Distributed Computing

&

Nuclear Reactor Analysis

Forrest B. Brown & Keith L. Derstine

Reactor Analysis Division
Argonne National Laboratory

(708) 252-9622
fbbrown@anl.gov

(708) 252-7718
klderstine@anl.gov

Abstract for contributed presentation, 1993 PVM Users Group Meeting (May 10-11, 1993, Knoxville TN)

Distributed Computing and Nuclear Reactor Analysis

Forrest B. Brown & Keith L. Derstine
Argonne National Laboratory

For over 30 years, Argonne National Laboratory has developed and applied large-scale software systems for the engineering analysis of nuclear reactor plants. Large code systems such as VIM (3D continuous energy Monte Carlo), DIF3D (3D finite-difference or nodal diffusion theory), MCC-2 (cross-section generation), SASSYS (transient analysis of reactor plants), etc., were developed and run on mainframe computers and supercomputers.

In the past two years, all of the reactor analysis codes and supporting software have been adapted to a distributed computing environment, based on a network of UNIX workstations. An integrated support environment and distributed computing methodology were established on the division network, including the following major components: (1) UNIX operating systems, with tcp/ip networking, (2) special configuration of the memory, disk, and scratch space on each workstation to accommodate both interactive and batch usage for large jobs, (3) DQS, the Distributed Queueing System from Florida State University, to manage the scheduling and load balancing for batch jobs, (4) the PVM software package from Oak Ridge National Laboratory to provide a framework for parallel processing of large applications, (5) the P4 package from Argonne to provide additional support for parallel applications, (6) numerous Bourne shell, C shell, and perl procedures to provide transition and convenience features for a diverse user group, (7) libraries of compatibility routines for I/O and environmental support, (8) source code management and version control using SCCS, (9) frequent and reliable backup services, and (10) continued development of the large production codes to improve their performance in a distributed computing environment. At present, the division workload is fully supported by a network of 15 SUN workstations, 2 IBM RS6000 workstations, and 50 NCD X-terminals. About 3,000 batch jobs are run with DQS in a typical month.

For computationally intensive tasks such as Monte Carlo and SPRT (sequential probability ratio testing), current development is focusing on parallel processing using PVM and P4. The PVM concept of a “virtual supercomputer” and the DQS approach to batch work are central to our environment. Parallel applications are intended to be scalable, so that development and small problems can be accomplished on the workstation network, and large applications can be off-loaded to Argonne’s new IBM SP1 parallel computer. The use of a message passing approach ensures that such portability and scalability can be realized.

Computing Technology Advances

Evolution —

- Commodity μ-processors $\Rightarrow$ ~2x gain/year in CPU power
- UNIX operating systems + networking $\Rightarrow$ distributed computing
- Workstations & parallel computers $\Rightarrow$ approaching CRAY power
- Price / performance $\Rightarrow$ ~10x less than CRAY

Revolution —

- CRAY Supercomputers $\Rightarrow$ **real** supercomputing
- Massively Parallel computers $\Rightarrow$ **cheap** supercomputing (??)
- Workstation Network $\Rightarrow$ **free** supercomputing (??)

