Cluster Computing in a College of Criminal Justice

Boris Bondarenko

and

Douglas E. Salane

Mathematics & Computer Science Dept.

John Jay College of Criminal Justice

The City University of New York

2004 USENIX Annual Technical Conference

Boston, MA

July 2, 2004

Outline

- Importance of cluster computing (HPC) in a college whose focus is criminal justice and public administration
- Cluster computing projects in progress and planned (research and instruction)
- Issues that arise in building and managing clusters in organizations with limited resources and staff
- Cluster, Linux, and open source developments

Institutional Background
John Jay College/CUNY

- **College:** Specialized Liberal Arts College within CUNY (13,000 students including 2000 graduate students).
- **Degrees:** Law and Police Science, Public Management, Fire Science, Security, Forensic Science, Computer Information Systems, M.S. in Forensic Computing (2004), Ph.D. in Criminal Justice.
- **Mission:** Advance the practice of criminal justice and public administration through research and by providing a professional workforce.

High Performance Computing at John Jay College I

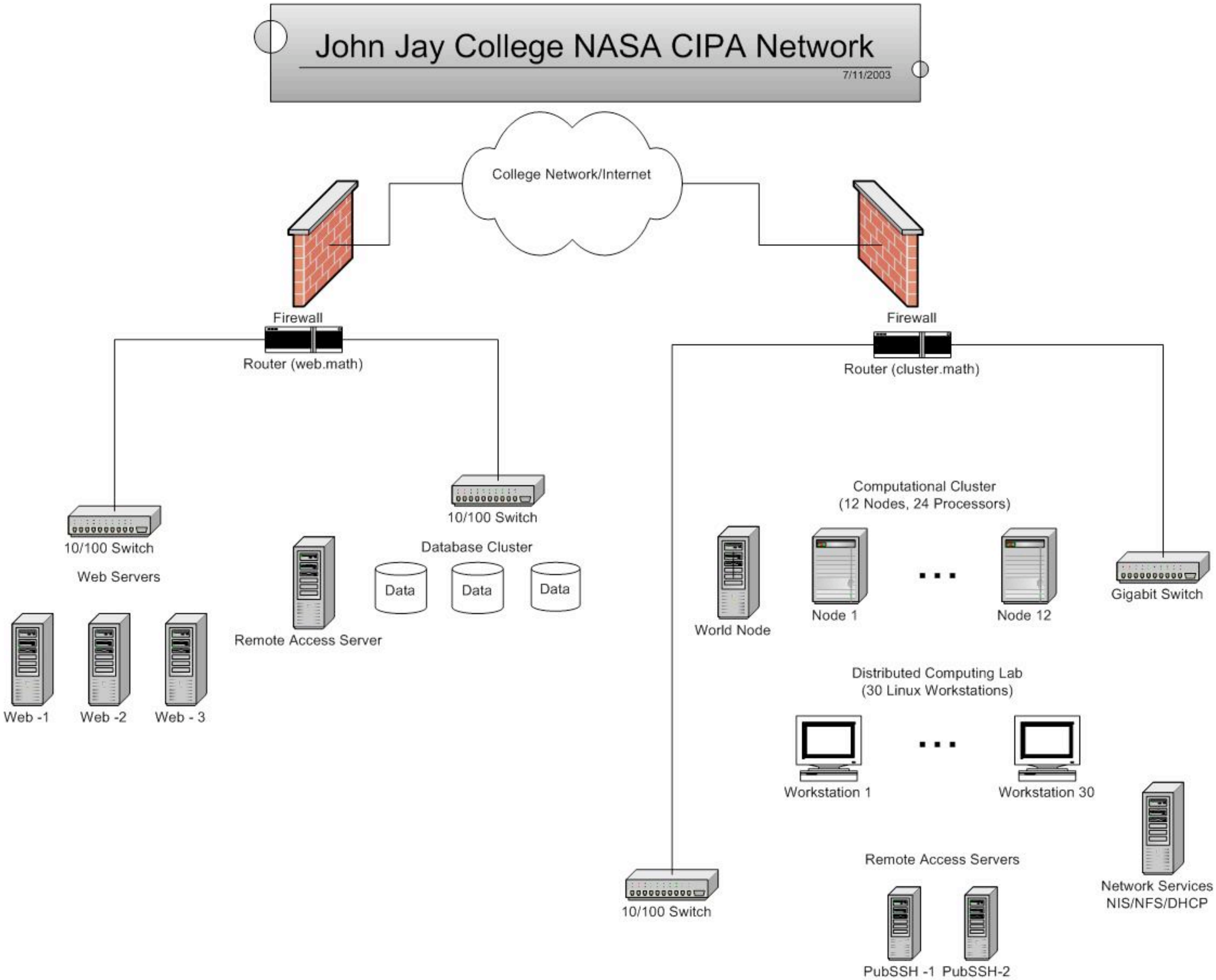
- Fire standards and codes for buildings (Computational Fluid Dynamics - NIST Fire Dynamics Simulator and Smoke View)
- Latent Semantic Indexing (Principal Component Analysis – Singular Value Decomposition)
- Toxicology (molecular modeling – Gaussian)
- FBI’s National Incident-Based Reporting System (NIBRS – database analysis and data mining)

