January 12, 2013

XSEDE: Accessing and Utilizing Advanced Computing Hardware to make your research go faster

Dirk Colbry, XSEDE Campus Champion

XSEDE
Extreme Science and Engineering Discovery Environment

NSF

What is Advanced Computing Hardware?

- Anything more advanced than your desktop
- Local resources
  – Lab, Department, Institution (iCER)
- National resources
  – NSF (XSEDE), DOE (Jaguar) , Others
- Commercial Resources (cloud computing)
  – Amazon, Azure, Liquid Web, Others

NSF XSEDE

Why use Advanced Computing Hardware?

- Science takes too long
- Computation runs out of memory
- Run out of disk space
- Need licensed software
- Need advanced interface (visualization)

Agenda

- Who am I
- What is XSEDE
- Steps to High Performance
- Common classes of computational science problems

I am an engineer

- Mechanical Engineering degree from Georgia Tech
- 3 years as a Mechanical Engineer for Delta Airlines in Atlanta
- 2 years as a Robotics Engineer for FANUC Robotics in Auburn Hills

I am a researcher

- Ph.D. in CSE from MSU, focusing on image processing and pattern recognition
- BEACON Affiliate
- Adjunct Faculty in ECE
- Research interests include optimizing image analysis workflows for researchers
- Currently collaborating with researchers in Engineering, Biology, Math and Statistics

I am a computational consultant

- Institute for Cyber Enabled Research
  - One-on-one consulting
  - HPC Programming
  - Proposal Writing
  - Training and Education
  - Outreach

  - Reduce the "Mean time to Science"

I am a XSEDE Campus Champion

- Quick access to XSEDE Resources
- One-on-one consulting for XSEDE Users
- Personal Relationships with XSEDE support Staff

Agenda

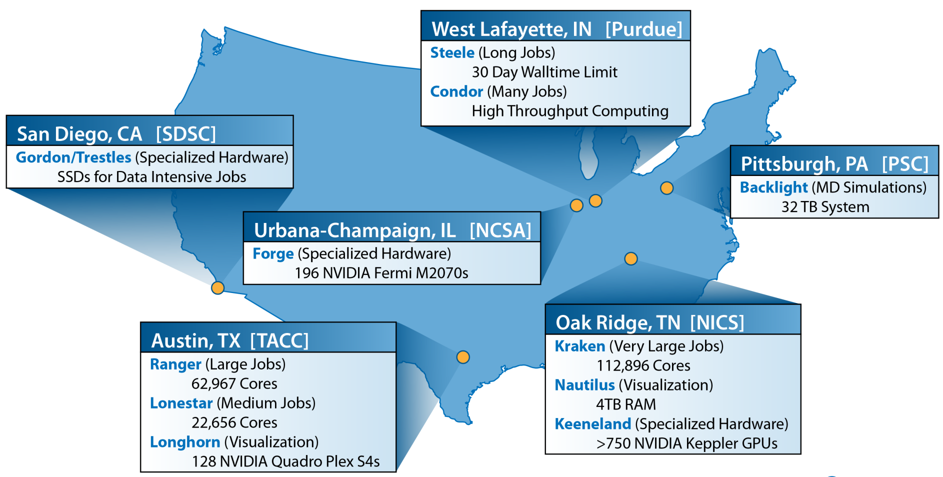
- Who am I
- What is XSEDE
- Steps to High Performance
- Common classes of computational science problems

What is EXSEDE?

- Formerly TeraGrid
- Funded by the National Science Foundation

"XSEDE is a single virtual system that scientists can use to interactively share computing resources, data and expertise. People around the world use these resources and services — things like supercomputers, collections of data and new tools — to improve our planet."

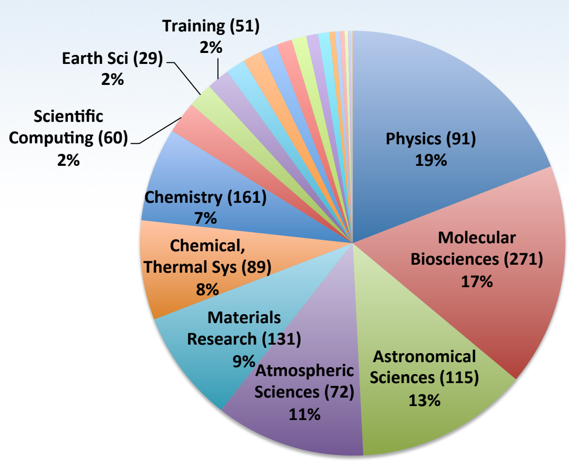
XSEDE Computing Resources



CONTACT: HELP@XSEDE.ORG

NSF 11 XSEDE

Who Uses XSEDE?