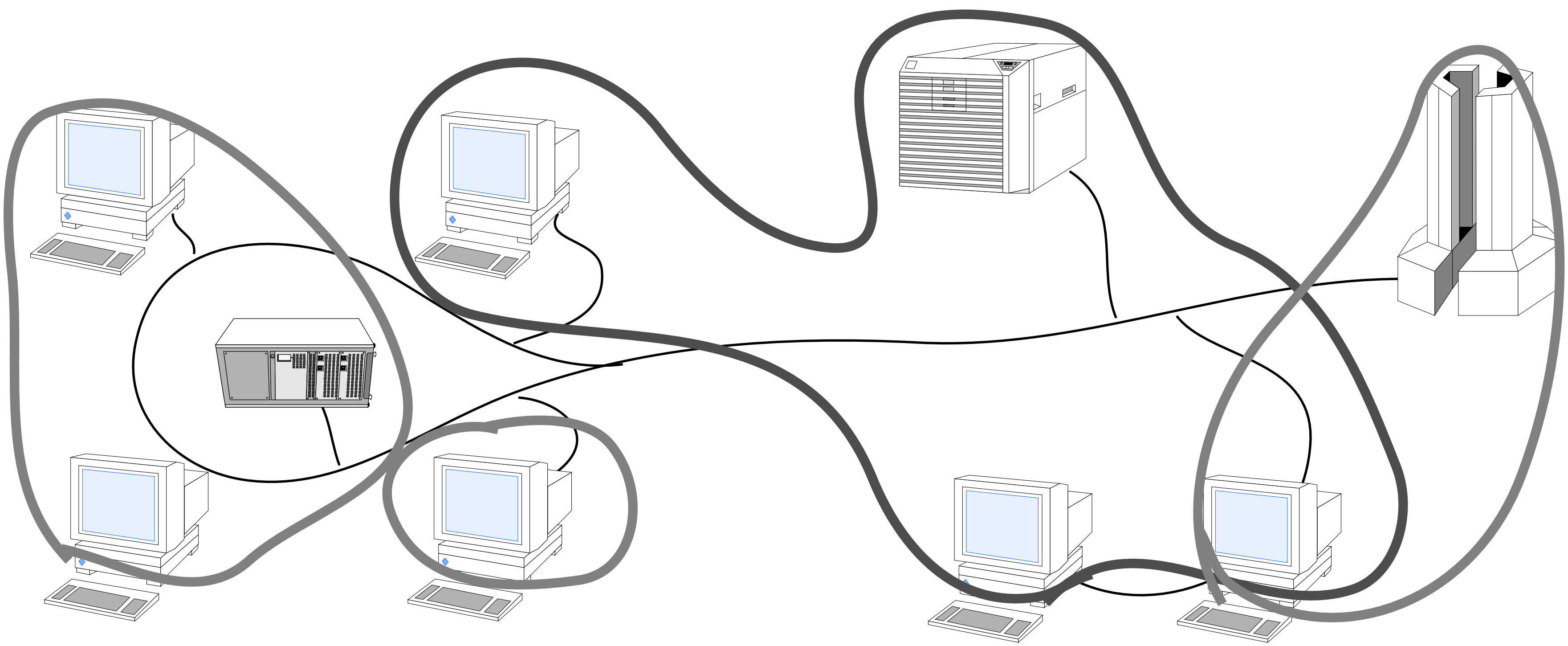
$\Rightarrow$ **Everyone now has access to "virtual supercomputers"**

Virtual Supercomputers

Hardware — Workstations + Network + Mainframes

- Mix & match computer hardware according to needs
- Flexible, cost-effective

Software — UNIX + tcp/ip + DQS + PVM + P4 + applications + ...



Reactor Analysis Division

Distributed Computing

Workstations + Network + Parallel Computing + Applications

RA Division computer hardware:

- 50 NCD X-terminals
- 15 SUN workstations (5 sparc2, 7 sparc-elc, 2 sparc1+, 1sparc1)
- 2 IBM workstations (2 rs6000/350)
- 64 MBytes memory on every workstation
- 40 GBytes disk storage
- Total CPU power ~ 3 Cray-XMP CPU's

