NERSC Overview

Richard Gerber
NERSC User Services

EETD Seminar
June 22, 2011

U.S. DEPARTMENT OF ENERGY | Office of Science

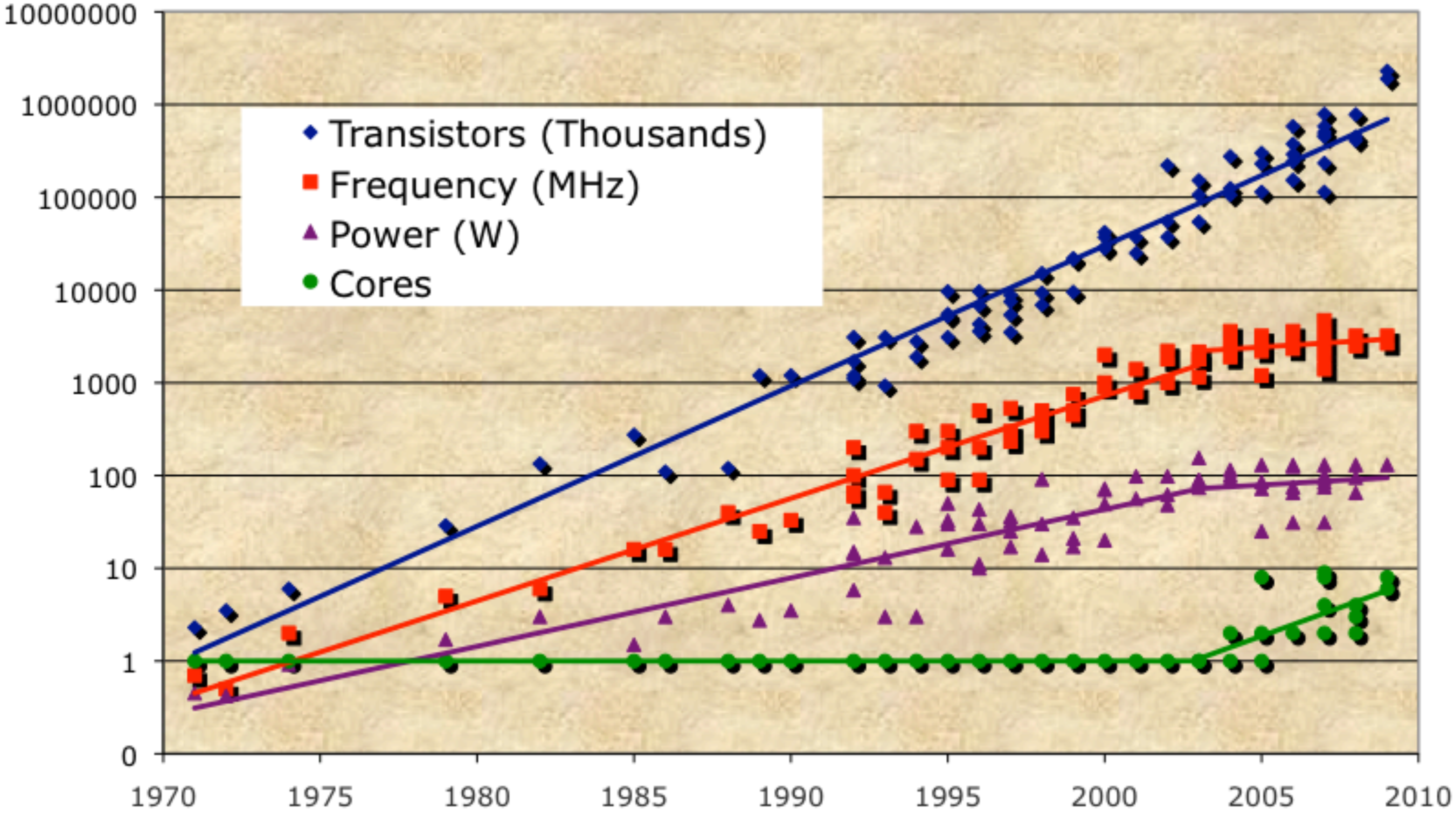
NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

EETD and NERSC

- **NERSC's mission is to provide computing and storage resources for energy-related research and engineering**
- **Broad support for fusion, materials, chemistry research**
  - Hydrogen storage, artificial photosynthesis, solar energy storage, wind farm design, efficient combustion, understanding LED droop
- **Energy efficiency research is an important part of this picture**
  - If there is a place for HPC in EETD, NERSC is capable and eager to help
- **If you remain tied to single-threaded serial computing, you are going to be quickly left behind**
  - Flops are cheap, so use them
  - Cost per Gflop: $1.1 Trillion in 1960, $15 M in 1984, $30,000 in 1997, $600 in 2003, $0.42 in 2007, $0.13 in 2009

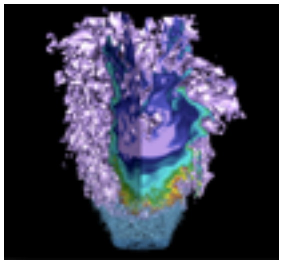
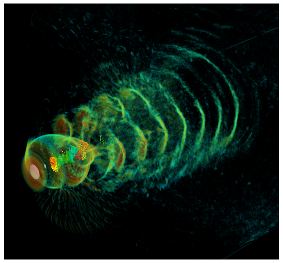
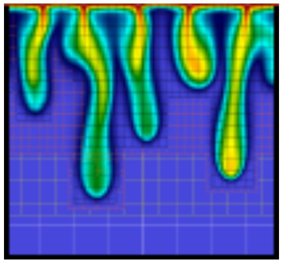
Revolution in Processors

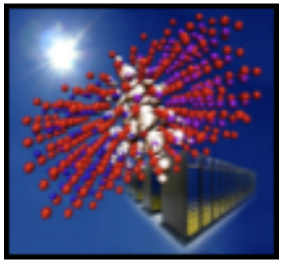
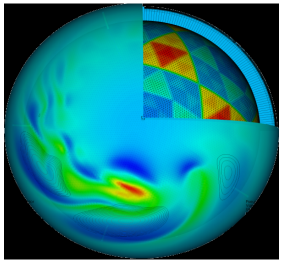


- Chip density is continuing increase ~2x every 2 years
- Clock speed is not
- Number of processor cores may double instead
- Power is under control, no longer growing

Moore's Law reinterpreted

- Number of cores per chip will double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism
- Your take-away:
  - *Future performance increases in computing are going to come from exploiting parallelism in applications*

Why NERSC?

