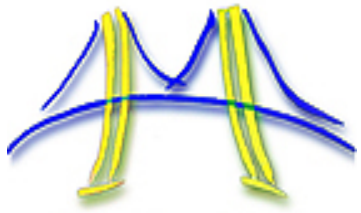
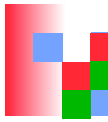
PARLab Parallel Boot Camp

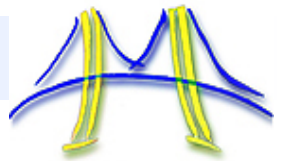


Short Course on Parallel Computing
August 16-18, 2010
parlab.eecs.berkeley.edu/2010bootcamp

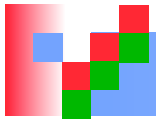
Jim Demmel
EECS and Mathematics
University of California, Berkeley



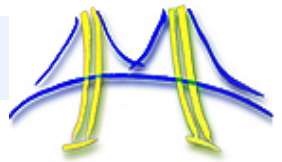
Outline



- Motivation and Goals
- Background
  - ParLab, a research center in Parallel Computing
  - The Designated Emphasis (DE) in Computational Science and Engineering (CSE)
  - CSE at Lawrence Berkeley National Lab
- Schedule and Instructors
- Logistics
- The Audience

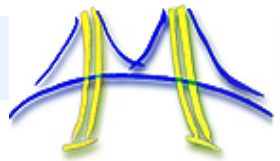


Motivation (1/2)

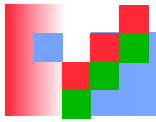


- Parallel Computing is becoming ubiquitous
  - Only way forward for computing industry (unless you don't care if your programs never run faster than in 2008)
  - Unfortunately, parallel programming is (still) harder than sequential programming
  - Until better (easier) programming tools come along, we need to train everyone in parallel programming
- So welcome!

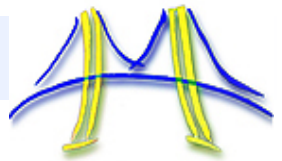
Motivation (2/2)



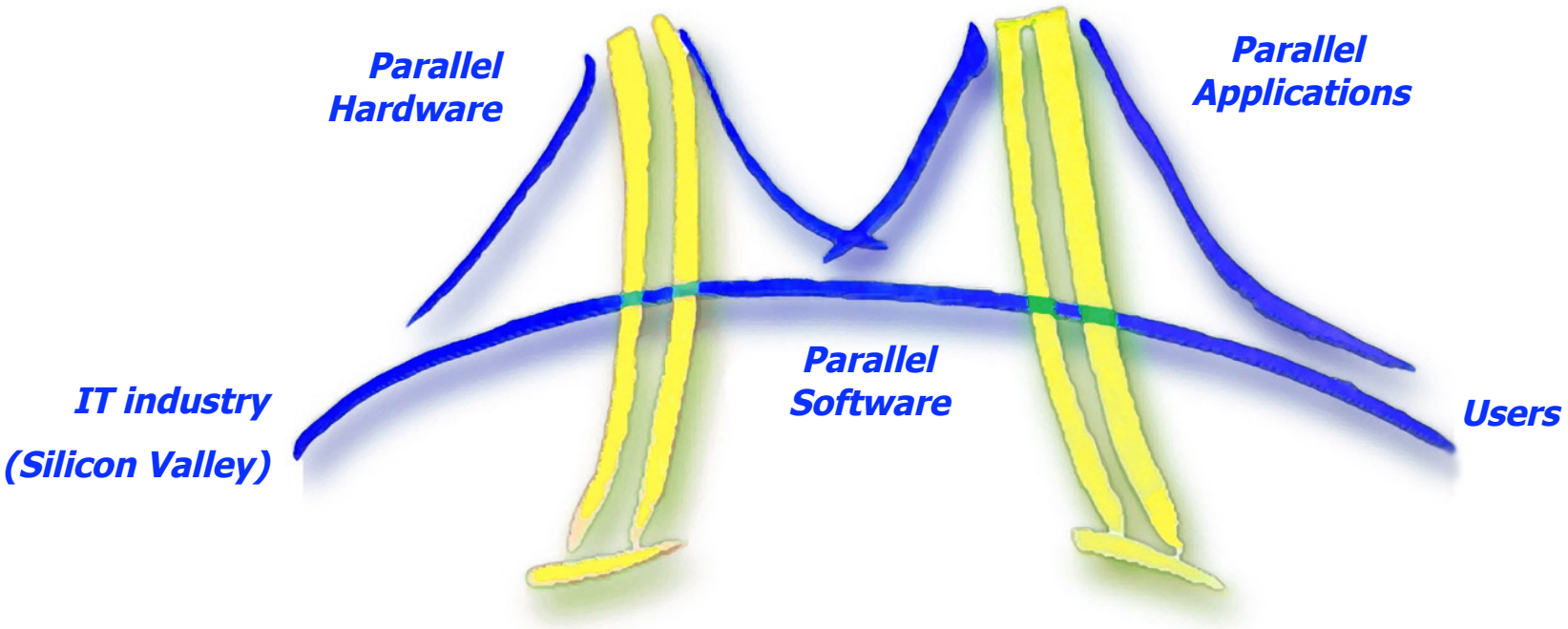
- Recent events at UCB will provide support for many new activities to develop and use parallel computing
  - ParLab established – parlab.eecs.berkeley.edu
    - » Research center about "Multicore Revolution"
  - Designated Emphasis in Computational Science and Engineering (CSE) established – cse.berkeley.edu
    - » New graduate program with 117 faculty from 22 departments