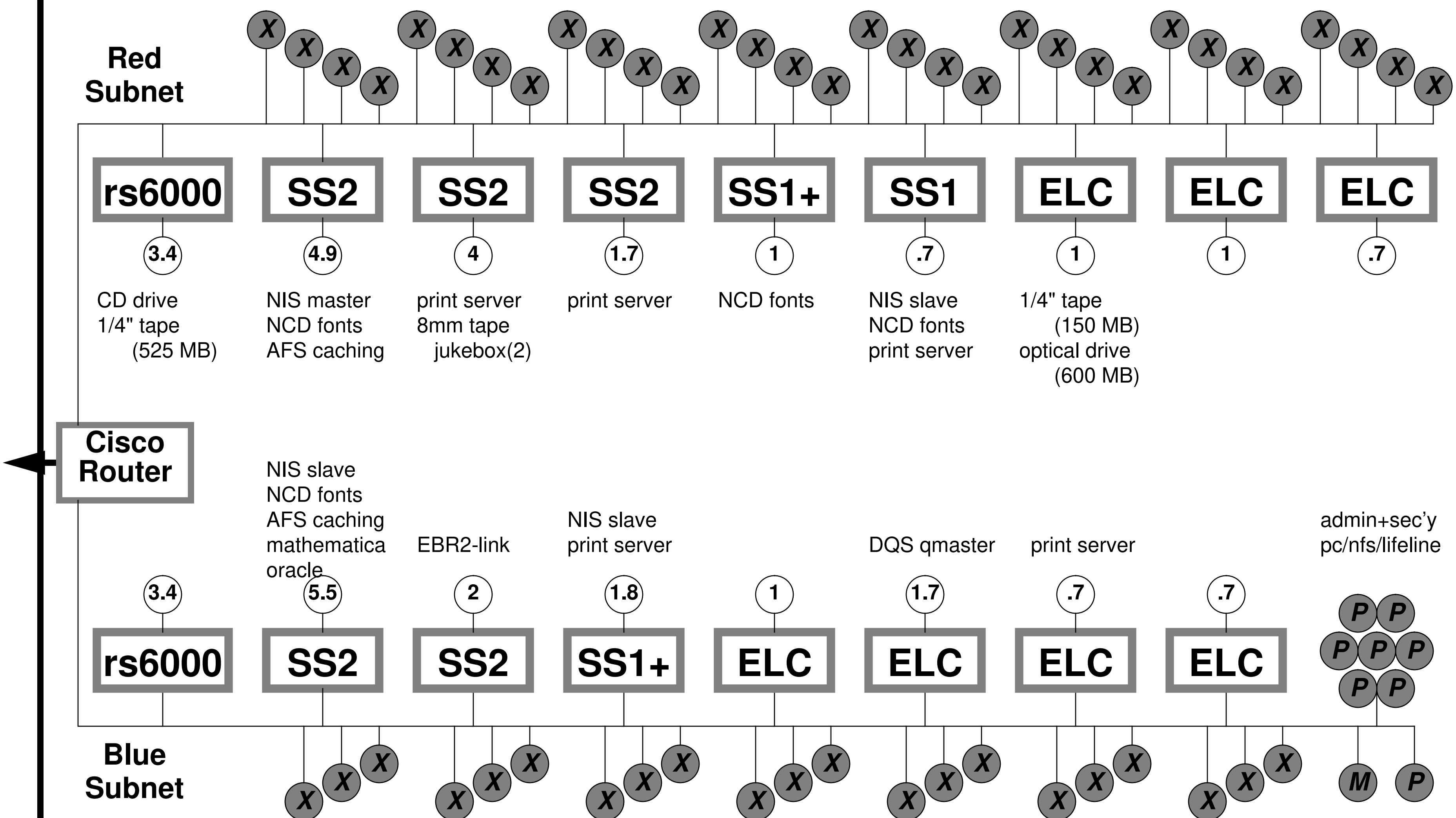
Issues:

- Total workload — job submission, scheduling, load-balancing, ...
- Large calculations — "virtual supercomputers", parallel programs, ...
- Applications software for nuclear reactor systems

Reactor Analysis Division ARGONNE

RA Network

X X-terminal
P PC
n n GB disk
M Mac



RA Network

Typical Node Configuration

- dataless client with 64 MB memory
- xdm daemon on each host, supports
  4 NCD X-terminals
- > .65 GB of disk space
  for root, swap, /var/tmp
- 200-400 MB /var/tmp reserved for
  local (batch) computations
- ~200 MB swap space (3x64)
- /tmp included in swap,
  via TMPFS on Sun

- supports interactive + batch use
- node is "owned" by the division,
  not by any single person
- home directories concentrated on
  3 Sparc2 servers
- 4 short-term filesystems (<28 days)
  on SS2's & RS6000's
- home & short are NFS-mounted,
  R/W access to all worksations
- single mail-server, mail-spool
  NFS-mounted
- lpr access to all supported printers

Parallel & Distributed Computing

Workstation Networks

- Justified by gains in productivity of individual engineers
- Can leverage investment into "virtual supercomputer"

Large-scale Distributed Computing — Requirements:

- Job queueing system
  - — scheduling lots of "ordinary" jobs on a distributed network
  - — load-balancing among different machines
- Parallel processing system
  - — fast turn-around for urgent work
  - — capabilities for very large problems
- Applications software

DQS

Distributed Queueing System — from SCRI - FSU

Features:

- Submits jobs to next available free machine
- Balances load among different machines
- Support for heterogeneous networks
- Support for parallel processing with PVM or P4 (RA Div.)