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Expert Services

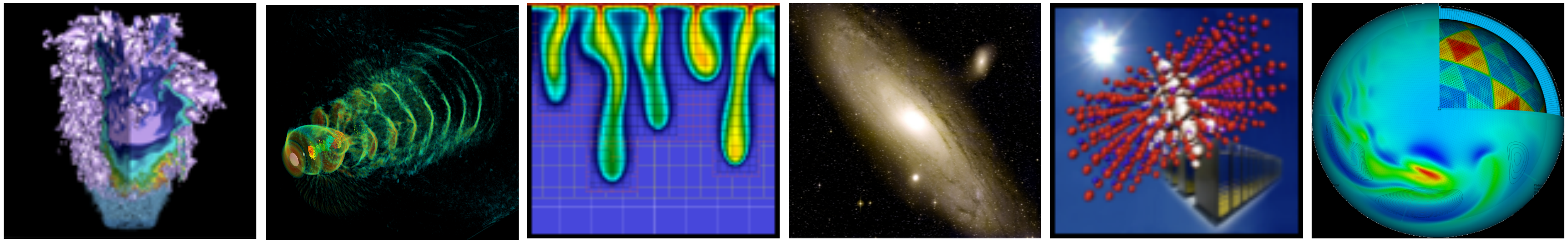
- NERSC's emphasis is on its users
  - Helping scientists and engineers be successful
- User-oriented systems and services
  - Sets NERSC apart from other centers
- Help Desk / Consulting
  - Immediate access to consultants (7 Ph.Ds)
- User group (NUG) has tremendous influence
  - Monthly teleconferences & yearly meetings
- Requirement-gathering workshops with scientists
- Agile response to special requests from users

Advantages (to you) of NERSC

- **Expert services**
  - Updated OS, software
  - NERSC does system administration, 24x7
  - Consulting and advice
  - Emphasis on helping users being successful
  - Reliable systems; auto backups for disaster recovery
- **Free access to vast amounts of computing, software, storage**
  - Play "what if" scenarios
  - Pre-compute a large library of configurations at high resolution/level of detail (or whatever)
  - Add physics and/or more parameters
  - Get compute-intensive results quickly

Disadvantages (to you)

- **NERSC computers are shared**
  - Not generally interactive
  - “Long” turnaround time: hours to days to a week
  - Per user, per system limits on # of active jobs
- **You must fill out an application for new projects**
  - Startup projects are easy to get
- **You have to learn how to use the systems**
  - You are busy already
- **You will probably have to change your code and/or your workflow to take best advantage of NERSC resources**



What is NERSC?

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

NERSC Facility Leads DOE in Scientific Computing Productivity

THE JOURNAL OF CHEMICAL PHYSICS
APPLIED PHYSICS LETTERS
nature
Journal of Fluid Mechanics
nature
WINDS OF CLIMATE CHANGE
Struct
Dynamomics: Protein Folding and Dynamics Go Large-Scale!

NERSC computing for science

- •4000 users, 500 projects
- •From 48 states; 65% from universities
- •Hundreds of users each day
- ***•1500 publications per year***

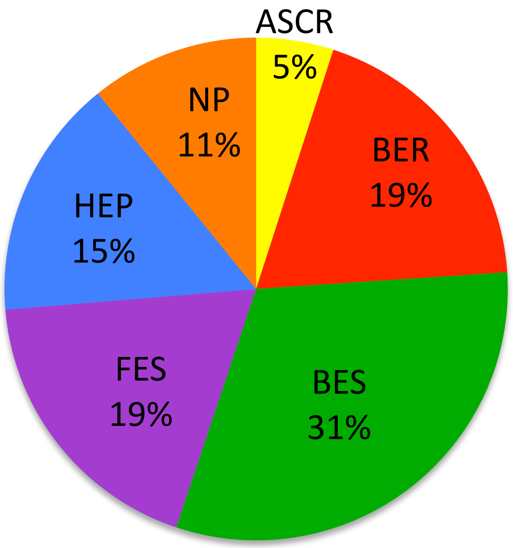
Systems designed for science

- •1.3 PF Petaflop Cray: Hopper
  - 3rd fastest computer in US
  - Fastest open Cray XE6
  - Additional .5 PF in Franklin and smaller clusters

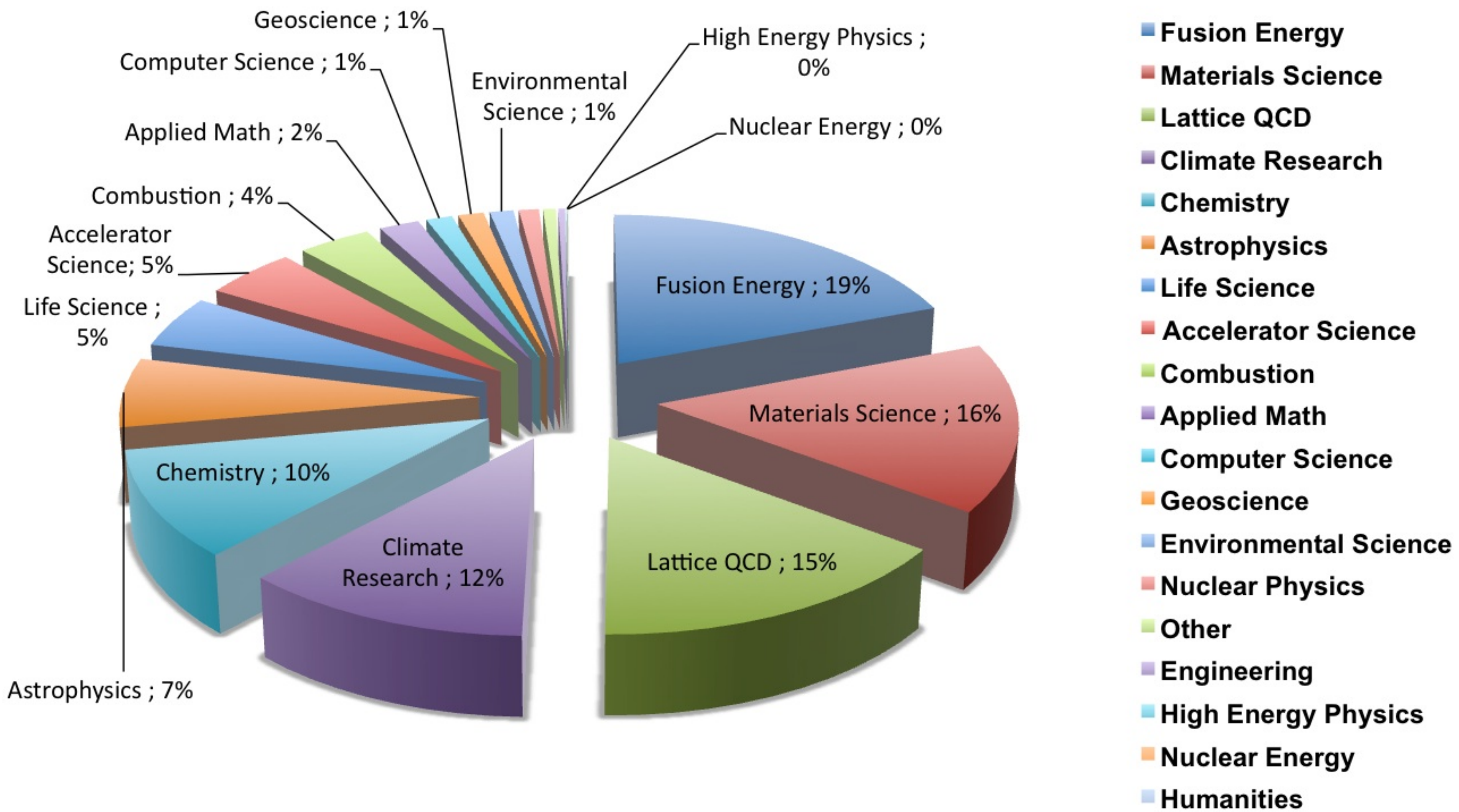
NERSC is the Production Facility for DOE Office of Science

