How to Program on 50,000 Processors

Karen Devine
Discrete Algorithms and Mathematics Department
Sandia National Laboratories, Albuquerque
kddevin@sandia.gov

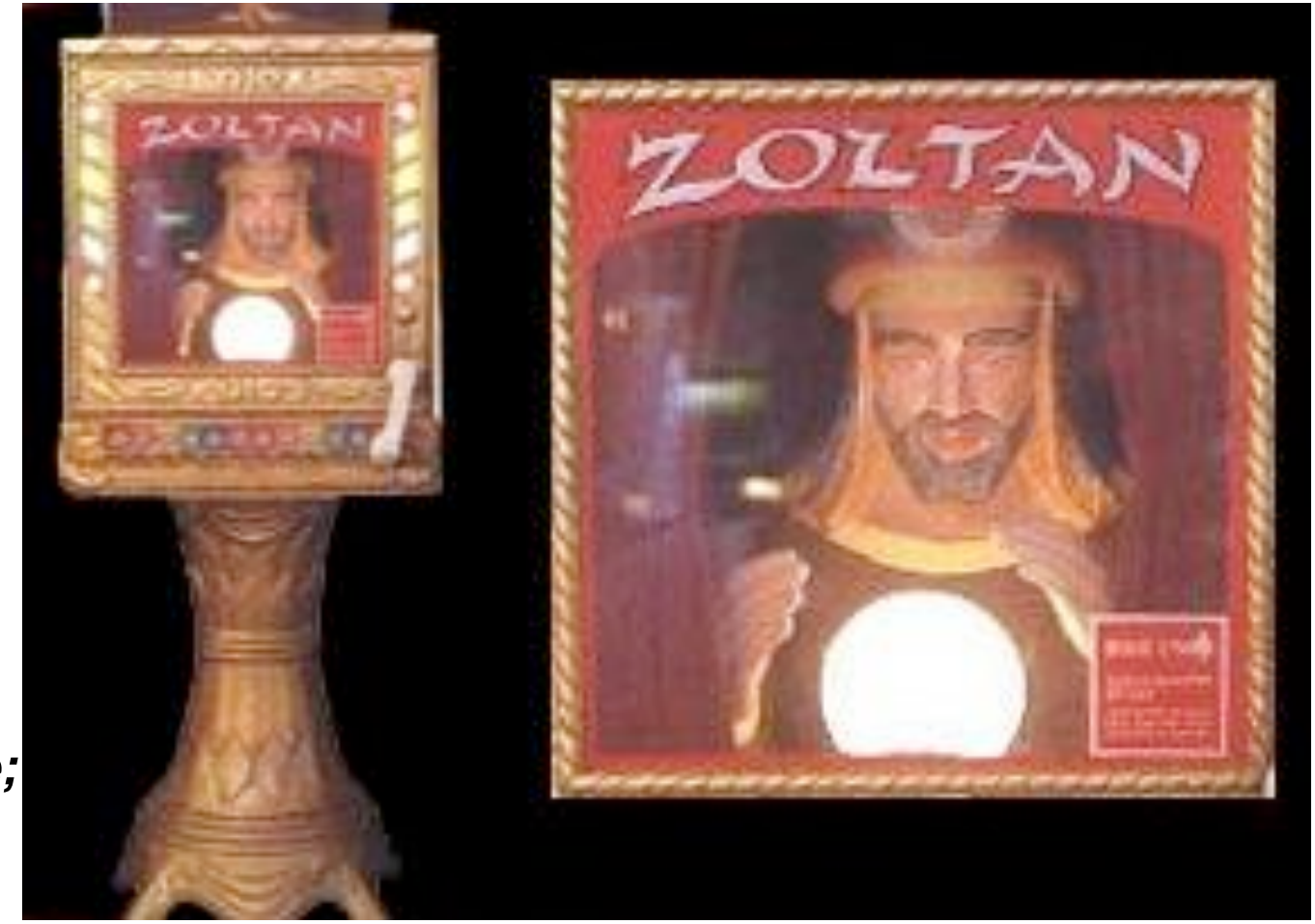
Work with:
Erik Boman, Bob Heaphy, Bruce Hendrickson (SNL)
Umit Çatalyürek (Ohio St.)
Rob Bisseling (SNL CSRI; Utrecht Univ.)
Robert Preis (SNL CSRI; Paderborn Univ.)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

50,000 Processors ?!?!

- Emerging problems require greater computing capability.
  - Multiscale simulations, MEMS, biology, data mining, …
- To make computers productive…
  - *Application software has to work with the hardware.*
  - More efficient computing models to reduce communication.
  - Accommodation of architecture characteristics.
- To get real science done…
  - *Software has to work for the application developers.*
  - Make software effective and easy-to-use.
- Examples from the Zoltan toolkit.
  - Support future computing needs.