Status:

- Supported by RA Division
  - — 3,000 jobs/month using DQS batch
  - — negligible impact on interactive use
  - — RA shell scripts work with DQS
- Also supported by ECT Division — lab-wide, /AFS
- Currently using version 2.0, starting to implement version 2.1

Special Concerns for DQS — PVM — P4

DQS & RA Network

- Because of X-terminal support, ***qidle*** is not implemented
- Must allow for jobs of up to ***1 week*** duration (VIM Monte Carlo)
- General ***group/queue*** structure:
  - — groups: *sun*, *sun.now*, *ibm*, *ibm.now*
  - — queues: 1 "normal" + 1 "now" on each SUN
3 "normal" + 3 "now" on each RS6000
  - — each "normal" queue subordinate to corresponding "now" queue

DQS & parallel jobs

- Accommodate both PVM & P4 using shell-scripts
- On qsub: specify "*-G grp n .....*", but **not** "*-pvm*" or "*-M qname*"
- In job: — shell script in job constructs either

  hostfile — PVM, specifies machines
procgroup — P4, specifies program-machine mapping

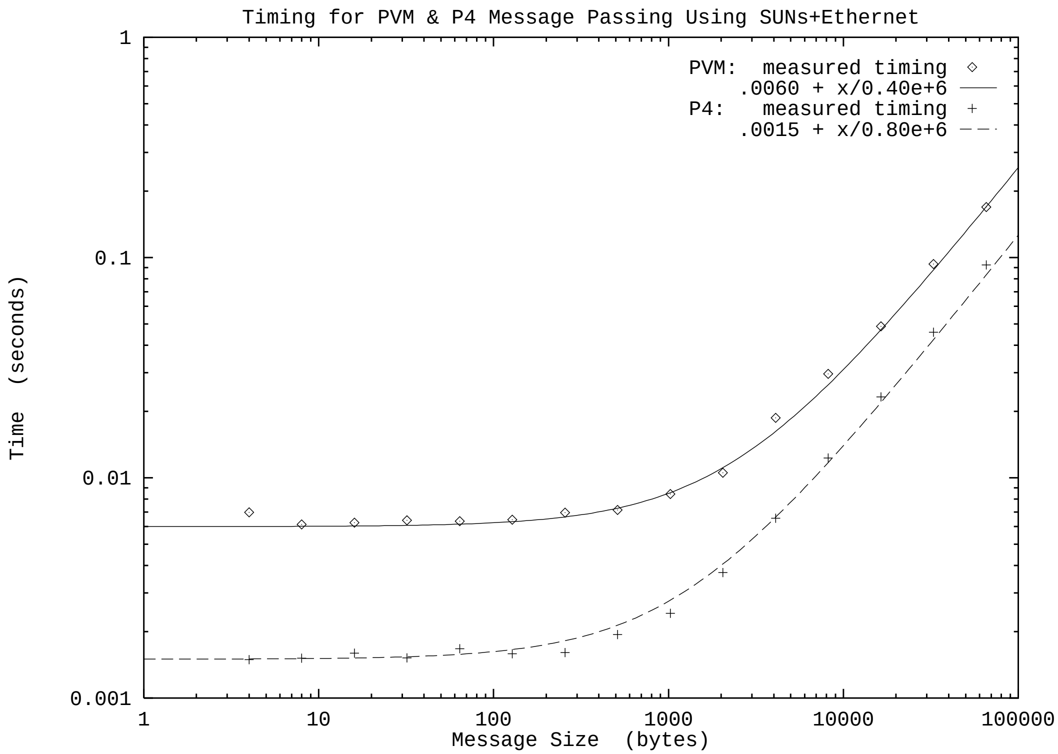
  — shell script also creates cleanup script
— shell script starts ***pvmd*** (for PVM)