- **NERSC serves a large population**
  - About 4,000 users
  - 400 projects
  - 500 codes
- **Unique resources**
  - Expert consulting and other services
  - High end computing systems
  - High end storage systems
  - Interface to high speed networking
- **Science-driven services**
  - Machines procured competitively using application benchmarks from DOE/SC
  - Allocations controlled by DOE/SC Program Offices to couple with funding decisions

2010 Allocations



NERSC Workload



NERSC 2011 Allocations By Science Area

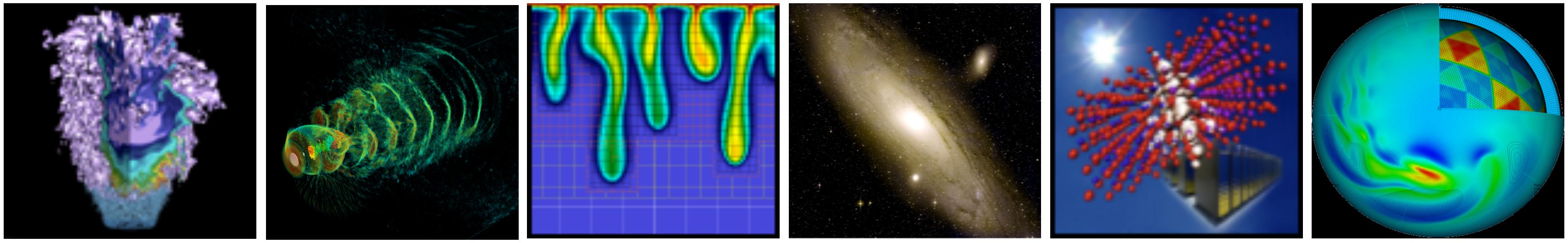
DOE Office of Advanced Scientific Computing Facilities

NERSC at LBNL

- **1000s users,100s projects**
- **Allocations:**
  - 80% DOE program managers
  - 10% ASCR Leadership Computing Challenge
  - 10% NERSC reserve
- **Science includes all of DOE Office of Science**
- **Machines procured competitively**

“Leadership Facilities” at Oak Ridge & Argonne

- **100s users 10s projects**
- **Allocations:**
  - 60% ANL/ORNL managed INCITE process
  - 30% ACSR Leadership Computing Challenge[*]
  - 10% LCF reserve
- **Science limited to largest scale; no commitment to DOE/SC offices**
- **Machines procured through partnerships**



High Performance Computing Systems

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Distributed Memory Systems

- **Most HPC systems are "distributed memory"**
  - Many nodes, each with its own local memory and distinct memory space
  - Nodes communicate over a specialized high-speed, low-latency network
  - SPMD (Single Program Multiple Data) is the most common model
    - Multiple copies of a single program (tasks) execute on different processors, but compute with different data
    - Explicit programming methods (MPI) are used to move data among different tasks

NERSC Systems

Large-Scale Computing Systems

Hopper (NERSC-6): Cray XE6

6,384 compute nodes, 153,216 cores

110 Tflop/s on applications; 1.27 Pflop/s peak

Franklin (NERSC-5): Cray XT4

9,532 compute nodes; 38,128 cores

~25 Tflop/s on applications; 356 Tflop/s peak

Clusters

140 Tflops total

Carver

IBM iDataplex cluster

PDSF (HEP/NP)

~1K core cluster

Magellan Cloud testbed

IBM iDataplex cluster

GenePool (JGI)

~5K core cluster

NERSC Global File system (NGF)

Uses IBM's GPFS

1.5 PB capacity

5.5 GB/s of bandwidth

HPSS Archival Storage

40 PB capacity

4 Tape libraries

150 TB disk cache

Analytics

Euclid

512 GB shared mem

Dirac

GPU testbed

48 nodes

Major Compute Systems

- **Hopper**
  - Hopper is NERSC's flagship computer for running high-performance parallel scientific codes.
- **Franklin**
  - Franklin, an earlier-generation Cray computer, augments the Hopper system.
- **Carver**
  - Carver provides a generic full Linux environment for codes that need operating system features that are not available on the Cray systems or don't demand massive parallelism.

Hopper - Cray XE6

- 1.2 GB memory / core (2.5 GB / core on "fat" nodes) for applications
- /scratch disk quota of 5 TB
- 2 PB of /scratch disk
- Choice of full Linux operating system or optimized Linux OS (Cray Linux)
- PGI, Cray, Pathscale, GNU compilers

- 153,408 cores, 6,392 nodes
- "Gemini” interconnect
- 2 12-core AMD 'MagnyCours' 2.1 GHz processors per node
- 24 processor cores per node
- 32 GB of memory per node (384 "fat" nodes with 64 GB)
- 216 TB of aggregate memory

Use Hopper for your biggest, most computationally challenging problems.

Franklin - Cray XT4

38,288 compute cores

9,572 compute nodes

One quad-core AMD 2.3 GHz Opteron processors (Budapest) per node

4 processor cores per node

8 GB of memory per node

78 TB of aggregate memory

1.8 GB memory / core for applications

/scratch disk default quota of 750 GB

Light-weight Cray Linux operating system

No runtime dynamic, shared-object libs

PGI, Cray, Pathscale, GNU compilers

Use Franklin for all your computing jobs, except those that need a full Linux operating system.

