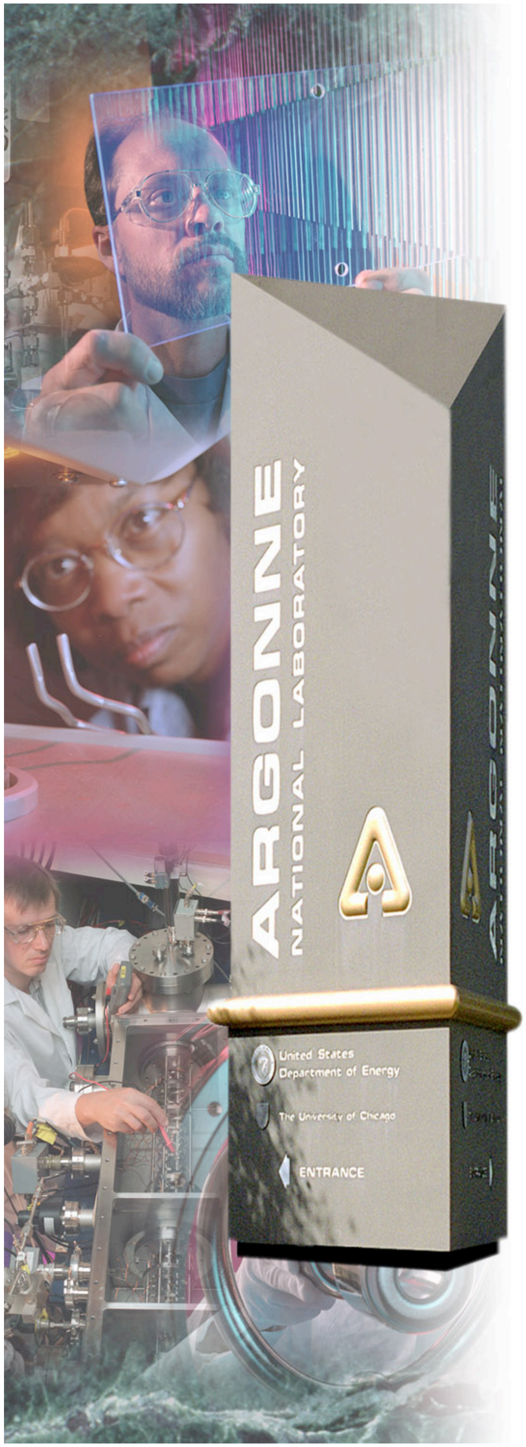


Thoughts on Capacity Computing

William D. Gropp
Mathematics and Computer Science
www.mcs.anl.gov/~gropp

Argonne National Laboratory

Office of Science
U.S. Department of Energy

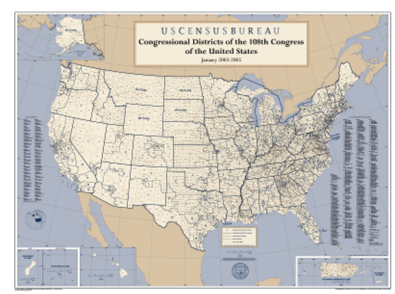
A U.S. Department of Energy
Office of Science Laboratory
Operated by The University of Chicago

What distinguishes capacity from capability?

- **It's the software! Why?**
- **Capacity is cheap (as perceived by users)**
- **So one definition of capacity:**
  - Systems where it is not cost-effective (to the users) to tune their code for performance
  - This is related to Marc's point from yesterday about expecting users of $10M machines to learn to use MPI IO effectively
- **A related one is based on Algorithms**
  - Capacity machines allow simpler algorithms because it is easier to meet scaling needs. E.g., the BSP model is adequate; a central master can be used for task farms
- **Another definition**
  - Capacity systems run (possibly parallel) Matlab
- **There are at least two types of capacity**
  - One laptop
  - Many laptops (ensemble studies)
- **Related to two types of use**
  - 1 user, zillion small or independent jobs
  - Zillion users, jobs perhaps too large for their laptop
  - … leads us to the next question

Role of capacity

- **Why do I need a capacity platform at all?**
  - How many people have a desktop in addition to their laptop? Why?
  - How much of the need for more compute is due to bad algorithms/ software (numerical recipes, normal equations for least squares; low-order approximations)?
  - How much is due to the zillion of small jobs case?
- **Where is my laptop inadequate?**
  - Runtime too long (capability from the bottom)
  - Memory too large
  - Need a affiliated resource (database, visualization, network)
- **Measured by direct publications, capacity computing is likely to generate the most papers**
  - Capacity is essential – 1000 node clusters in every congressional district would advance science
  - Of course, successful capability computing may change the direction of science or solve valuable engineering problems



Dominant Directions

- **I'll take the Software angle; thus:**
- **Accuracy and reliability**
  - Dan Reed described $200M benefit from predicting where to evacuate ahead of hurricane
  - "In August 1991, the Sleipner A, an oil and gas platform built in Norway for operation in the North Sea, sank during construction. The total economic loss amounted to about $700 million. After investigation, it was found that the failure of the walls of the support structure resulted from a serious error in the finite element analysis of the linear elastic model." (http://www.ima.umn.edu/~arnold/disasters/sleipner.html)
  - Doing this calculation today with error estimation is very possible (compute resources are available; these calculations were 16+ years ago)
- **Continued enhancement of simulations possible on laptops**
  - 1-d -> 2-d -> 3-d
  - Laptops now exceed Cray 1 in almost every measure

Technical challenges

- **Make better use of available resources**
  - For example, managing ensemble studies
  - Run on the idle cycles for laptops, and remaining desktops (or toasters, refrigerators, …)
  - This is a technical barrier because we need way to describe problems, manage workflows, handle faults, security
    - *No standard programming language for this!*
  - Note that running separate runs to make ensemble studies may not be the most efficient use of resources
    - *Stochastic programming*
    - *Better use of memory bandwidth in combining ensembles*
- **Exploiting Multicore Processors**
  - Automatic compilation for multicore is far off
  - Just look at research papers on SMP barriers and other operations.
    - *A compiler that could do a good job on such a simple operation could become a published author*
- **Power (but everyone will say that)**

Future capacity

- **Laptops**
- **Will there be a change in model?**
  - No, software doesn't change that fast ☺
- **Will web-services style change capacity computing?**
  - Perhaps, but many practical issues remain
  - Recall those 1000 node clusters in each congressional district. Even a modest node (multicore) should be 32 GF peak
  - Total capacity available from just these systems is roughly 32TF * 435 > 10PF