High Performance Computing at John Jay College II

- Aircraft control systems (Parallel computation of Schur Form for rapid solution of Riccati Equation)
- Research and Instruction in mathematical software (ScaLAPACK, HPL Benchmark)
- Instruction in systems areas of computing, parallel algorithms, and distributed algorithms (NASA CIPA)
- Password Cracking (Teracrack SDSC)

Cluster Computing Facilities

- **Computational Cluster (Beowulf Cluster):** worldnode, 12 compute nodes (24 Pentium IV XEON (1.8 and 2.4 GHz processors, 1 GB RAM, 512K L2 cache), 20 GB local disk, Gigabit Ethernet, MPICH over TCP/IP, NFS File server, Linux 2.4.20-8smp
- **Database Cluster:** 4 nodes - remote access server, web server, Microsoft SQL and Oracle 10g
- **Distributed Computing Laboratory:** Computing Laboratory with 30 Linux Workstations (partnership with Science Dept.)


John Jay College NASA CIPA Network
7/11/2003
College Network/Internet
Firewall
Router (web.math)
Firewall
Router (cluster.math)
10/100 Switch
10/100 Switch
Web Servers
Database Cluster
Data
Data
Data
Remote Access Server
Web -1
Web -2
Web - 3
Computational Cluster
(12 Nodes, 24 Processors)
World Node
Node 1
Node 12
Gigabit Switch
Distributed Computing Lab
(30 Linux Workstations)
Workstation 1
Workstation 30
Remote Access Servers
Network Services
NIS/NFS/DHCP
10/100 Switch
PubSSH -1
PubSSH-2

Cluster Design Considerations I

- **Architecture** Vendor supported blade/rack system or pile of PCs

- **Cluster Software** cluster distribution software (OSCAR - ORNL, NPAIC ROCKS - SDSC, or Scyld Beowulf) vs. self-configuration (Kickstart+ shell scripts)

- **File System** NFS; Andrew; GFS – Sistina Systems; Lustre – CFS, Inc.; PVS – ANL, GPFS - IBM