PVM & P4

Computational power for large calculations

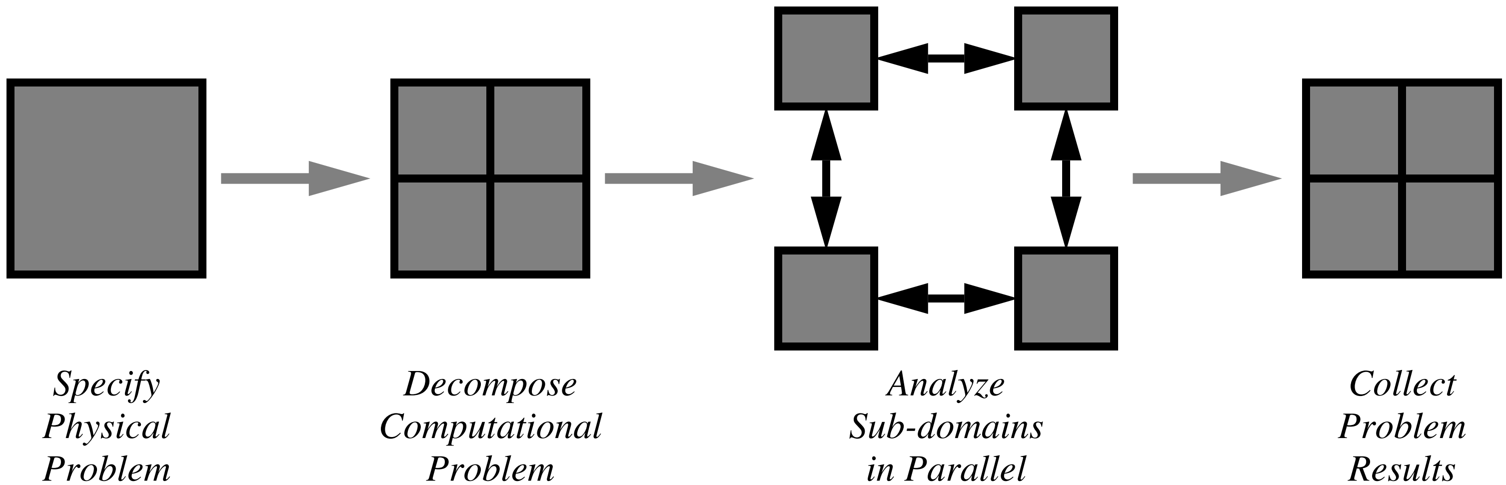
- P4 — from Argonne (version 1.2)
- PVM — from ORNL, Parallel Virtual Machines (version 2.4 now, 3.1 soon)

Parallel processing using loosely-coupled machines & message-passing



Parallel Programming

Model for Parallel Programming



Parallel Software Development

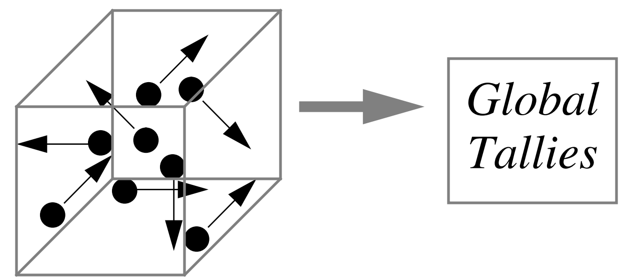
- High-Level: — break algorithm into independent tasks
- Low-Level: — sort-out local vs. global data
  — nearly all coding is re-usable
  — message-passing for control & global data

Parallel Algorithm Examples

VIM Monte Carlo

Physical Problem

- 3-D geometry, general & arbitrary
- Continuous-energy physics
- Used extensively for benchmarking
  (ANL, Studsvik, Comm. Ed., ...)

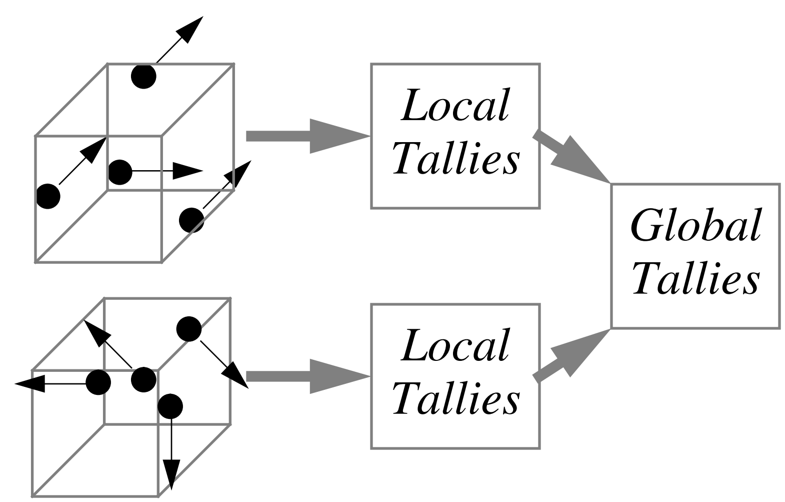


Conventional Solution Algorithm

- Random walk for neutrons
- Tally events of interest

Parallel Algorithm

- Dispatch neutrons to different processors
- Independent random walks
- Local tallies on each processor
- Combine local tallies into global results