- >2 billion cpu-hours allocated
- 1400 allocations
- 350 institutions
- 32 research domains

NSF XSEDE

XSEDE supports a breadth of research

- Earthquake Science and Civil Engineering
- Molecular Dynamics
- Nanotechnology
- Plant Science
- Storm modeling
- Epidemiology
- Particle Physics
- Economic analysis of phone network patterns
- Brain science
- Analysis of large cosmological simulations
- DNA sequencing
- Computational Molecular Sciences
- Neutron Science
- International Collaboration in Cosmology and Plasma Physics

Sampling of much larger set. Many examples are new to use of advanced digital services. Range from petascale to disjoint HTC, many are data driven. XSEDE will support thousands of such projects.

13

Types of Allocations

FREE*

- Campus Champion
  - Get your feet wet with XSEDE
  - < 10k cpu-hours
  - 2 day lead time
- Start-Up
  - Benchmark and gain experience with resources
  - 200k cpu-hours
  - 2 week lead time
- Education
  - Class and workshop support
  - Short term (1 week to 6 months)
- XSEDE Research Allocation (XRAC)
  - Up to 10M cpu-hours
  - 10 page request, 4 month lead time

14

Steps to getting your allocation

- Step One – Campus Champion Allocation
  - Log onto the Portal and get an account
  - Send Campus Champion (me!) your portal account ID
- Step Two – Start-Up/Educational Allocation
  - Sign up for a startup account
  - Do benchmarking
- Step Three – XRAC
  - Requires written proposal and CVs

https://www.xsede.org/how-to-get-an-allocation

Other Resources

- Science Gateways
- Extended Support
- Open Science Grid
- FutureGrid
- Blue Waters (NCSA)
- Titan (ORNL/NICS)
- ALCF (Argonne)
- Hopper (NERSC)

Campus Champions

- Michigan State University
  - Dirk Colbry and Ben Ong (Here at iCER)
- North Carolina A&T State University
- University of Idaho
- University of Texas at Austin
  - TACC
- University of Washington
  - Jeff Gardner

NSF XSEDE

Agenda

- Who am I
- What is XSEDE
- Steps to High Performance
- Common classes of computational science problems

NSF XSEDE

Steps in Using Resource

- Connect to Resource (get an allocation)
- Determine required software
- Transfer required input files and source code
- Compile programs (if needed)
- Test software/programs on a developer node
- Write a submission script
- Submit the job
- Get your results and write a paper!!

Types of problems

- CPU bound
  – Lots of computing (think simulation)
- Memory bound
  – Requires lots of memory (think genomics)
- I/O bound
  – Requires lots of data (think astronomy)

(many problems fall in more than one category)