Short Course Goals



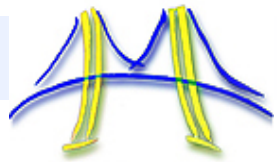
- Teach the basics about parallelism
  - How to program, including hands-on lab
- Tools you can use now (simple and advanced)
- Tools we hope to build, and ongoing research



Berkeley ParLab Project

Krste Asanovic, Ras Bodik, Jim Demmel, Tony Keaveny, Kurt Keutzer, John Kubiatowicz, Edward Lee, Nelson Morgan, Dave Patterson, Koushik Sen, John Wawrzynek, David Wessel, and Kathy Yelick

7 Dwarfs of High Performance Computing (HPC)



Structured Grid

Dense Matrix

Sparse Matrix

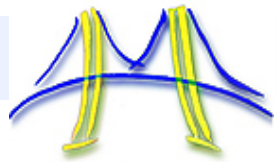
Spectral (FFT)

Particle Methods

Unstructured Grid

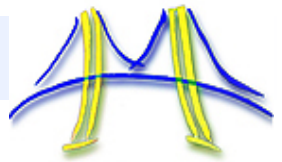
Monte Carlo

7 Dwarfs – Are they enough?

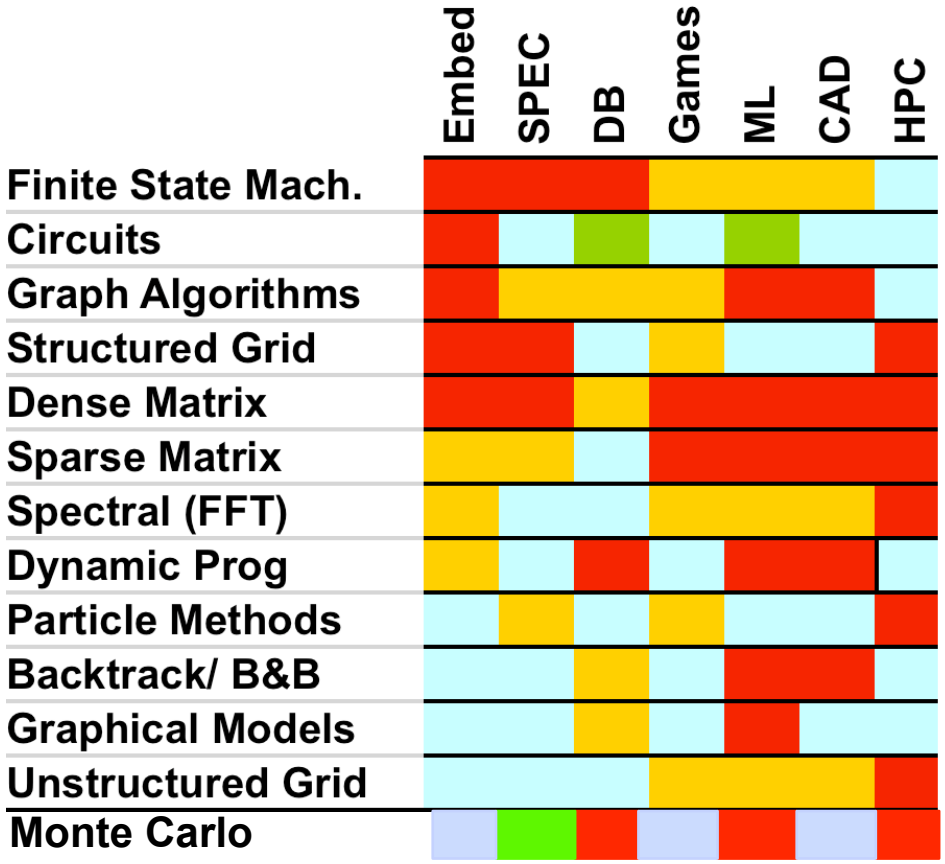


	Embed	SPEC	DB	Games	ML	CAD	HPC
Structured Grid							
Dense Matrix							
Sparse Matrix							
Spectral (FFT)							
Particle Methods							
Unstructured Grid							
Monte Carlo							

13 Motifs (nee "Dwarf") of Parallel Computing



Popularity: (Red Hot / Blue Cool)

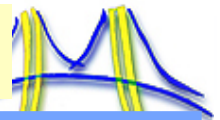


Motifs in ParLab Applications
(Red Hot / Blue Cool)

		Embed	SPEC	DB	Games	ML	CAD	HPC	Health	Image	Speech	Music	Browser
1	Finite State Mach.												
2	Circuits												
3	Graph Algorithms												
4	Structured Grid												
5	Dense Matrix												
6	Sparse Matrix												
7	Spectral (FFT)												
8	Dynamic Prog												
9	Particle Methods												
10	Backtrack/ B&B												
11	Graphical Models												
12	Unstructured Grid												

- What happened to Monte Carlo?

Programming Pattern Language 1.0 Keutzer& Mattson



Applications