Status

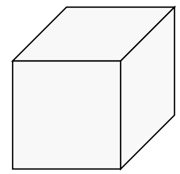
- Parallel development in progress,
  production use expected soon

Parallel Algorithm Examples

DIF3D Diffusion Theory

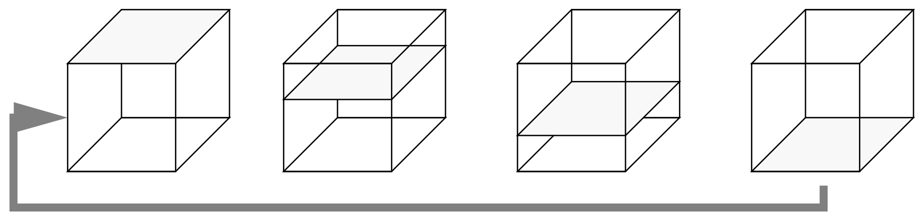
Physical Problem

- 3-D mesh, diffusion theory
- Finite-difference or nodal
- Workhorse for reactor physics



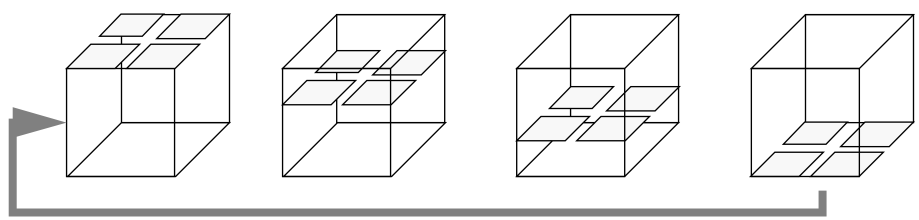
Conventional Solution Algorithm

- Decompose into planes
- Solve planes 1-at-a-time,
  sweep over planes



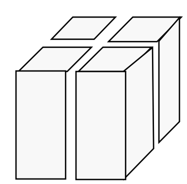
Simple Parallel Algorithm

- Decompose planes
- Solve sub-domains in parallel
  exchange boundary values



Advanced Parallel Algorithm

- Decompose into 3-D blocks
- Minimizes communications



Status

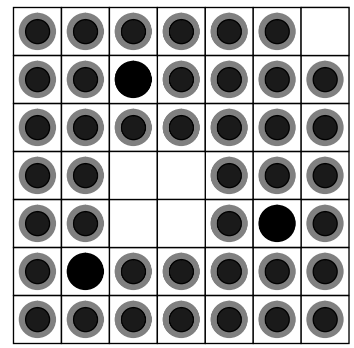
- Planning

Parallel Algorithm Examples

GTRAN2 — Assembly Transport Code

Physical Problem

- 2-D / 3-D, *exact* assembly geometry
- Transport theory, using collision probabilities



Parallel Algorithm

- Ray-tracing to compute CP's — perfectly parallel
- Global solution — similar to DIF3D

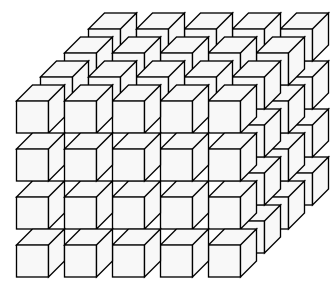
Status

- In testing — promising, near-linear speedup

VARIANT — Variational Nodal Transport

Physical Problem

- 3-D nodal transport theory



Parallel Algorithm

- Compute response-matrices — perfectly parallel
- Global solution — similar to DIF3D

Status

- Planning

Status

All computing performed on RA Network

RA Division computer codes — well-known, verified, supported

- VIM Monte Carlo
- DIF3D diffusion
- VARIANT nodal transport
- MC2, REBUS, RTC, ...
- SASSYS plant analysis
- COMMIX fluids
- POLYFAIL code (gas-tagging)
- etc, etc, etc

DQS

- 3,000 jobs/month using DQS batch
- effective scheduling & load-balancing for batch work
- negligible impact on interactive use

P4 & PVM

- Parallel processing support — "Virtual Supercomputer"
- Development progressing on some codes, planned for others