Other Famous (?) Zoltans



Fortune-telling game;
Model for "Zoltar"
in movie "Big"

Other Famous (?) Zoltans

"When the Russian army unearths the vault of Dracula, they accidentally unleash his undead human slave and the Count's vampire hellhound Zoltan. But these fiends need a new master and head for Los Angeles to find Dracula's last living descendant, family man Michael Drake. Now with the help of an international vampire hunter, can Drake destroy Zoltan and his pack of blood-crazed devil dogs before 'man's best friend' can fetch the final soul of the damned?"

Zoltan, Hound of Dracula

1978 Movie starring Jose Ferrar

Available on DVD for $9.98 at amazon.com

Other Famous (?) Zoltans

Zoltan
the
Adequate

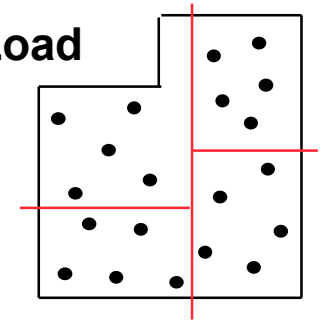
- "Geek magician" guaranteeing "100% adequacy in all he does."
- http://www.justadequate.com/zoltan.htm
- zoltan@justadequate.com

The Zoltan Toolkit

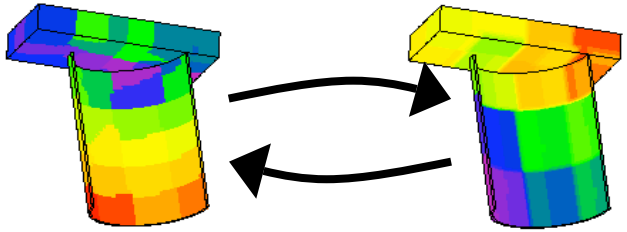
Data services for unstructured, dynamic and/or adaptive computations.

http://www.cs.sandia.gov/Zoltan

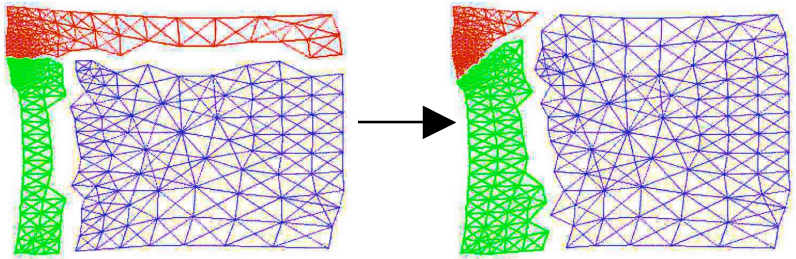
Dynamic Load Balancing



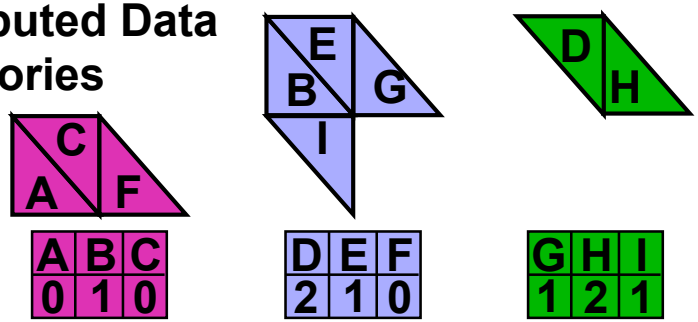
Unstructured Communication



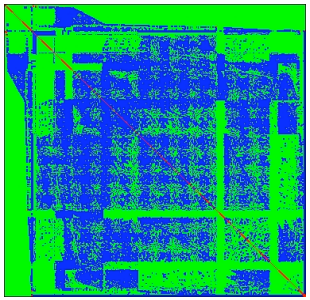
Data Migration



Distributed Data Directories



Matrix Ordering



Dynamic Memory Debugging

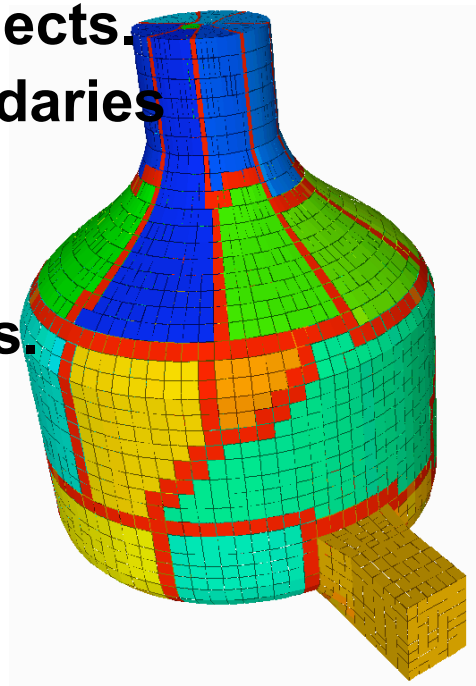
Partitioning / Load Balancing

- Partitioning problem: Assign work to processors to
  - Minimize processor idle time (i.e., balance loads), and
  - Minimize interprocessor communication.