Carver - IBM iDataPlex

3,200 compute cores

400 compute nodes

2 quad-core Intel Nehalem 2.67 GHz processors per node

8 processor cores per node

24 GB of memory per node (48 GB on 80 "fat" nodes)

2.5 GB / core for applications (5.5 GB / core on "fat" nodes)

InfiniBand 4X QDR

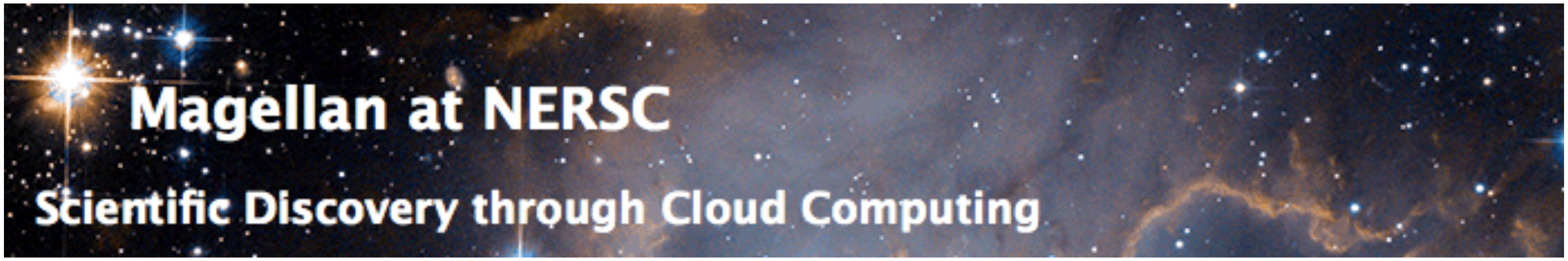
NERSC global /scratch directory quota of 20 TB

Full Linux operating system

PGI, GNU, Intel compilers

Use Carver for jobs that use up to 512 cores, need a fast CPU, need a standard Linux configuration, or need up to 48 GB of memory on a node.

Magellan - IBM IDataPlex



Dedicated to HPC Cloud Computing research

4,480 compute cores

560 compute nodes

Two quad-core Intel Nehalem 2.67 GHz processors per node

8 processor cores per node

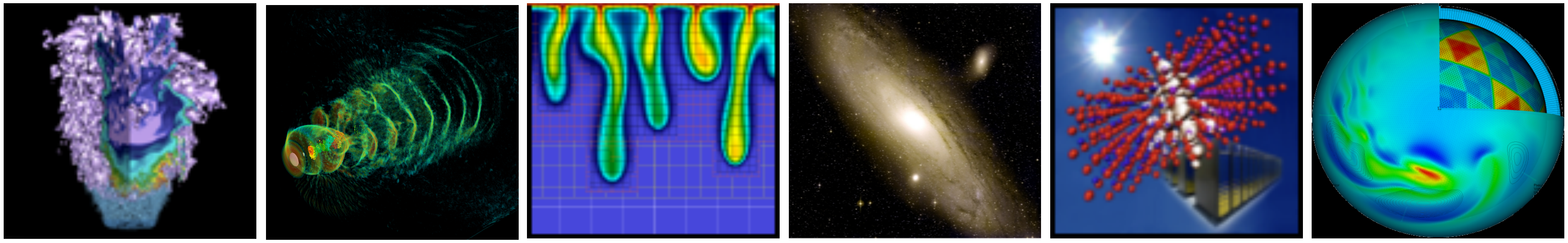
24 GB of memory per node (48 GB on 160 "fat" nodes)

2.5 GB / core for applications (5.5 GB / core on "fat" nodes)

NERSC global /scratch directory quota of 20 TB

Full Linux operating system

PGI, GNU, Intel compilers



Using NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

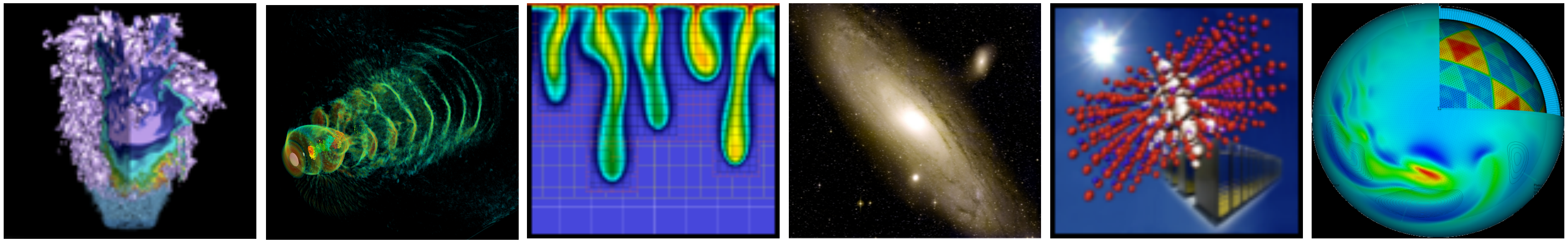
BERKELEY LAB Lawrence Berkeley National Laboratory

Allocations

- You must have an allocation of time to run jobs at NERSC (be a member of a "repo")
- Project PIs apply through the ERCAP process
- Computer time and storage allocations are awarded by DOE Program Offices
- Most allocations are awarded in the fall
  - Allocation year starts in January
  - 2011: Additional awards made for May 1 start of Hopper production service
  - Small startup allocations are awarded throughout the year
  - Additional time available through NISE and ALCC

Jobs at NERSC

- Most jobs are parallel, using 10s to 100,000+ cores.
- Mostly run as batch scripts; limited interactive access
- Many use custom codes; others use pre-installed applications
- Typically run a few hours, up to 48. Longer runs can be accommodated if needed and logistically possible.
- Many jobs “package” lower concurrency runs into one job
  - Even many “serial jobs”
  - Load balance may be an issue



