMS ordering, no gaps are allowed in the allocation
- **RMS_SLOAD** – sorted node order as returned by LSF, gaps are allowed in the allocation

Task Geometry

Platform™

Task Geometry (TG)

Addresses the locality of related tasks

Job submission requires some effort … but the rest is transparent to the end user

- Step 1

  Set LSB_PJL_TASK_GEOMETRY environment variables

- Step 2

  Use bsub -n and -R "span[ptile=]" to make sure LSF selects appropriate hosts to run the job

Task Geometry: Step 1

Set LSB_PJL_TASK_GEOMETRY environment variables

- e.g., LSB_PJL_TASK_GEOMETRY="{(2,5,7)(0,6)(1,3)(4)}"
- This job spawns 8 tasks and span 4 nodes
  - Tasks 2,5, and 7 will run on the same node – the first node
  - Tasks 0 and 6 will run on the same node – the second node
  - Tasks 1 and 3 will run on the same node – the third node
  - Task 4 will run on one node alone – the fourth node

Task Geometry: Step 2

Use bsub -n and -R "span[ptile=]" to make sure LSF selects appropriate hosts to run the job

- This functionality guarantees the geometry but not the host order
- Job submission directives must ensure each host selected by LSF can run any group of tasks specified in LSB_PJL_TASK_GEOMETRY
- Over-book CPUs to achieve this
- bsub -n x -R "span[ptile=y]" –a mpich_gm myjob

  x = y * (the # of nodes)

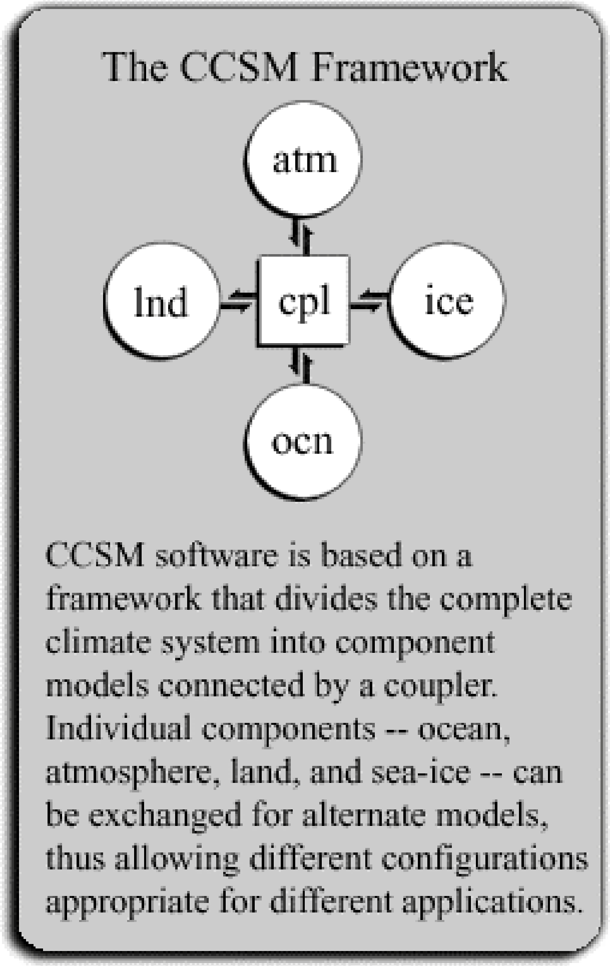
  y = the maxinum # of tasks in one group in LSB_PJL_TASK_GEOMETRY
- e.g., bsub -n 12 -R "span[ptile=3]" –a mpich_gm myjob

TG Example: Community Climate System Model (CCSM)

Multiple Process,
Multiple Data

i.e., MPMD application:

OpenMP + MPI



http://www.ccsm.ucar.edu/models/ccsm3.0

Summary

Platform™

Summary

Infrastructural software *enables* production HPC

Infrastructure needs to be factored into HPC productivity assessments

- Use ‘economic measures’
  - ‘Skill’ in assimilation and reanalysis efforts
- Evaluate time-to-production
- Increase parallel computing ROI
  - Abstract complexity via RSL, programming language or some combination …
- Comply with standards/certifications
  - Linux Standards Base
  - Common Criteria Certification
  - The Open Grid Services Architecture

http://www.darpa.mil/ipto/programs/hpcs

http://www.scimag.com
October 2004 issue dedicated to HPC

Thank you.

Platform™