- In next generation systems:
  - Processor speeds increasing faster than network speeds.
  - Decompositions with minimal communication costs are more important.

Traditional Graph Model

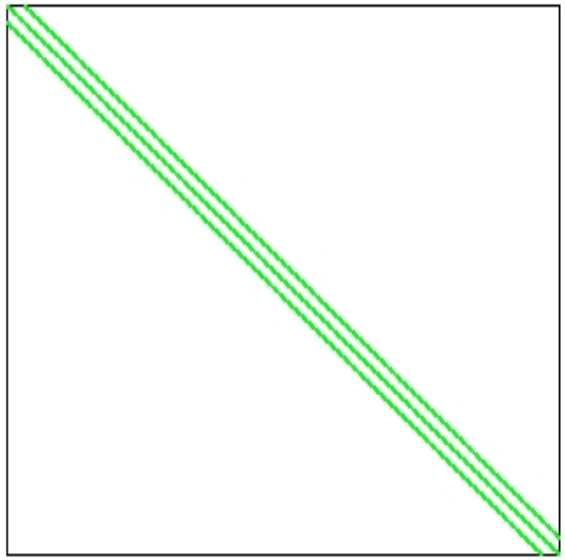
- Kernighan, Lin, Schweikert, Fiduccia, Mattheyes, Simon, Hendrickson, Leland, Kumar, Karypis, et al.
- Represent simulation as a (weighted) graph:
  - Vertices == computation associated with objects.
  - Edges == dependencies between two objects.
  - Weight of edges cut by subdomain boundaries *approximates* communication volume.
- Graph partitioning:
  - Assign equal vertex weight to processors.
  - Attempt to minimize weight of cut edges.



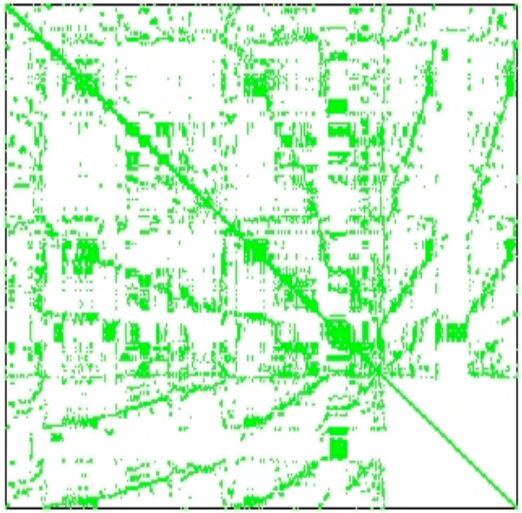
Success of Graph Partitioning

- **Widely used in finite element simulations.**
  - **Local support --> localized data dependencies.**
  - **Edge cut metric is reasonable approximation of communication volume.**

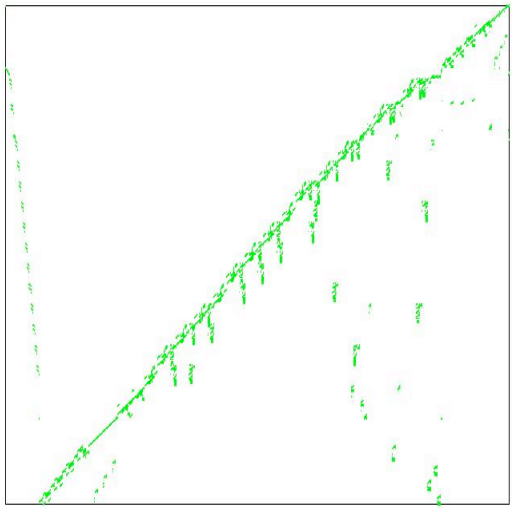
- **Serial & parallel libraries:**
  - **Chaco (Sandia)**
  - **METIS/ParMETIS (U. Minn.)**
  - **Jostle/PJostle (U. Greenwich)**
  - **Party (U. Paderborn)**
  - **Scotch (U. Bordeaux)**



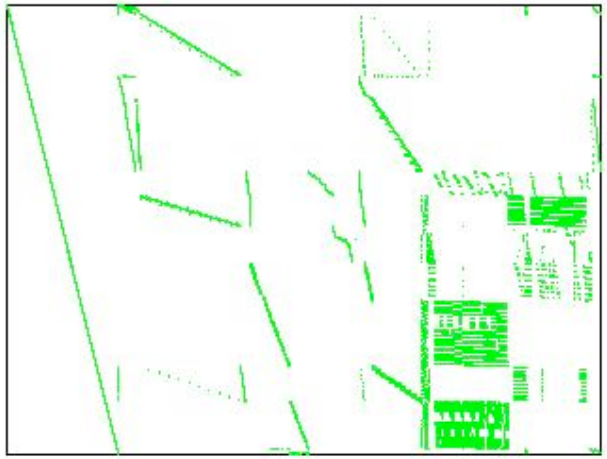
Where Graph Partitioning Is Insufficient