Productivity Layer

Choose your high level structure – what is the structure of my application? Guided expansion

Pipe-and-filter

Agent and Repository

Process Control

Event based, implicit invocation

Choose your high level architecture - Guided decomposition

Task Decomposition ↔ Data Decomposition

Group Tasks Order groups data sharing data access

Model-view controller

Iterator

Map reduce

Layered systems

Arbitrary Static Task Graph

Identify the key computational patterns – what are my key computations? Guided instantiation

Graph Algorithms

Dynamic Programming

Dense Linear Algebra

Sparse Linear Algebra

Unstructured Grids

Structured Grids

Graphical models

Finite state machines

Backtrack Branch and Bound

N-Body methods

Circuits

Spectral Methods

Refine the structure - what concurrent approach do I use? Guided re-organization

Event Based

Divide and Conquer

Data Parallelism

Geometric Decomposition

Pipeline

Discrete Event

Task Parallelism

Graph algorithms

Digital Circuits

Efficiency Layer

Utilize Supporting Structures – how do I implement my concurrency? Guided mapping

Fork/Join

CSP

Distributed Array Shared-Data

Shared Queue

Shared Hash Table

Master/worker

Loop Parallelism

BSP

Implementation methods – what are the building blocks of parallel programming? Guided implementation

Thread Creation/destruction

Process/Creation/destruction

Message passing

Collective communication

Speculation

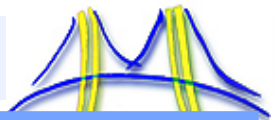
Transactional memory

Barriers

Mutex

Semaphores

Our Pattern Language 2.0



Applications

Choose your high level structure – what is the structure of my application?