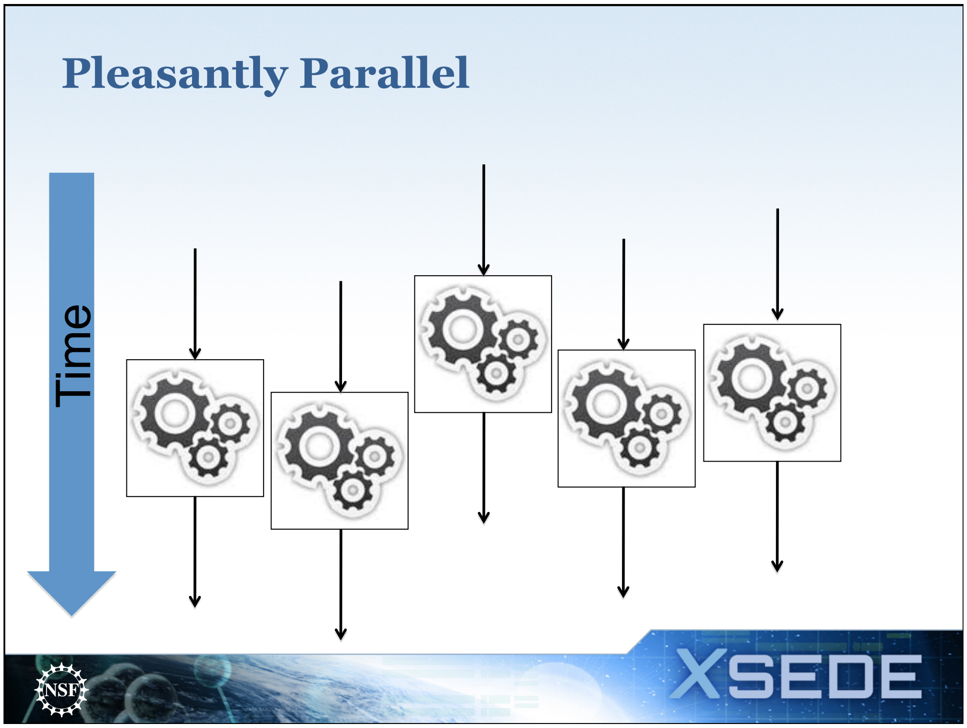
Steps to High Performance

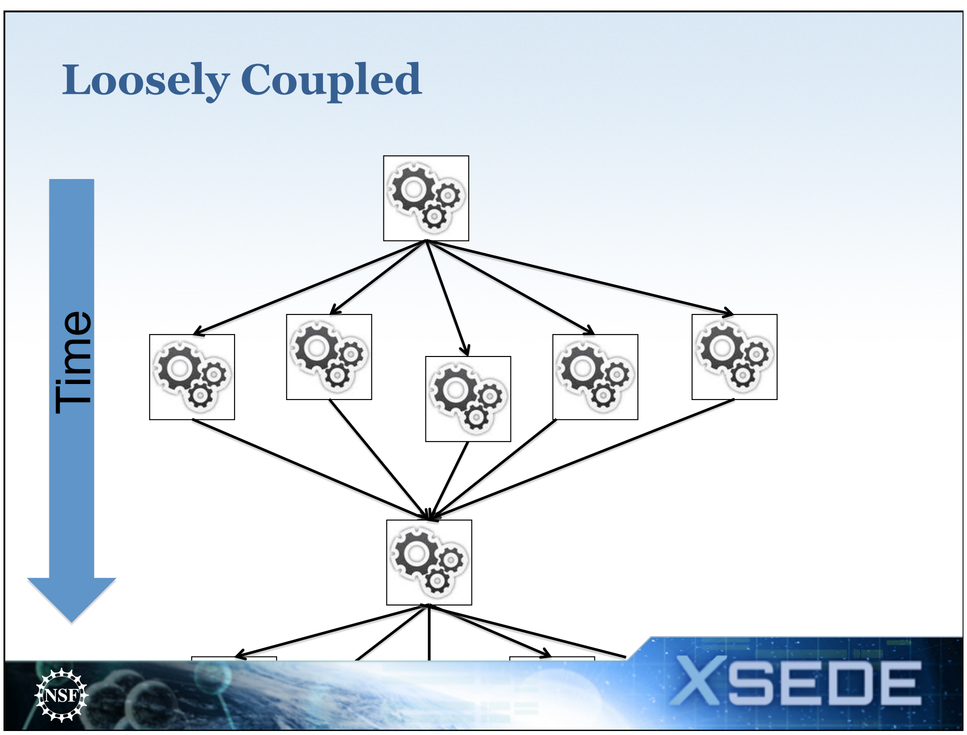
Note: Every application is different

1. Analyze your code
   - Profilers (gprof, vtune, tau)
   - Debuggers / memory trackers (gdb, totalview)
2. Optimize calculations
   - Trade memory for time (i.e., never do the same calculation twice)
3. Find ways to parallelize
   - Look for loops
   - Find iterations independent from each other
   - Determine how much information needs to be transferred

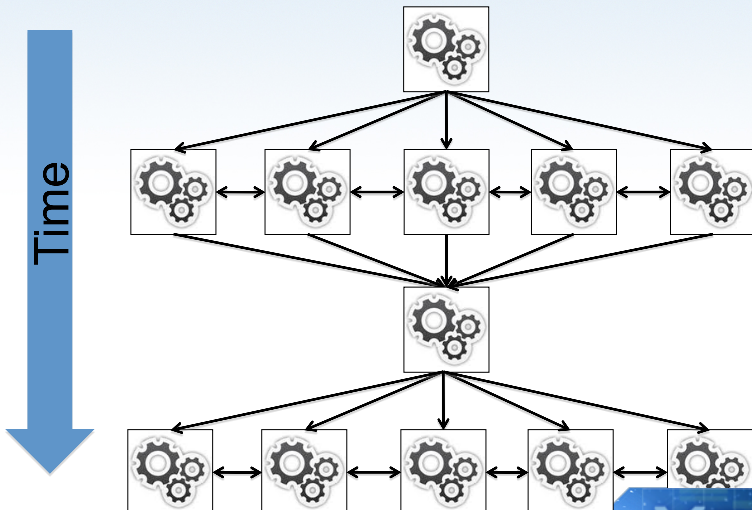
How much Communication?