"Semi-dense," highly connected (Circuits, biology, databases)



Structurally non-symmetric (Chemical processing, DFT)

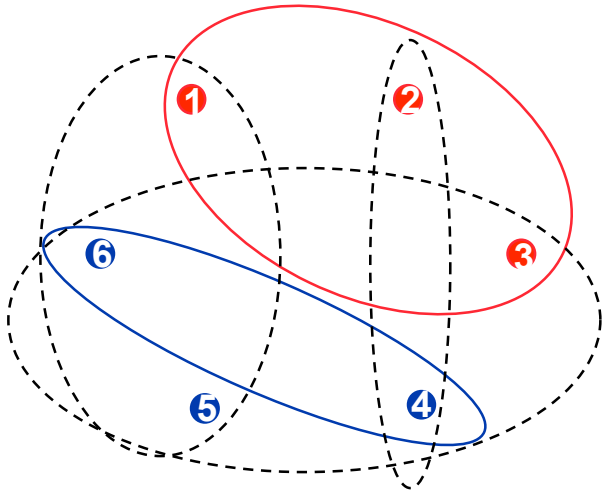


Rectangular (Linear programming, least squares)

Matrices from Tim Davis' Matrix Collection, U. FL

Alternative: Hypergraph Model

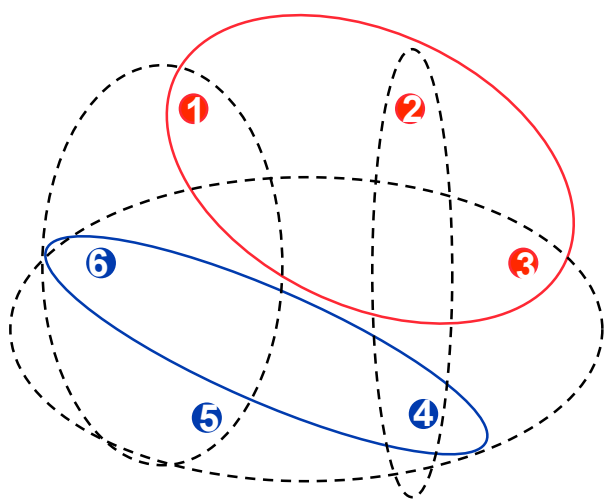
- Alpert, Kahng, Hauck, Borriello, Çatalyürek, Aykanat, Karypis, et al.
- Represent simulation as a (weighted) *hyper*graph.
  - Vertices == computation associated with objects.
  - Edges == dependencies between two *or more* objects.
- *Hyper*graph partitioning:
  - Assign equal vertex weight to processors.
  - Attempt to minimize weight of cut *hyper*edges.

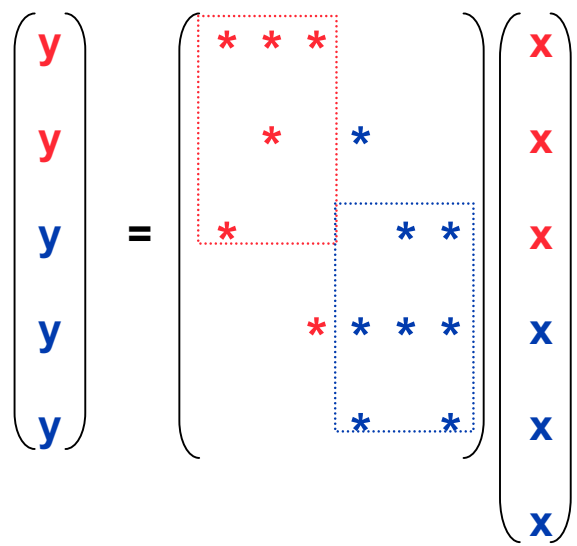


Hypergraph == Matrix

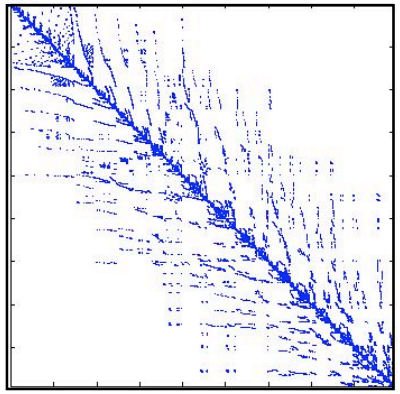
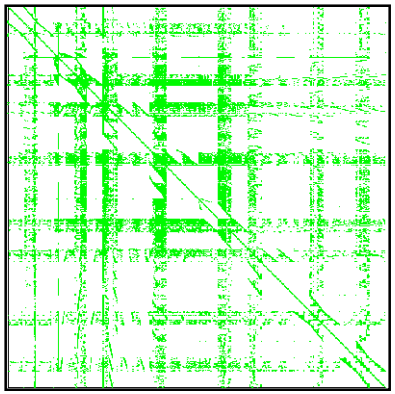
- View matrix as hypergraph. (Çatalyürek & Aykanat)
  - Vertices == columns
  - Edges == rows
- Can represent non-symmetric and/or rectangular matrices.
- Communication volume associated with edge *e* is *exactly*:

$$CV_e = (\#\text{ processors in edge } e) - 1$$





Graph vs. Hypergraph Partitioning

	DNA electrophoresis (Cage12, van Heukelum) 130K rows 2M nonzeros 			Circuit simulation: Xyce ASIC (Hoekstra) 320K rows 2.6M nonzeros 		
Partitioning Method	Comm. Volume	Reduction	Time (secs.)	Comm. Volume	Reduction	Time (secs.)
Graph (ParMETIS)	222,316		2.2	456,055		97.4
Parallel HG (Zoltan PHG)	184,861	17%	9.3	388,720	15%	7.9

Architecture-Aware Computing

- High-performance computing done on variety of architectures.
  - Clusters (of clusters (of clusters))
  - Clusters of shared memory processors
  - Supercomputers
  - Grids
- Heterogeneity in hardware.
  - Networks
  - Processors
- Enable applications to adapt to their environment.

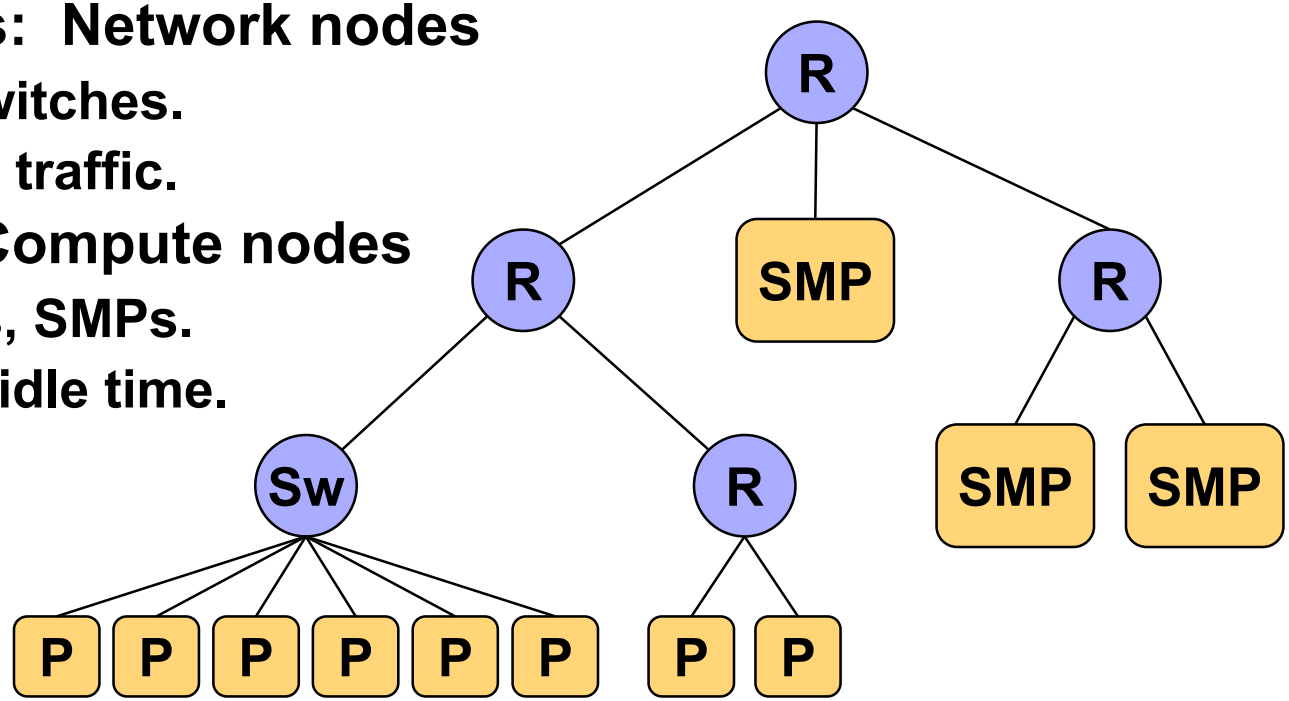
Determining the Environment

- Static environment represented by:
  - "Metadata" describing system (e.g., through files).
  - Benchmarks run to determine characteristics of environment.
- Examples:
  - Atlas (Dongarra, et al.)
  - PHiPAC (Bilmes, et al.)
  - Cache-aware multigrid (Douglas, Hu)
  - SALSA (Eijkhout, et al.)
  - Jostle graph partitioner's Network Cost Matrix (Walshaw, et al.)
- Dynamic/shared environment described by:
  - Processor utilization statistics
  - Network traffic and/or turnaround
- Examples:
  - Network Weather Service (Wolski, et al.)
  - Remos (Lowekamp, et al.)
  - DRUM (Faik, et al.)