Guided expansion

- Pipe-and-filter
- Agent and Repository
- Process Control
- Event based, implicit invocation
- Model-view controller
- Iteration
- Map reduce
- Layered systems
- Arbitrary Static Task Graph

Choose you high level architecture? Guided decomposition

Task Decomposition ↔ Data Decomposition

Group Tasks Order groups data sharing data access

Identify the key computational patterns – what are my key computations? Guided instantiation

- Graph Algorithms
- Dynamic Programming
- Dense Linear Algebra
- Sparse Linear Algebra
- Unstructured Grids
- Structured Grids
- Graphical models
- Finite state machines
- Backtrack Branch and Bound
- N-Body methods
- Circuits
- Spectral Methods

Productivity Layer

Refine the structure – what concurrent approach do I use? Guided re-organization

- Event Based
- Divide and Conquer
- Data Parallelism
- Geometric Decomposition
- Pipeline
- Discrete Event
- Task Parallelism
- Graph algorithms
- Digital Circuits

Efficiency Layer

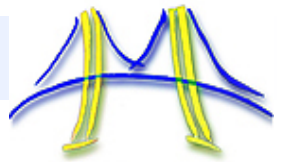
Utilize Supporting Structures – how do I implement my concurrency? Guided mapping

- Fork/Join
- CSP
- Distributed Array
- Shared Data
- Shared Queue
- Shared Hash Table
- SPMD
- Master/worker
- Loop Parallelism
- BSP

Implementation methods – what are the building blocks of parallel programming? Guided implementation

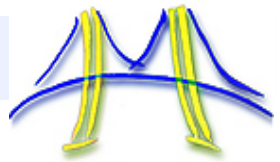
- Thread Creation/destruction
- Process Creation/destruction
- Message passing
- Collective communication
- Speculation
- Transactional memory
- Barriers
- Mutex
- Semaphores

Designated Emphasis (DE) in Computational Science and Engineering (CSE)



- Goals
- Participants (117 faculty from 22 departments – so far)
- How the DE works
- Resources and Opportunities
- Details at cse.berkeley.edu

Designated Emphasis (DE) in CSE



- New "graduate minor" – approved, starting 2008
- Motivation
  - Widespread need to train PhD students in large scale simulation, or analysis of large data sets
  - Opportunities for collaboration, across campus and at LBNL
- Graduate students participate by
  - Getting accepted into existing department/program
  - Taking CSE course requirements
  - Qualifying examination with CSE component
  - Thesis with CSE component
  - Receive "PhD in X with a DE in CSE"
  - Details at cse.berkeley.edu

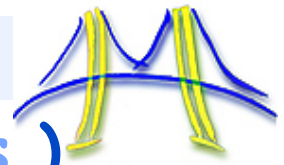
Participating Departments (1/2)

(# faculty by "primary affiliation", # courses)

- Astronomy (7,3)
- Bioengineering (3,1)
- Biostatistics (2,0)
- Chemical Engineering (6,0)
- Chemistry (8,1)
- Civil and Environmental Engineering (7,8)
- Earth and Planetary Science (6,3)
- EECS (19,14)
- IEOR (5,5)
- School of Information (1,0)

Participating Departments (2/2)

(# faculty by "primary affiliation", # courses)

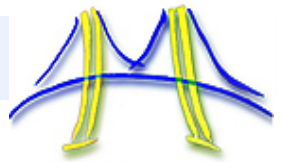


- Integrative Biology (1,0)
- Materials Science and Engineering (2,1)
- Mathematics (15, 4)
- Mechanical Engineering (9, 6)
- Neuroscience (7,1)
- Nuclear Engineering (2,1)
- Physics (1,1)
- Political Science (2,0)
- Statistics (5, 11)
- New: Biostatistics, Public Health