System Architecture

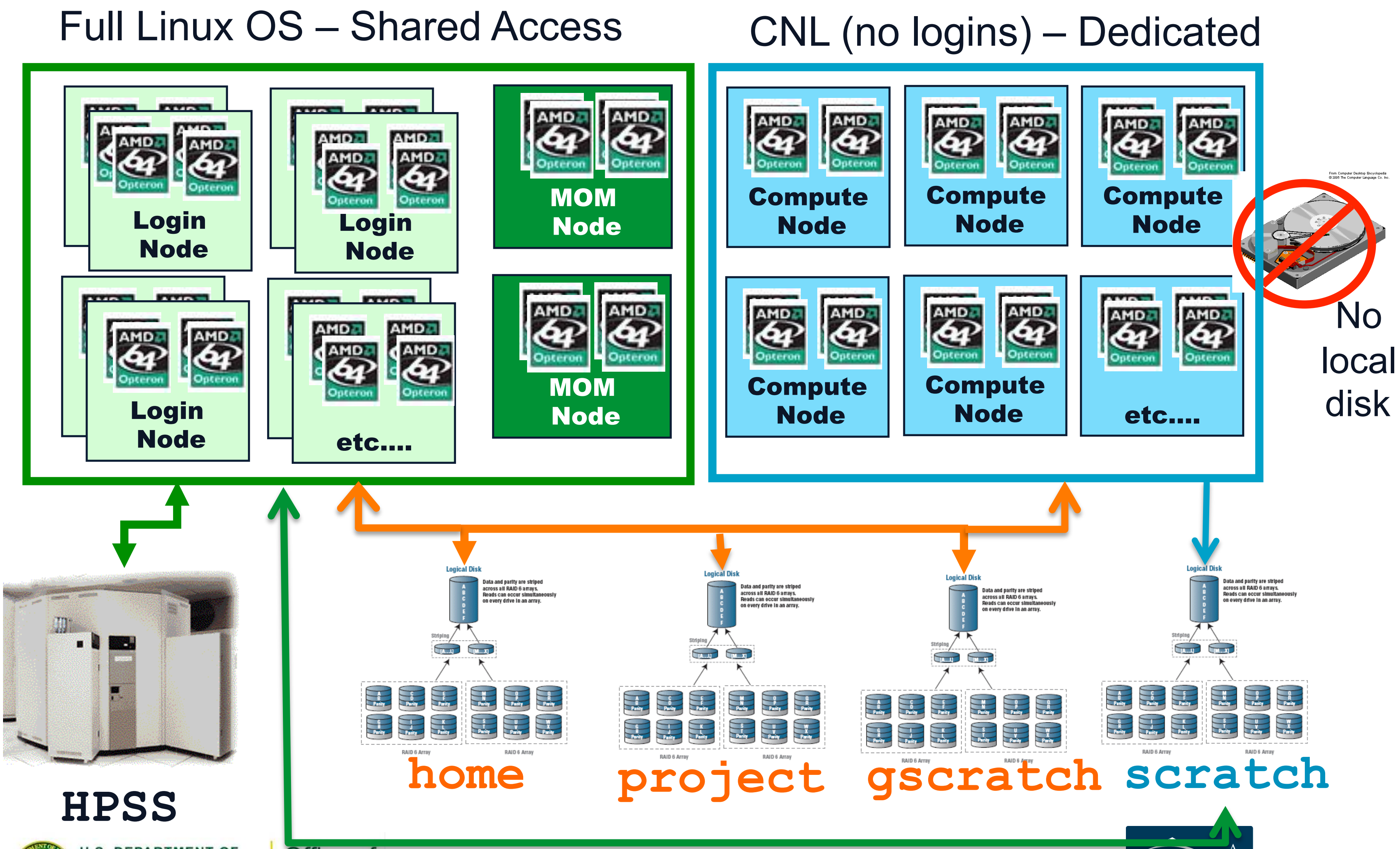
HPC Node

- **A “node” is a (physical) collection of CPUs, memory, and interfaces to other nodes and devices.**
  - Single memory address space
  - Memory access “on-node” is significantly faster than “off-node” memory access
  - Often called an “SMP node” for “Shared Memory Processing”
    - Not necessarily “symmetric” memory access as in “Symmetric Multi-Processing”

Login Nodes and Compute Nodes

- Each supercomputer has 3 types of nodes that you will use directly
  - Login nodes
  - Compute nodes
  - “MOM” nodes
- Login nodes
  - Edit files, compile codes, run UNIX commands
  - Submit batch jobs
  - Run short, small utilities and applications
- Compute nodes
  - Execute your application; dedicated to your job
  - No direct login access
- “MOM” nodes
  - Execute your batch script commands
  - Carver: “head” compute node; Cray: shared “service” node

Cray Systems



Full Linux OS – Shared Access

CNL (no logins) – Dedicated

No local disk

HPSS

home

project

gscratch

scratch

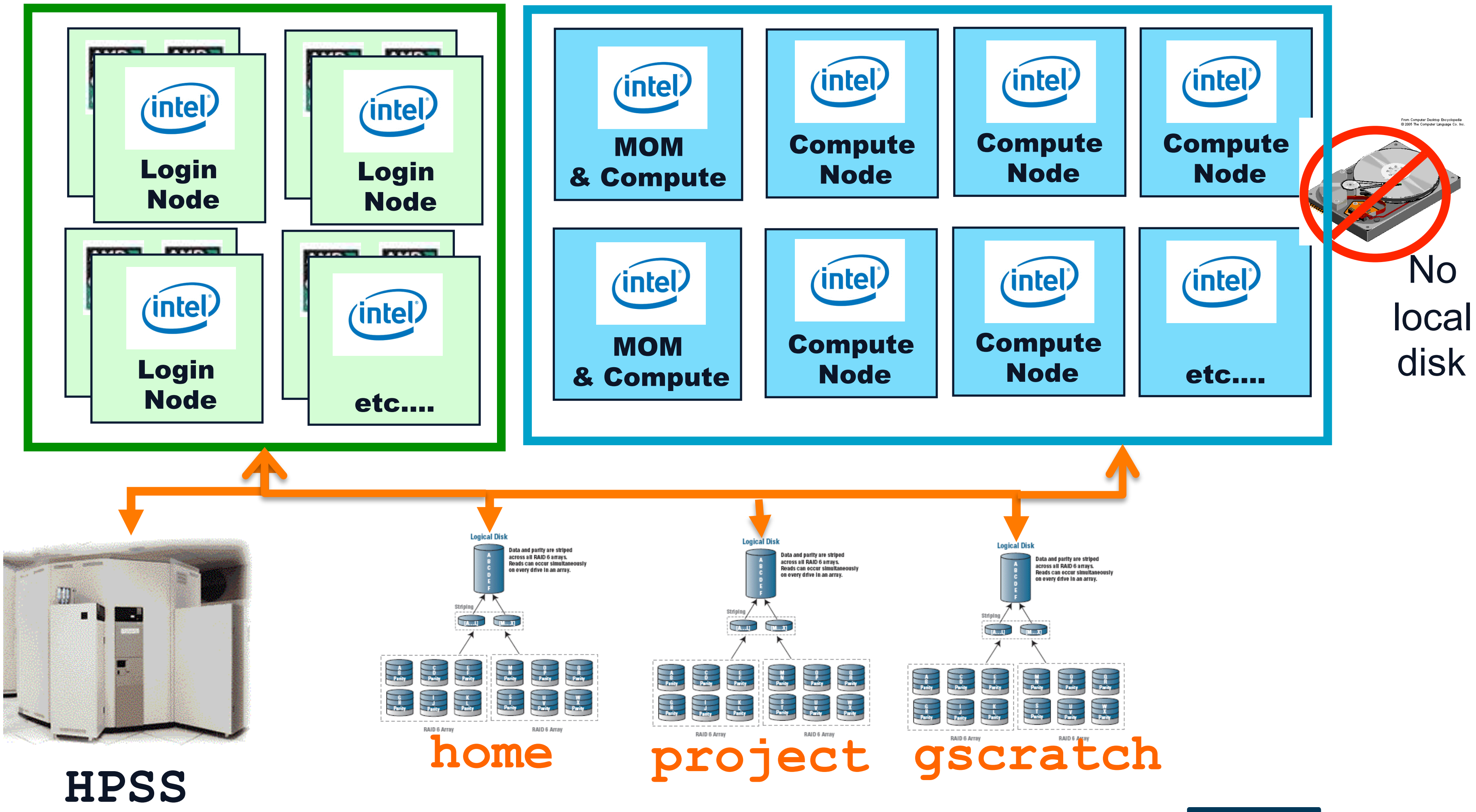
U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Carver / Magellan / Dirac

Full Linux OS – Shared

Full Linux (no logins) – Dedicated



Login Node

Login Node

Login Node

etc....