Dynamic Resource Utilization Model (DRUM)

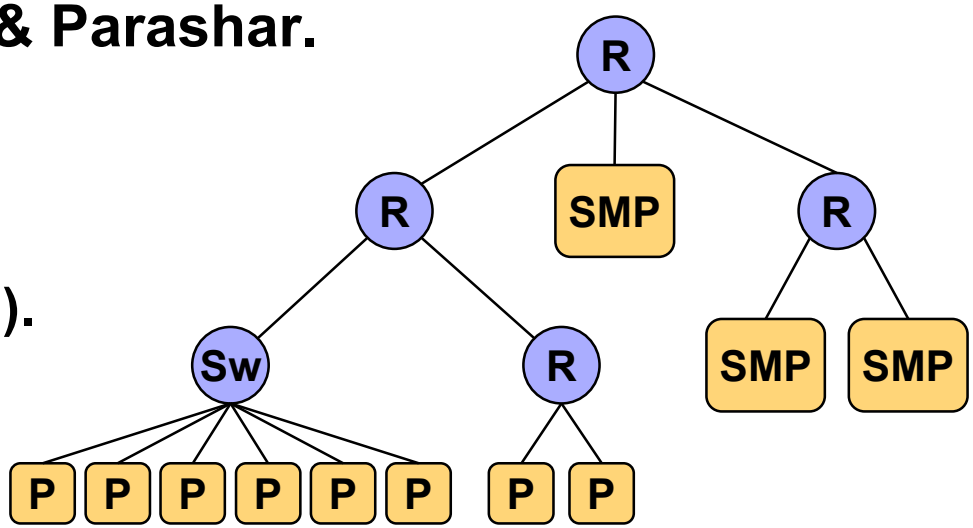
- Faik, Flaherty, Gervasio (RPI); Teresco (Williams)
- Accepts static machine configuration and benchmarks.
- Monitors dynamic CPU utilization & network traffic.
- Builds model of system as hierarchy of components.

- Internal nodes: Network nodes
  - Routers, switches.
  - Bandwidth, traffic.
- Leaf nodes: Compute nodes
  - Processors, SMPs.
  - Utilization, idle time.



Architecture-Aware Partitioning

- DRUM + Zoltan
- DRUM computes "power" of a node.
  - Combination of available computing time and network bandwidth.
- Zoltan uses "power" as percentage of work to assign to a node.
  - Input to standard Zoltan partitioner.
- Apply to leaves only or to each level of hierarchy.
- Similar approach: Sinha & Parashar.
- Extensions:
  - Memory limitations.
  - Partition to control heat generation (DeBenedictis).



Simulation Software Design

- New models and advanced architectures place great demands on application developers.

- Shouldn't expect physicists/chemists/biologists/etc. to deal with all the details.
  - Development outside area of expertise.
  - Time and resource consuming.
  - Reinventing the wheel.

What is the best approach for development of complex simulation software?

One Option: Frameworks

- Embed application in a simulation framework.
- Advantages:
  - *Many capabilities in one package.*
  - Common look and feel for framework applications.
  - Framework team provides support.
- Disadvantages:
  - *Many capabilities in one package.*
  - Require use of framework data structures.
  - Difficult to integrate with existing applications.
  - Steep learning curve for application developers.
  - Strong dependence on framework developers.

Another option: Software Toolkits

- Construct applications from smaller software "parts."
- "Tinker-toy parallel computing" -- B. Hendrickson
- Toolkits include …
  - Related services applications commonly need.
  - Support for wide range of applications.
  - Easy-to-use interfaces.
  - Data-structure neutral design.
- Toolkits avoid …
  - Prescribed data structures
  - Heavy framework
  - Limited freedom for application developers.

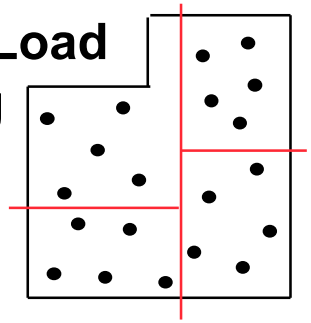
Hasbro, Inc.

The Zoltan Toolkit

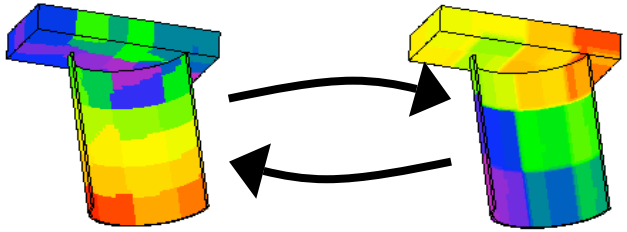
Data services for unstructured, dynamic and/or adaptive computations.