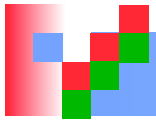
Course Structure

- 3 kinds of students, course requirements
  - CS , Math, "Applications"
- Each kind of student has 3 course requirements in other two fields
  - Goal: enforce cross-disciplinary training
  - Non-CS & Non-Math students:
    - » 1 or 2 Math courses from list
    - » 1 or 2 EECS courses from list
    - » Other classes from Stat, IEOR
  - Math & CS students: substitute 1 or 2 courses from "applied" department for 1 or 2 inside

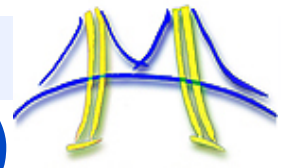
Example Course - CS267



- "Applications of Parallel Computing"
  - Long version of this short course!
  - see www.cs.berkeley.edu/~demmel/cs267_Spr10
- Taught every Spring, during Spr09 semester to:
  - UC Berkeley, UC Merced, UC Santa Cruz, UC Davis
  - All lectures on web (slides + video), freely available
- Google "parallel computing course" to get older version, with detailed text-book like notes on algorithms

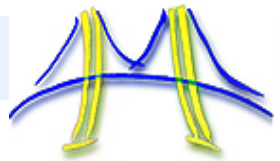


A few sample CS267 Class Projects (all posters and video on web page)



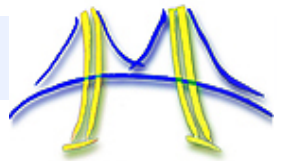
- Content based image recognition
  - "Find me other pictures of the person in this picture"
- Faster molecular dynamics, applied to Alzheimer's Disease
- Better speech recognition through a faster "inference engine"
- Faster algorithms to tolerate errors in new genome sequencers
- Faster simulation of marine zooplankton population
- Sharing cell-phone bandwidth for faster transfers

Some CSE Resources



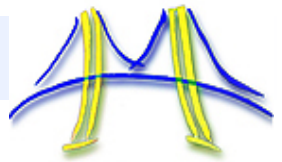
- Executive Director Masoud Nikravesh
  - nikravesh@cs.berkeley.edu
- Student Affairs Officer Pat Berumen
  - patbcoe@berkeley.edu
- Head Graduate Adviser Andy Packard
  - pack@me.berkeley.edu
- New courses …
- Computing resources
  - Cloud computing, start up allocations from LBNL/NERSC, clusters
- LBNL …

New CSE Courses Being Developed (campus and industrial support)



- Python for science
  - Josh Bloom (Astronomy)
  - 3 day summer short course (Aug 23-25) + seminar
- Understanding Molecular Simulation
  - Phil Geissler (Chem) and Berend Smit (ChemE)
  - Matlab based, students from Chem, ChemE, MSE, ME, BioPhys
- MatLab Applications in the Earth Sciences
  - Burkhard Militzer (Earth & Planetary Science)
  - Machine learning for understanding simulations/data sets
- Optimization Methods in Engineering
  - Laurent El Ghaoui (EECS)
  - Matlab (CVX) based, models not algorithms
- Other courses proposed

New CSE Courses Being Developed (campus and industrial support)



- Python for science
  - Josh Bloom (Astronomy)
  - 3 day summer short course (Aug 23-25) + seminar
- Understanding Molecular Simulation
  - Phil Geissler (Chem) and Berend Smit (ChemE)
  - Matlab based, students from Chem, ChemE, MSE, ME, BioPhys
- MatLab Applications in the Earth Sciences
  - Burkhard Militzer (Earth & Planetary Science)
  - Machine learning for understanding simulations/data sets
- Optimization Methods in Engineering
  - Laurent El Ghaoui (EECS)
  - Matlab (CVX) based, models not algorithms
- Other courses proposed