- Pleasantly parallel
  - No communication required
- Loosely coupled
  - Typically sync at regular intervals
- Tightly coupled
  - Constant communication


Pleasantly Parallel
Time
XSEDE
NSF


Loosely Coupled
Time
XSEDE
NSF

Tightly Coupled



Types of communication

- Shared Memory
  - Typically very fast
  - OpenMP and GPGPU interfaces (sort of)
  - Doesn't scale well
- Shared File System
  - Slower communication / locking issues
  - Very easy to use
  - Speed of the network is a bottleneck
- Shared Network
  - Speed of the network is a bottleneck
  - Scales well
  - MPI and Map-Reduce interfaces

Agenda

- Who am I
- What is XSEDE
- Steps to High Performance
- Common classes of computational science problems

Example Problems

- Boundary Simulations
- Data Analysis
- Search

Example: Boundary simulations

1. Divide a 2D or 3D simulation space into a grid of cells
2. Define information that is transferred at the boundary of the cells
3. Simulate the dynamics of the cell during a time interval
4. Repeat steps 2 and 3

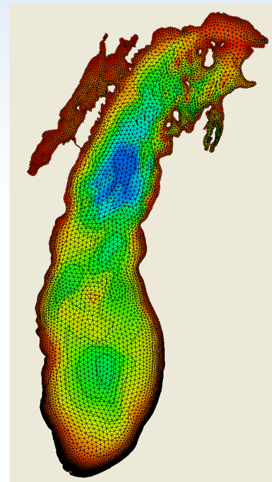
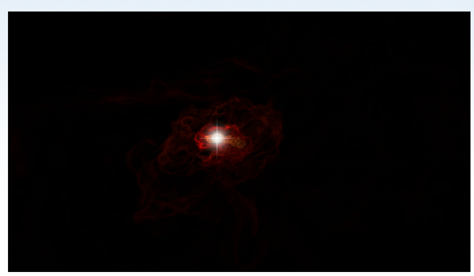


Image Provided by Dr. Mantha Phanikumar, MSU

Boundary Simulations

- Fluid dynamics
- Finite element analysis
- Molecular dynamics
- Weather
- Etc.

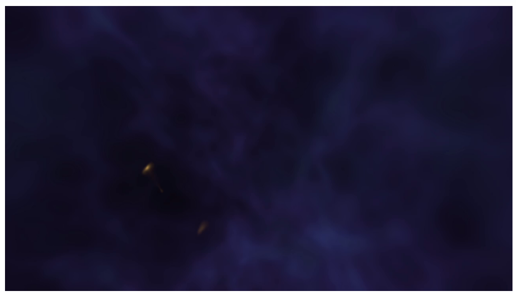


ENZO Simulation, Drs. O'Shea and Smith

- System of PDE (Partial Differential equations)
- Mathematically equivalent to inverse of a matrix

Boundary Simulation

- Tightly to loosely coupled
- Typically solved with MPI
- PDE solutions available for GPU and OpenMP

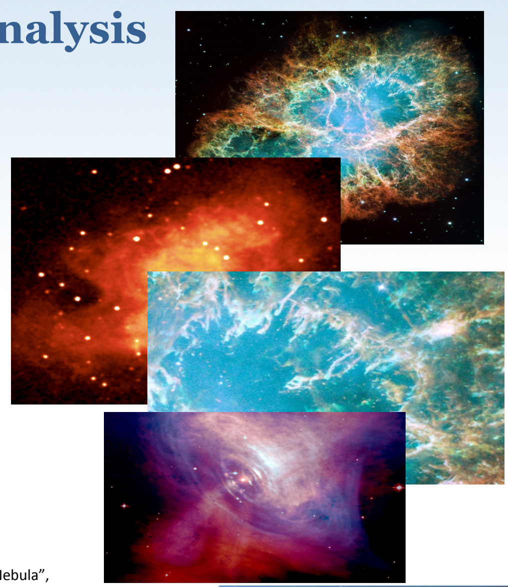


Simulation of the formation of a spiral galaxy
Drs. Turk, O'Shea and Smith

XSEDE

Example: Data Analysis

1. Input data file
2. Find features, search or filter data in some way
3. Output Results



Images from, "Understanding the H_2 Emission from the Crab Nebula",
C.T. Richardson, J.A. Baldwin, G.J. Ferland, E.D. Loh, Charles A. Huehn, A.C. Fabian, P.Salomé

XSEDE

Data Analysis

- Computer vision tasks
- Bioinformatics
- Astrophysics
- Etc.