http://www.cs.sandia.gov/Zoltan

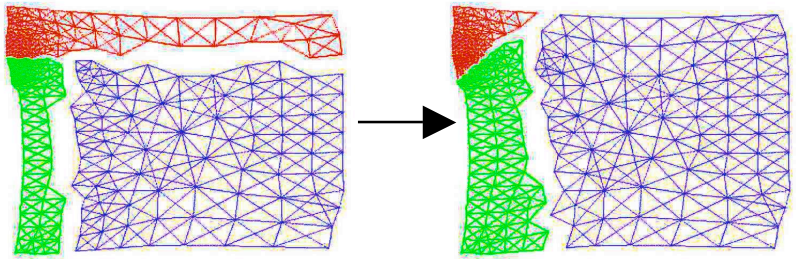
Dynamic Load Balancing



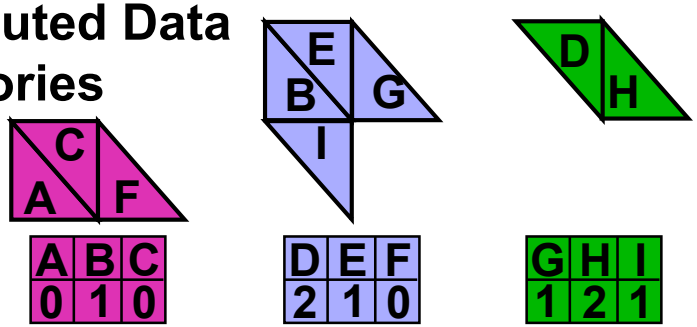
Unstructured Communication



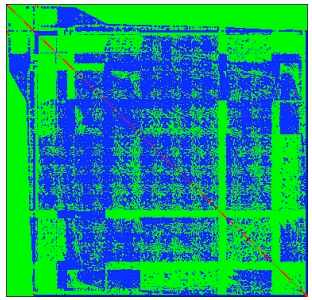
Data Migration



Distributed Data Directories



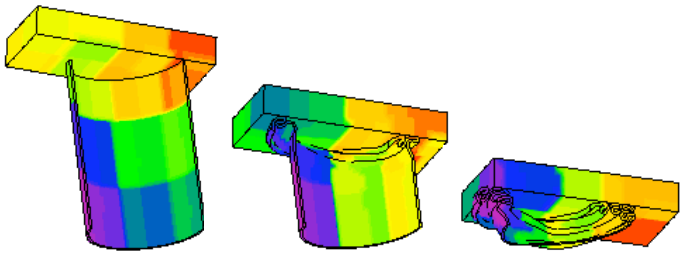
Matrix Ordering



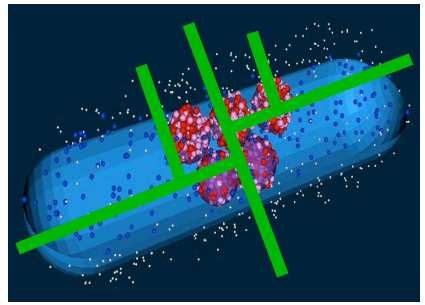
Dynamic Memory Debugging

Support for Many Applications

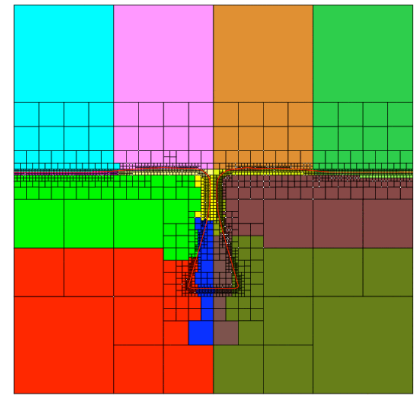
- **Data-structure neutral design supports different application data structures.**



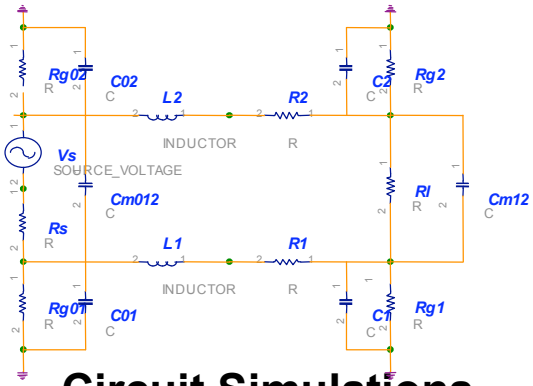
Contact detection
(ACME)



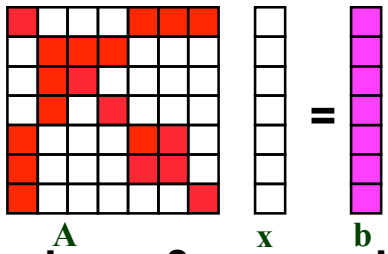
Particle Simulations
(ChemCell)



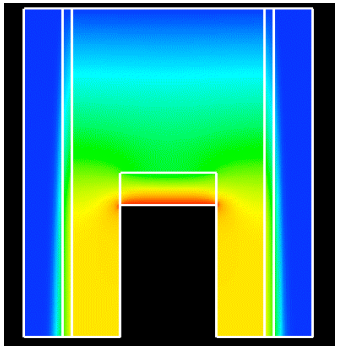
Adaptive Mesh Refinement
(Chisels, SIERRA,
ALEGRA/Nevada, CTH-AMR)