Cluster Design Considerations II

- **Interconnect** Gigabit Ethernet, Myrinet, Quadrics, InfiniBand

- **Message passing** MPICH over TCP/IP

- **Monitoring** Ganglia UC Berkeley, Supermon - LANL, direct console access

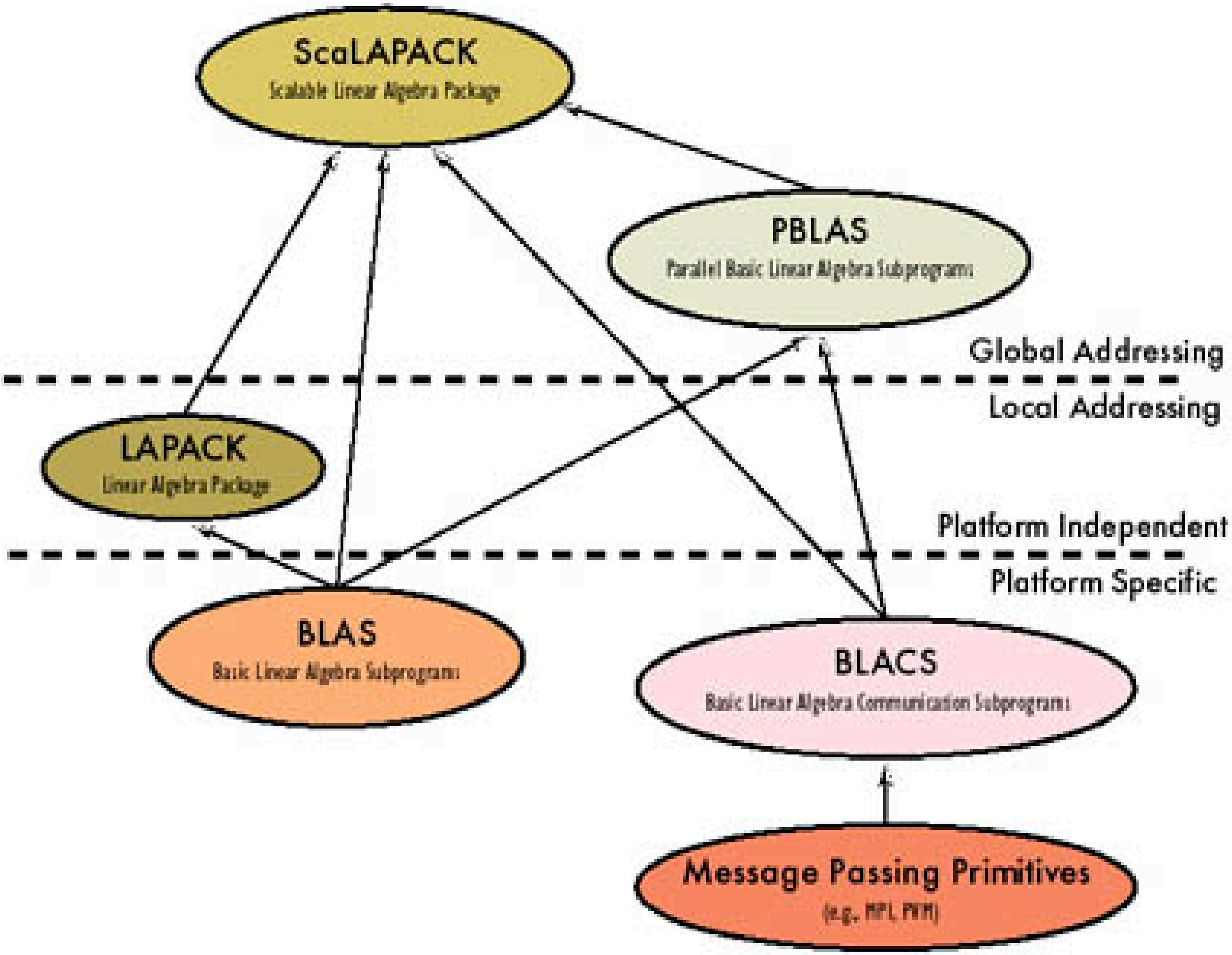
- **Testing** Netpipe – AMES Laboratory, BLACS, MPI Testers

ScaLAPACK

- Dense matrix computations in a distributed memory environment (clusters and MPP machines)
- Linear systems, least squares, eigenvalues, matrix decompositions (e.g., LU, QR, SVD)
- Reliable software with good error reporting facilities
- Not easy to use. User must write code to distribute the matrix over the process grid. User must set algorithmic parameters (e.g., block size, process array dimensions)

ScaLAPACK

A Software Library for Linear Algebra Computations on Distributed-Memory Computers



Basic Linear Algebra Communications Subroutines (BLACS)

- Setup/teardown process topologies (Array of processes most common)

- Point-to-point & broadcast send/receive of rectangular and trapezoidal matrices

- Miscellaneous routines (e.g., barrier, matrix element wise sum, max and min)

- Test routines to ensure reliable communications

Using BLACS to Detect Errors

- Broadcast testing routine: generates matrix on selected process, broadcasts it, receiving routines test for correct transmission.