MOM & Compute

Compute Node

Compute Node

Compute Node

MOM & Compute

Compute Node

Compute Node

etc....

No local disk

HPSS

home

project

gscratch

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

Parallel Models

- **SPMD**
  - Single Program, Multiple Data
  - Most common way to run codes at NERSC
  - *N* copies of your program/application/ code execute at the same time
  - Each instance does calculations involving a portion of a large data set
  - Temperature in a house grid example

SPMD Execution Model

- Each code instance executes on one compute “core”
- Each instance holds its data in local private memory
- If other instances need data or the results of calculations owned by another instance, the two must “pass the data (message)” to where it is needed
- Library of functions (MPI)

Shared-Memory Threaded Parallelism

- A single executable creates multiple process threads of execution that all have access to a shared pool of memory
- Typically one thread per compute core
- Number of threads limited to the number of cores on a node
- Computational work distributed to threads by programmer or maybe compiler

Hybrid Model

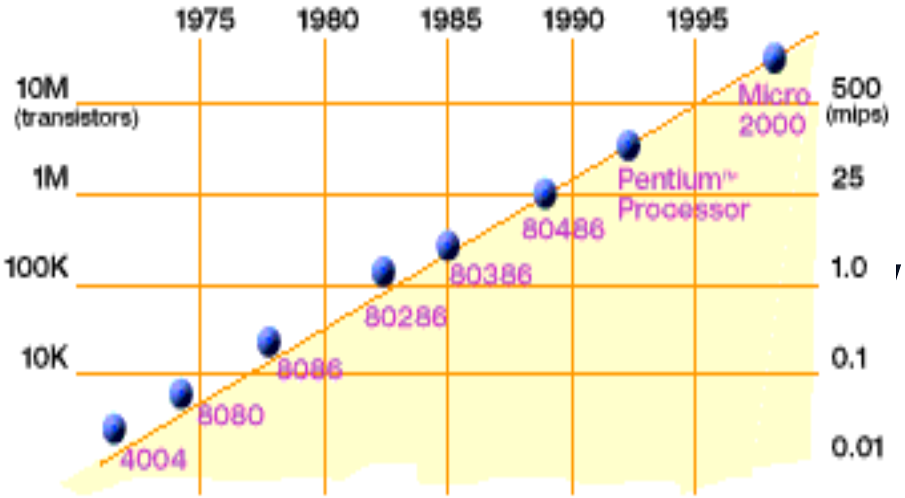
- **Run multiple instances**
  - Communicate via MPI function calls
- **Multiple threads per instance**
  - Often 1 MPI task and cores_per_node threads per task
  - Architecture may dictate best ratio of threads/ MPI task
- **Now two layers of parallelism**

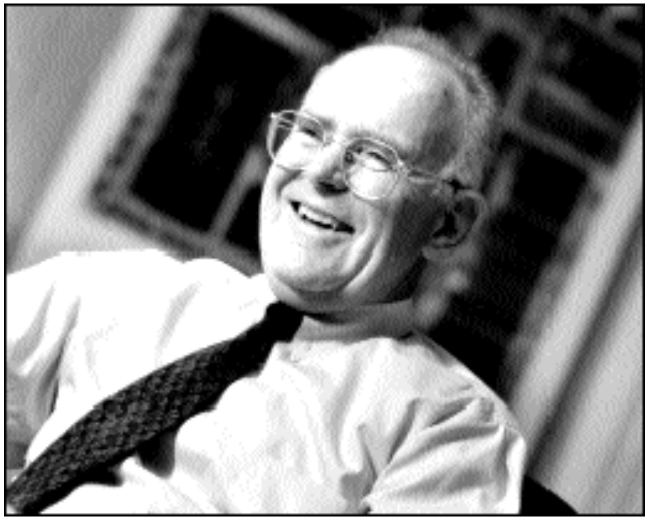
GPUs

- **NVIDIA Tesla “gaming chip”**
  - 515 Flops
  - vs. ~ 50 Gflops on AMD hex-core on Hopper
  - 448 lightweight cores w/ private memory
  - Currently implemented as a “coprocessor” on a PCI card
- **Program with CUDA**
- **“Disaster” from a programming perspective**
  - Third layer of parallelism: MPI+Threads+CUDA
  - You do the memory management between GPU and CPU
  - A new API to learn
  - Few and immature programming tools
  - Future of CUDA and “coprocessor” paradigm uncertain at best
- **Not all codes will benefit from GPU acceleration**
- **There is hope for tighter integration of CPU and GPU and better programming models**

Why Do You Care About Parallelism?

Moore's Law





2X transistors/Chip Every 1.5 years

Called "Moore's Law"

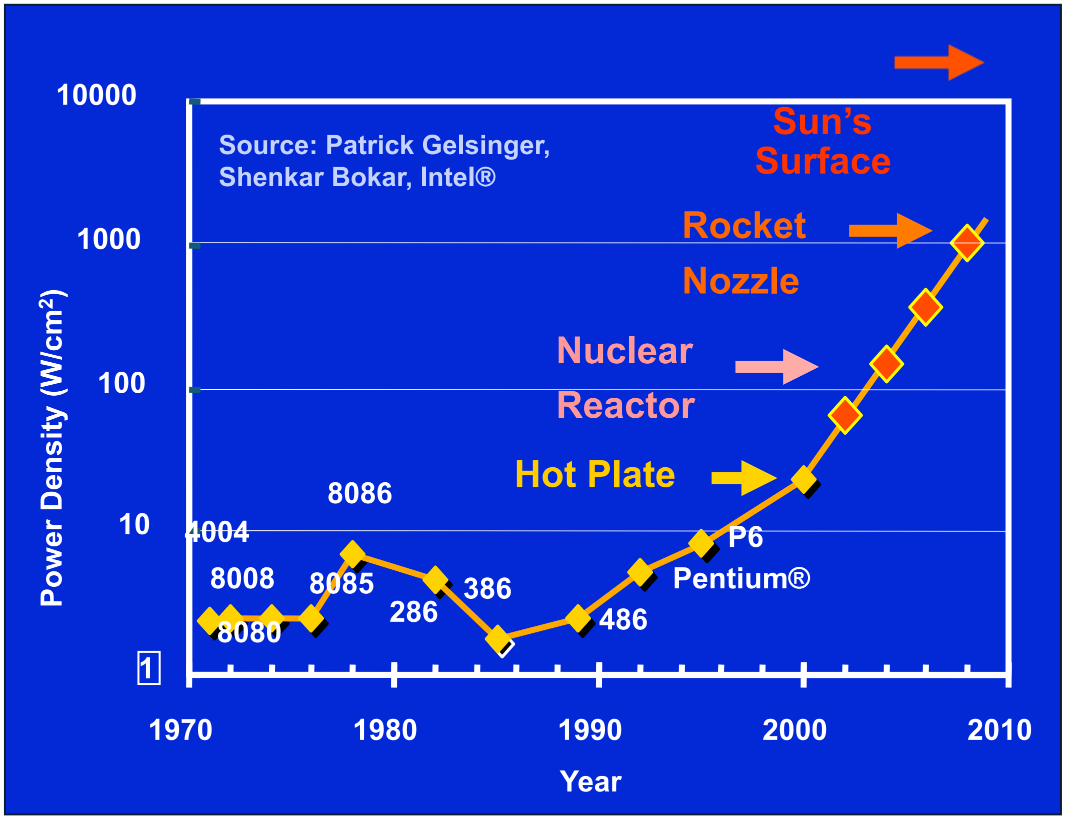
Microprocessors have become smaller, denser, and more powerful.

Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

Slide source: Jack Dongarra

Power Density Limits Serial Performance

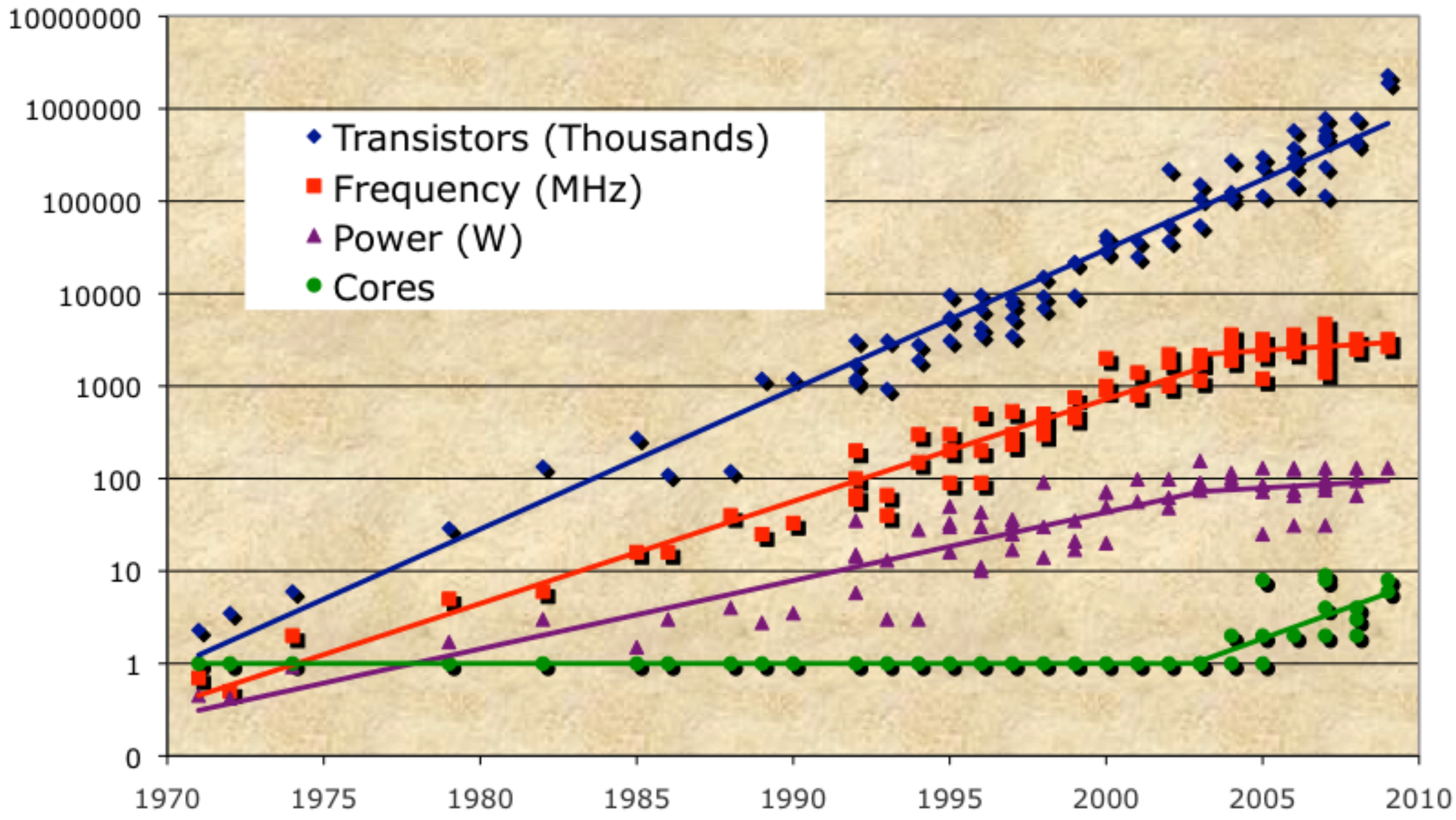
- Concurrent systems are more power efficient
  - Dynamic power is proportional to V^2fC
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly
  - Save power by lowering clock speed



Source: Patrick Gelsinger, Shenkar Bokar, Intel®

- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- More transistors, but not faster serial processors