Circuit Simulations
(Xyce)



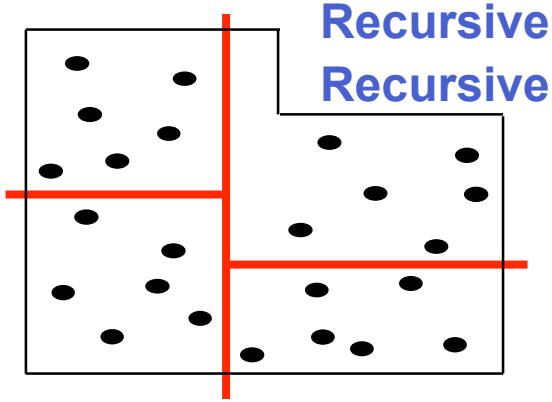
Linear solvers & preconditioners
(Trilinos, ML)



Multiphysics Simulations
(MPSalsa)

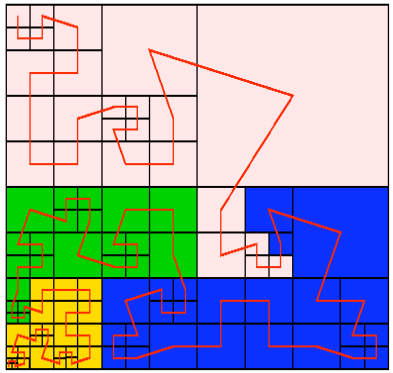
Zoltan Toolkit Flexibility

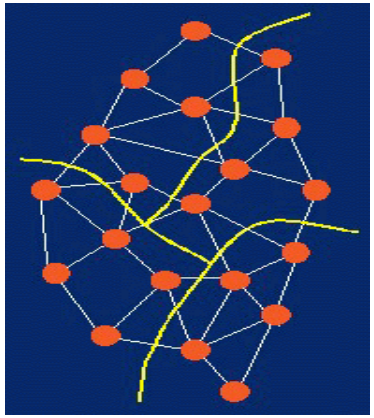
- **Different application needs: no one-size-fits-all solutions.**



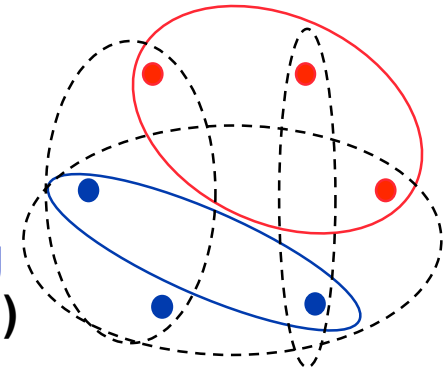
Recursive Coordinate Bisection (Berger, Bokhari)
Recursive Inertial Bisection (Taylor, Nour-Omid)

Space Filling Curves (Peano, Hilbert)
Refinement-tree Partitioning (Mitchell)
Octree Partitioning (Loy, Flaherty)





ParMETIS (Karypis, Schloegel, Kumar)
Jostle (Walshaw)



Hypergraph Partitioning
(Catalyurek, Aykanat, Karypis, Bisseling)

Toolkit Examples

Linear/Nonlinear solvers
- PETSc (Argonne)
- Trilinos (Sandia)

ODE solvers
- Sundials (LLNL)

Dynamic load balancing
- Zoltan (Sandia)
- DRAMA (NEC/Europe)

APPLICATION

Optimization
- TAO (Argonne)

Communication
- UPS (LANL)

Mesh adaptivity
- Pyramid (JPL)
- MeshSim (Simmetrix, Inc.)
- AOMD (RPI)

Visualization
- VTK (Kitware, Inc.)

Toolkits: Pros and Cons

- Advantages
  - "Expert" implementations of needed functionality.
  - Easy to add to existing applications.
  - Less cumbersome and limiting than frameworks.
  - Broad testing community.
  - Time and cost saving.
- Disadvantages
  - Trust (but open-source distribution can help).
  - Implement interfaces (but easier than implementing algorithms or adopting framework data structures).
  - Memory and performance costs (maybe; maybe not).

What else do we need to program 50,000 processors?

- *Accurate communication models (e.g., hypergraphs)*
- *Architecture-aware software*
- *Effective software design (e.g., toolkits)*
- **Robust Compilers**
- **"Flight Data Recorder"**
- **Debuggers**
- **Profilers**

For more information…

- Zoltan Home Page
  - http://www.cs.sandia.gov/Zoltan
  - User's and Developer's Guides
  - Download Zoltan software under GNU LGPL.
  - Hypergraph and architecture-aware partitioning to be released Q4 FY05.

- Email:
  - zoltan@cs.sandia.gov
  - kddevin@sandia.gov