- Process (0,1) reports errors, invalid element at A(12,16):

  Expected -.2417943949438026

  Received -.2417638773656776

Basic Linear Algebra Subroutines (BLAS)

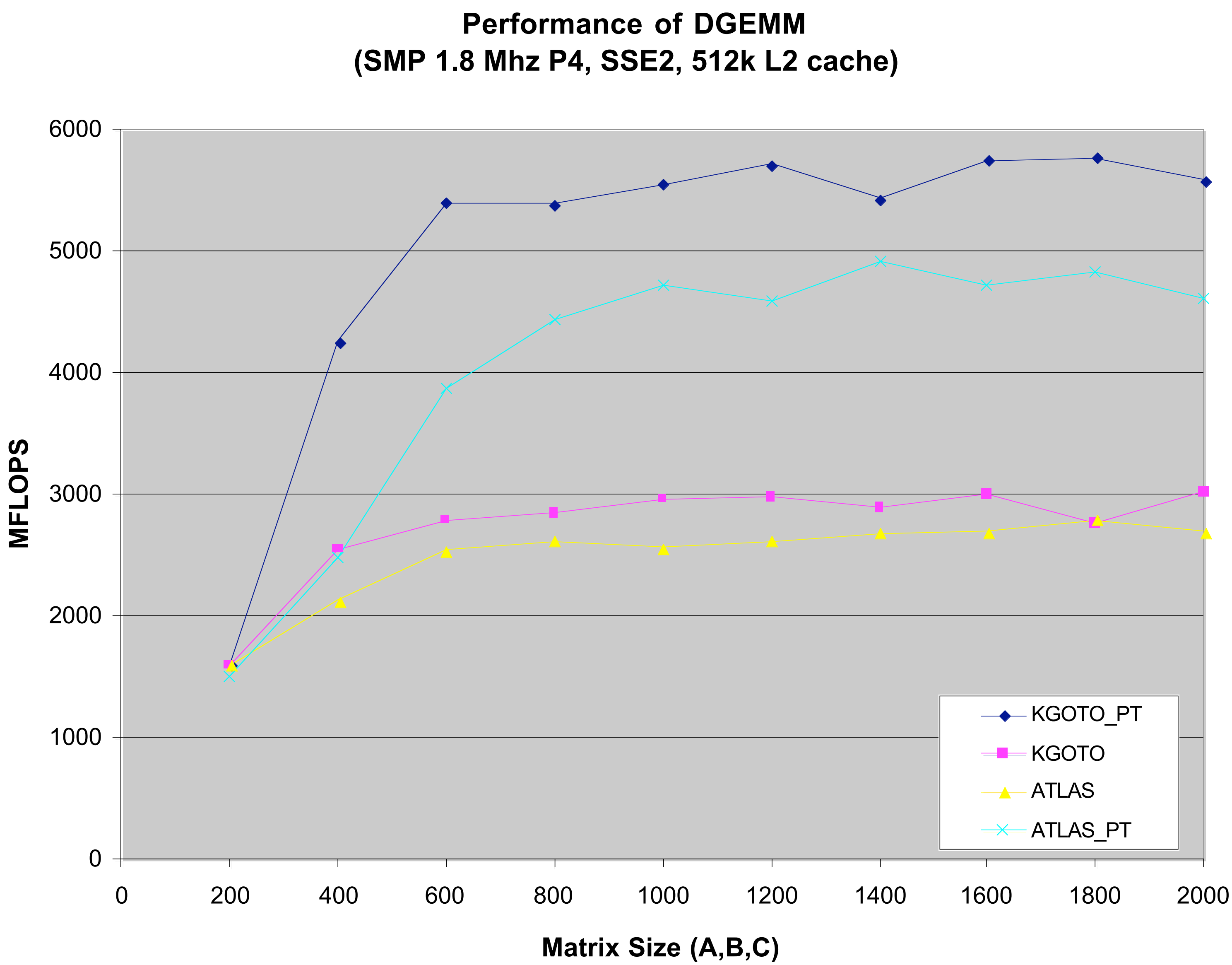
- Perform scalar, matrix vector and matrix matrix operations. Block algorithms to take advantage of memory hierarchies.
- Must be optimized for a specific processor.
- Three versions: Intel Math Kernel Library (MKL), ATLAS Generated, and KGoto. Multithreaded and single threaded versions.