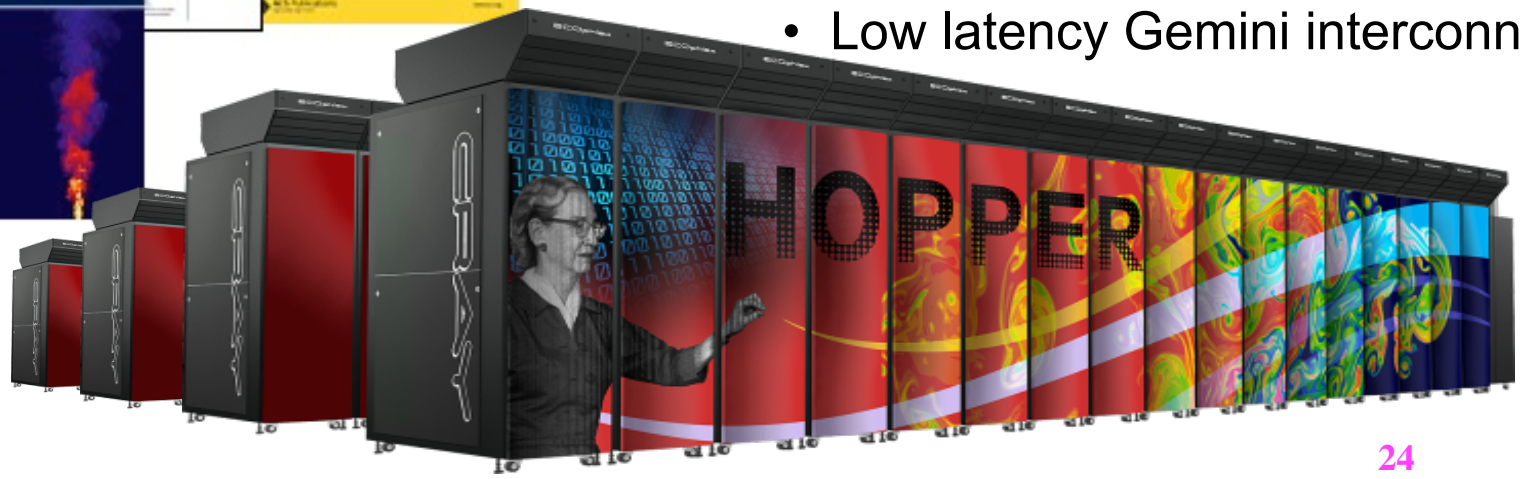
NERSC
National Energy Research Scientific Computing Center

www.nersc.gov

NERSC Overview

NERSC represents science needs
- Over 3000 users, 400 projects, 500 code instances
- Over 1,600 publications in 2009
- Time is used by university researchers (65%), DOE Labs (25%) and others

1 Petaflop Hopper system, late 2010
- High application performance
- Nodes: 2 12-core AMD processors
- Low latency Gemini interconnect



NERSC Systems

Large-Scale Computing Systems

Franklin (NERSC-5): Cray XT4

- 9,532 compute nodes; 38,128 cores
- Each node has an AMD quad core processor and 8 GB of memory
- ~25 Tflop/s on applications; 352 Tflop/s peak

Hopper (NERSC-6): Cray XE6

- Phase 1: Cray XT5, 668 nodes, 5344 cores
- Phase 2: > 1 Pflop/s peak (2 sockets / node, 12 cores / socket)

Clusters

105 Tflops total

Carver

- IBM iDataplex cluster

PDSF (HEP/NP)

- Linux cluster (~1K cores)

Magellan Cloud testbed

- IBM iDataplex cluster

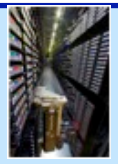
NERSC Global Filesystem (NGF)

Uses IBM's GPFS

1.5 PB; 5.5 GB/s

HPSS Archival Storage

- 40 PB capacity
- 4 Tape libraries



Analytics

Euclid (512 GB shared memory)