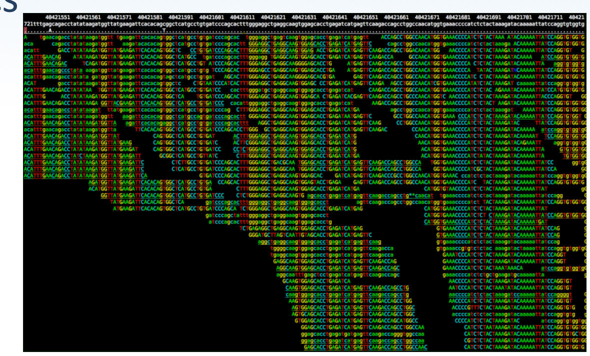
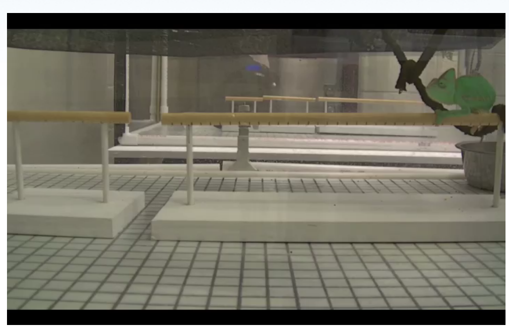


Image generated using SAMtools

Data Analysis

- Loosely coupled
- Bulk of computation is typically pleasantly parallel
- Can be I/O bound



Video Provided by Dr. Fred Dyer

Example: Search

- Randomly generate test candidates
- Evaluate the quality of solution
- Repeat until found

Image Provided by Dr. Warren F. Beck, MSU

Search

- Evolution (Avida)
- Genetic Algorithms (Heeds)
- RANSAC
- Monte Carlo
- Etc.

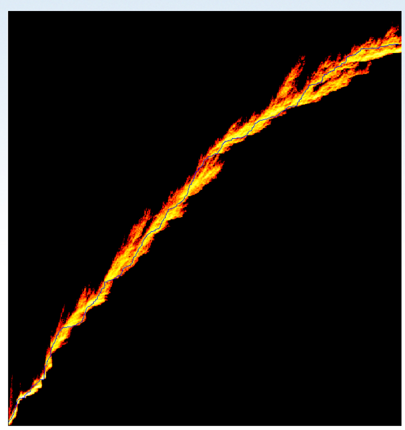
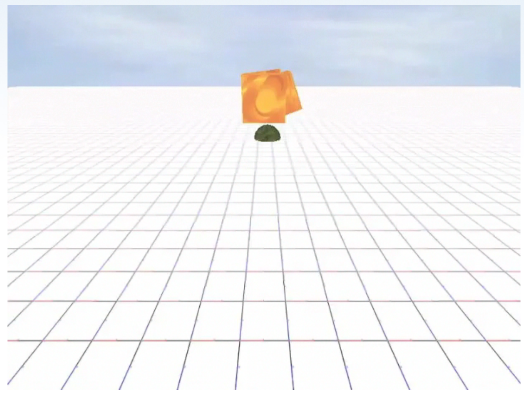


Image Provided by Dr. Charles Ofrea, MSU

Search

- Pleasantly parallel
- The more the better
- Typically not I/O bound
- Typically not memory bound



Evolution of an artificial organism that can move and forage for food, Dr. Nicolas Chaumont

XSEDE

Summary

- Advanced computational hardware can help you do more science faster
- Advanced computing typically requires knowledge of a primitive interface (command line)
- When parallelizing your computation,
  think about where the loops are,
  what needs to be communicated,
  and where there are bottlenecks
- There are many existing computational resources that can help you get started

XSEDE

Questions

- Local Campus Champion
  - That's me! colbrydi@msu.edu
- Centralized XSEDE help
  - help@xsede.org
- Extended one-on-one help (ECSS):
  - https://www.xsede.org/ecss
- Training
  - http://www.xsede.org/training