BLAS matrix multiply routine DGEMM

- C = alpha*AB + beta*C,
  alpha and beta are scalars, A,B and C are matrices

- Critical for performance of many ScaLAPACK routines and HPL (e.g. HPL benchmark on Livermore MCR Cluster raised from 5.69 to 7.63 TFLOPS)

- Best results on Pentium IV: KGoto BLAS (special coding to minimize cache and TLB misses)


Performance of DGEMM
(SMP 1.8 Mhz P4, SSE2, 512k L2 cache)
MFLOPS
Matrix Size (A,B,C)
0
1000
2000
3000
4000
5000
6000
0
200
400
600
800
1000
1200
1400
1600
1800
2000
KGOTO_PT
KGOTO
ATLAS
ATLAS_PT

HPL Benchmark Results (from Top 500)

	R	Site	CPUs	R_{max}	R_{peak}
Earth Simulator	1	Japan	5,120	35,860	40,960
MCR Linux Network X	12	LLNL	2,304 2.4Ghz X	7,634	11,060
HP Alpha Server	67	NASA GSFC	1392	2,164	2,784
E*Trade X	499	E*Trade Financial	290 2.4Ghz X	634	1,392
JJ Cluster		John Jay	24 2.4 & 1.8Ghx X	47	98

R – rank in Top 500 Super Computers list
Rmax – Linpack Benchmark (GFLOPS)
Rpeak – Theoretical Highest Performance (GFLOPS)

FBI National Incident Based Reporting System (NIBRS)

- Develop an Oracle database version of NIBRS and make it available to criminal justice research community
- Support online analysis and data mining through a web portal
- Provide mechanism for automatic updates
- Employ cluster/grid computing to provide high throughput and availability

NIBRS

- Data warehouse: Oracle 10G database on Linux Red Hat AS 3 Server
- 13 segments (flat files), 6 Main segments (administrative/incident, offense, property, victim, offender, arrestee), largest 3.2 million records, 100 to 200 bytes per record, 39 reference tables
- 2000/2001 data 1.29 Gbyte, expect about 10 Gbyte for 1995 to present

Cluster Developments

- Single System Image (cluster monitoring, OS version skew, single process space)
- Commodity low latency interconnect technology that provides unified I/O (Remote Direct Memory Access, InfiniBand?)
- Nodes that consume less power
- Cluster applications that provide error checking

Collaborators

- **NIBRS** Peter Shenkin, Raul Cabrera, Atiqual Mondal, and Samra Vlasnovec; Math and Computer Science Dept.
- **Parallel Schur Decomposition** Mythilli Mantharam, Math and Computer Science Dept.
- **Fire and Smoke Simulation** Glenn Corbet, Fire Science Dept.
- **Molecular Modeling** Ann Marie Sapse and Robert Rothchild, Science Dept.

Contact Information

- Douglas E. Salane
  NASA CIPA Cluster Computing Project
  dsalane@jjay.cuny.edu
  web.math.jjay.cuny.edu

- Bibliography available

Credits

- NASA Curriculum Partnership Improvement Award
- Graduate Research and Technology Initiative of CUNY (01,02,03)
- Open Source and freely available software (Linux, GNU compilers and languages, Apache, PHP, Oracle Academic License)