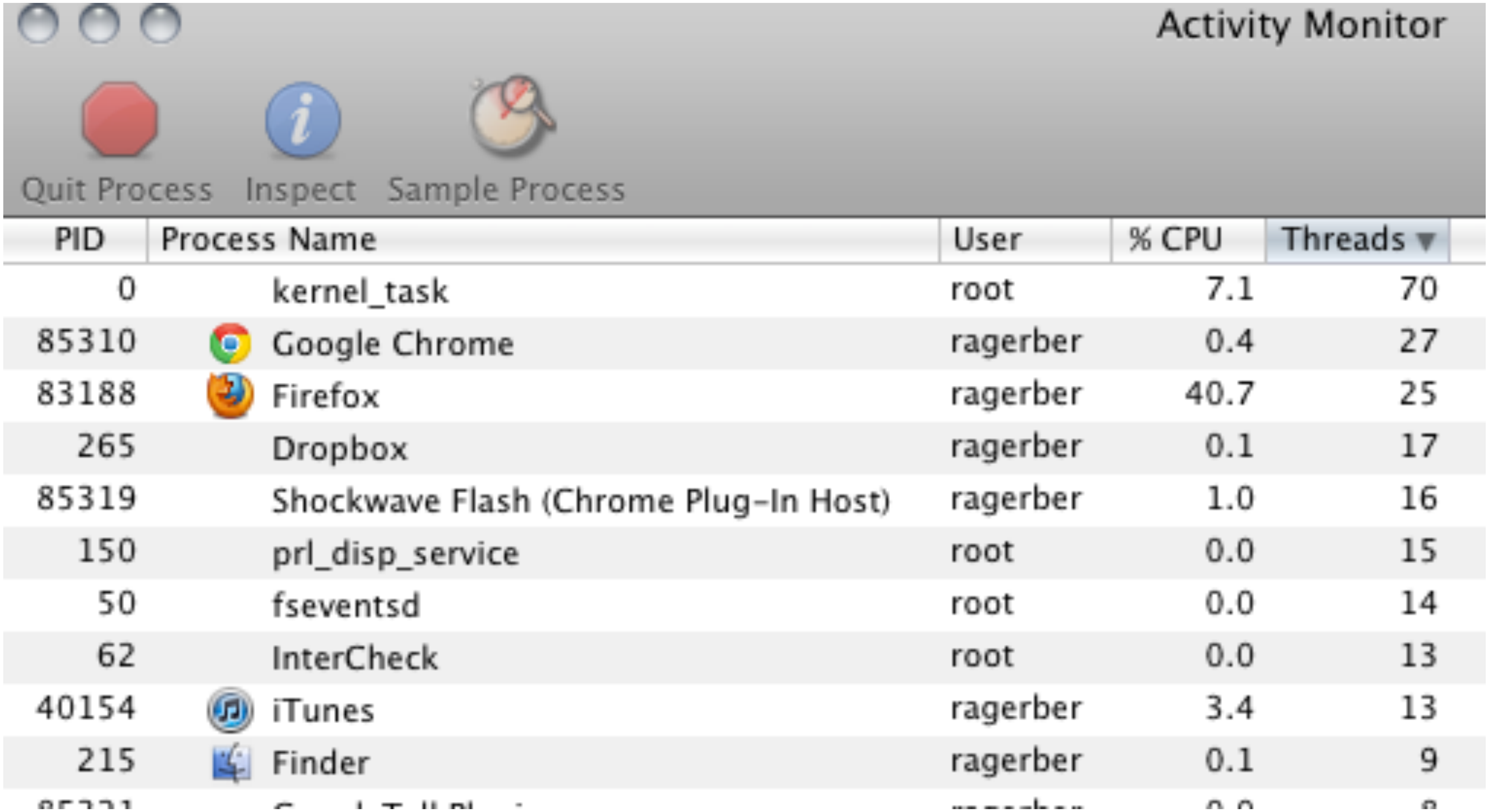
Revolution in Processors



- Chip density is continuing increase ~2x every 2 years
- Clock speed is not
- Number of processor cores may double instead
- Power is under control, no longer growing

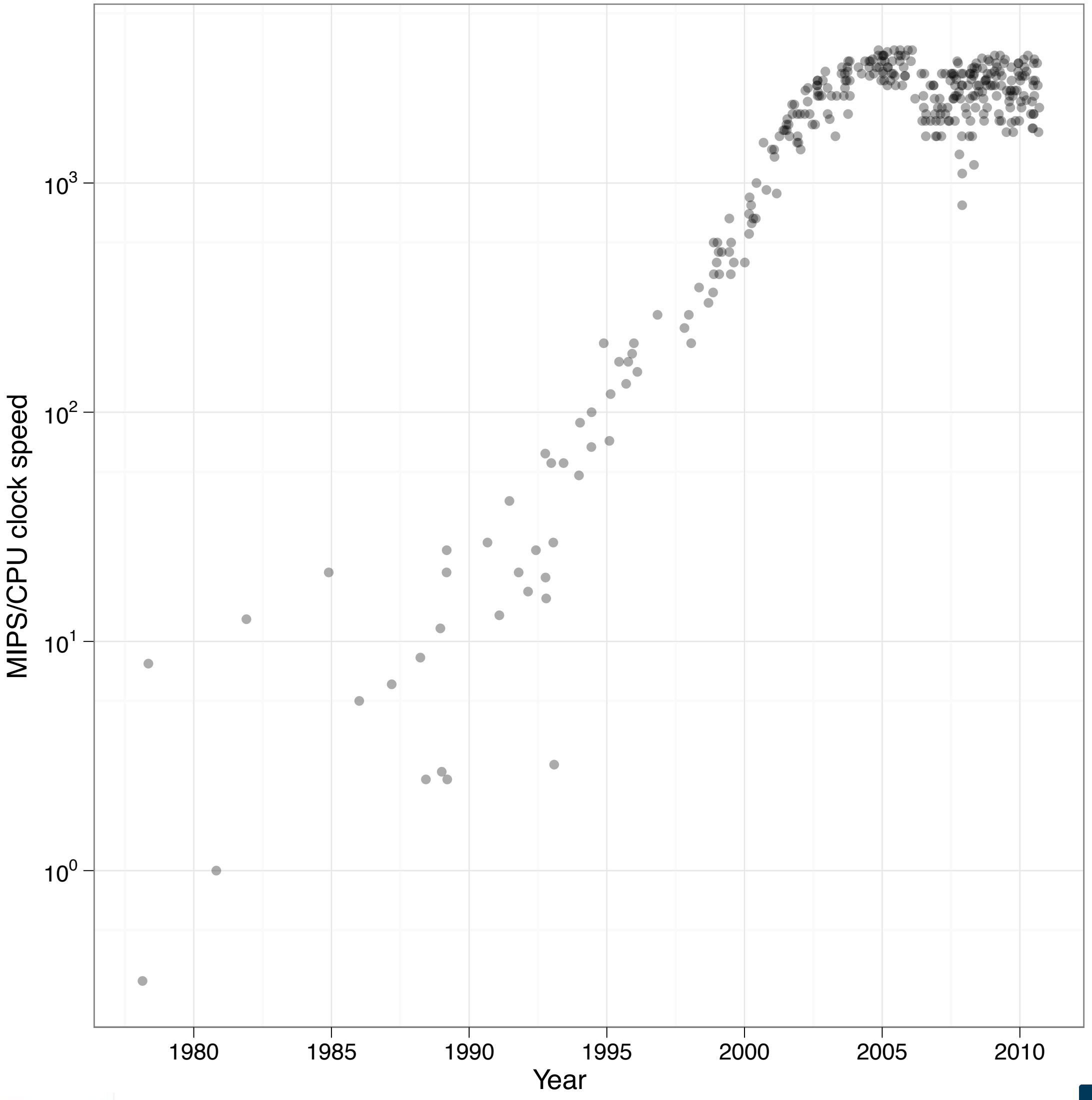
Moore’s Law reinterpreted

- Number of cores per chip will double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism
- Your take-away:
  - ***Future performance increases in computing are going to come from exploiting parallelism in applications***



PID	Process Name	User	% CPU	Threads ▼
0	kernel_task	root	7.1	70
85310	Google Chrome	ragerber	0.4	27
83188	Firefox	ragerber	40.7	25
265	Dropbox	ragerber	0.1	17
85319	Shockwave Flash (Chrome Plug-In Host)	ragerber	1.0	16
150	prl_disp_service	root	0.0	15
50	fseventsd	root	0.0	14
62	InterCheck	root	0.0	13
40154	iTunes	ragerber	3.4	13
215	Finder	ragerber	0.1	9

CPU Clock Frequency



NERSC

National Energy Research
Scientific Computing Center