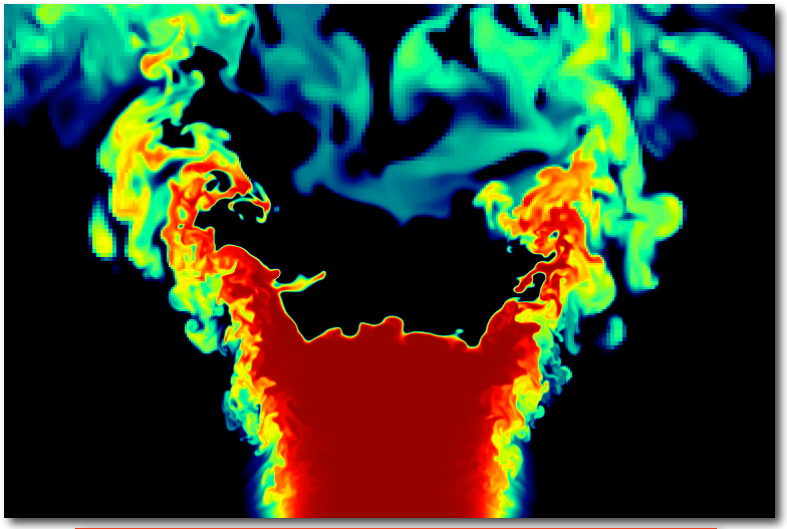
Dirac GPU testbed (48 nodes)

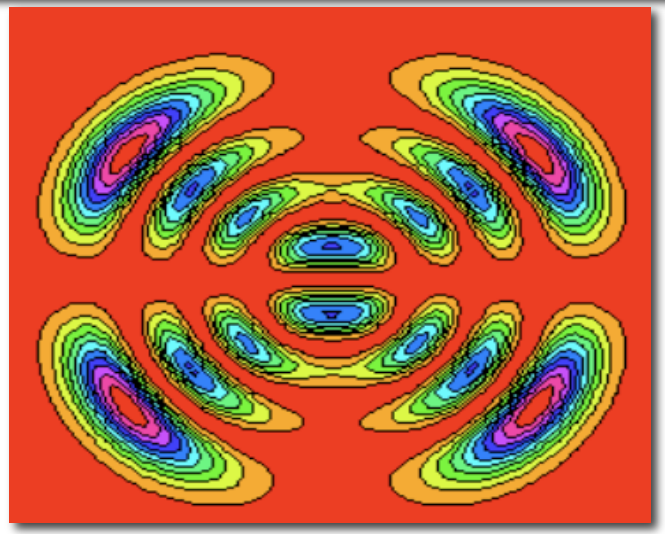
CRD
computational
research division

Computational Research Division

crd.lbl.gov

Computational Research Division



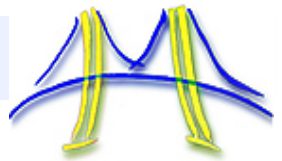


The Computational Research Division (CRD) creates computational tools and techniques that enable scientific breakthroughs, by conducting applied research and development in computer science, computational science, and applied mathematics.

CRD consists of three departments:

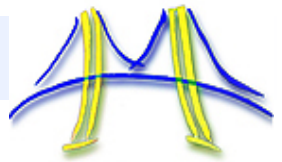
1. High Performance Computing Research (HPCRD) - Juan Meza
2. Advanced Computing for Science (ACS) - Deb Agarwal
3. Biological Data Management and Technology Center (BDMTC) - Victor Markowitz

Parallel Computing Short Courses – offered by LBNL



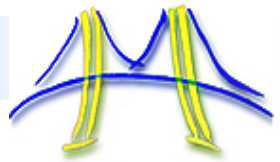
- 11th Workshop on DOE Advanced CompuTational Software (ACTS) Collection
  - Aug 17-20 – this week!
  - acts.nersc.gov/events/Workshop2010/
  - How to use selected computational tools developed for high performance computing
  - ScaLAPACK, PETSc, Hypre, Zoltan, GlobalArrays, …
  - Feel free to visit (their web site)
- Computational Science Summer School on Proven Algorithmic Techniques for Many-Core Processors
  - Aug 2-6, 2010
  - iccs.lbl.gov/news-summer-2010.html
  - GPU programming

Schedule and Instructors (1/3)



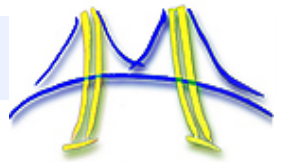
- Monday, Aug 16
  - 9-9:30 am – Introduction and Welcome
    - » Jim Demmel (UCB)
  - 9:30-12pm – Introduction to Parallel Architectures and Pthreads
    - » John Kubiatowicz (UCB)
  - 12-1:15pm – Lunch (see web page for suggested venues)
  - 1:15-2:15pm – Shared Memory Programming with OpenMP
    - » Barbara Chapman (U. Houston)
  - 2:15-3:00pm – Shared Memory programming with TBB
    - » Michael Wrinn (Intel)
  - 3:00-3:30pm – Break
  - 3:30-4:30pm – Parallel Advisor
    - » Mark Davis (Intel)
  - 4:30-5:00pm – Break/Transition to HP Auditorium, 306 Soda Hall
  - 5:00-6:00pm – Hands-on Lab (rooms announced in HP Auditorium)
  - 6:00pm –Reception in Wozniak Lounge!

Schedule and Instructors (2/3)



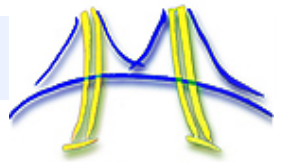
- Tuesday, Aug 17
  - 8:45-9:45am – Sources of Parallelism and Locality in Simulation
    - » Jim Demmel (UCB)
  - 9:45-10:45am – Distributed Memory Programming in MPI and UPC
    - » Kathy Yelick (UCB and LBNL)
  - 10:45-11:15am – Break
  - 11:15-12:15pm – Debugging Parallel Code
    - » Jacob Burnim (UCB)
  - 12:15-1:30pm – Lunch
  - 1:30-2:30pm – Architecting parallel software with design patterns
    - » Kurt Keutzer (UCB)
  - 2:30-3:00pm - Break / Transition to Wozniak Lounge, 4th floor Soda Hall
  - 3-6pm – Hands-on Lab

Schedule and Instructors (3/3)



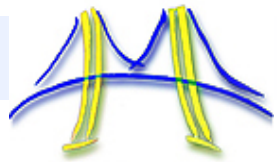
- Wednesday, Aug 18
  - 8:45-10:45am – Autotuning of Common Computational Patterns
    - » Jim Demmel (UCB)
  - 10:45-11:15am – Break
  - 11:15-12:15pm – Building Parallel Applications
    - » Nelson Morgan, David Wessel, Tony Keaveny, Leo Meyerovich (UCB)
  - 12:15-1:30pm – Lunch
  - 1:30-2:30pm – Cloud Computing
    - » Matei Zaharia (UCB)
  - 2:30–3:30pm – Performance Analysis Tools
    - » Karl Fuerlinger (UCB)
  - 3:30-4:00pm - Break
  - 4:00-5:00pm – GPU, CUDA, OpenCL Programming
    - » Mark Murphy (UCB)

Logistics

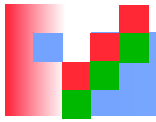


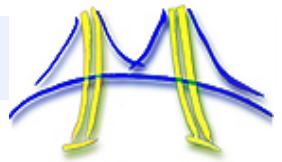
- Coffee
  - Available outside CITRIS Auditorium, not allowed in lecture hall!
- Live webcast of lectures
  - mms://media.citris.berkeley.edu/parlab2010
  - Email questions to parlabbootcamp2010@gmail.com
- Lecture Materials
  - Slides and archived video will be posted on bootcamp website
- Labs
  - Bring your own laptop
  - We supply wireless access, accounts on Franklin and Hopper
    - » Account problems: ask your TA
      - Razvan Carbunescu (Head TA), Michael Anderson, Erin Carson, Nick Knight
  - Lab assignment(s) posted at www.eecs.berkeley.edu/~carazvan/2010bootcamp/index.html

The Audience - you



- There are 335 registrants
  - 152 on-site, 183 off-site registrants
- As of last Wednesday (303 registrants)
  - 86 from 36 companies
  - 217 from 52 universities and research organizations, from 12 countries





LET'S GET STARTED!