Jack Dongarra  
University of Tennessee  
http://www.cs.utk.edu/~dongarra/  
http://icl.cs.utk.edu/

Innovative Computing Laboratory

- International Known Research Group
- Size: About 40 people
  - 15 students; 15 full time; 10 support
- Funding
  - NSF
    - Supercomputer Centers (NPACI & NCSA)
    - Next Generation Software (NGS)
    - Info Tech Res. (ITR)
    - Middleware Init. (NMI)
  - DOE
    - SciDAC
    - Math in Comp Sci (MICS)
- DOD
  - Modernization
- Work with companies
  - Microsoft, MathLab, Intel, Sun Microsystems, Myricom, HP
- PhD Dissertation, MS Project
- Equipment
  - A number of clusters
  - Desktop machines
  - Office setup
- Summer internships
  - Industry, ORNL, ...
- Travel to meetings
- Participate in publications

10/19/2002 9:07 AM
Four Thrust Research Areas

» Numerical Linear Algebra Algorithms and Software
  » EISPACK, LINPACK, BLAS, LAPACK, ScaLAPACK, PBLAS, Templates, ATLAS
  » Self Adapting Numerical Algorithms (SANS) Effort
    » LAPACK For Clusters
    » SALSA

» Heterogeneous Network Computing
  » PVM, MPI
  » FT-MPI, NetSolve

» Software Repositories
  » Netlib, NA-Digest
  » NHSE, RIB, NSDL

» Performance Evaluation
  » Linpack Benchmark, Top500, PAPI

Collaboration

» CS Department here at UTK
» Oak Ridge National Laboratory
» UC Berkeley/UC Davis
» UC Santa Barbara/UC San Diego
» Globus/ANL/ISI
» Salk Institute
» Danish Technical University/UNIC
» Monash University, Melbourne Australia
» Ecole Normal Superior, Lyon France
» ETHZ, Zurich Switzerland
» ETL, Tsukuba Japan
» Kasetsart U, Bangkok, Thailand
What Next?

» Jack -- Welcome
» Sudesh Agrawal-- NetSolve
» Kevin London -- PAPI
» Graham Fagg -- Harness/FT-MPI
» Asim Yarkhan -- GrADS
» Victor Eijkhout-- SANS
Introduction

» What is NetSolve
   » Is a research project started almost 6 yrs back.
   » NetSolve is a client-server system that enables users to solve complex scientific problems over the net.
   » It allows users to access both hardware and software computational resources distributed across the net.

How Does NetSolve Work?
Usability

- Easy access to software
  - Access standard and/or custom libraries.
  - No need to know internal details about the implementation.
  - Simple interface or API to access these libraries and software
- Easy access to hardware
  - Access to machines registered with NetSolve system.
  - User’s laptop can now access the power of super computers.
  - No need to worry about crashing user machine.
- User friendly interface to access the resources
  - C, Fortran interface
  - Matlab
  - Octave
  - Mathematica
  - Web

Features of NetSolve

- Asynchronous and Synchronous requests
- Sequencing
- Task Farming
- Fault Tolerance
- Dynamic addition and deletion of resources
- Pluggability with Condor-G
- Pluggability with NWS
- Pluggability with Globus
- Interface with IBP
Future plans

- NetSolve-E, which would be a revolutionary evolution of NetSolve.
- Client and Server can sit behind NATs and be able to talk to each other
- We would be able to incorporate different types of resources
- More dynamics would be added, to allow plug and play capability into the system.
- Resources would be able to come and go on the fly
- Many more......
- In short, a revolution is going to happen in a year or two 😊
- For more information contact us at NetSolve@cs.utk.edu

Final Note

Thanks
PAPI – A performance application programming interface

Kevin London

Overview of PAPI

» Performance Application Programming Interface
» The purpose of the PAPI project is to design, standardize and implement a portable and efficient API to access the hardware performance monitor counters found on most modern microprocessors
PAPI Implementation

Java Monitor GUI

Portable Layer

PAPI Low Level

PAPI High Level

Machine Specific Layer

PAPI Machine Dependant Substrate

Kernel Extensions

Operating System

Hardware Performance Counter

PAPI Staff

Current Staff Members
- Jack Dongarra
- Kevin London
- Philip Mucci
- Shirley Moore
- Keith Seymour
- Dan Terpstra
- Haihang You
- Min Zhou

Former Staff Members
- Qichao Dong
- Cricket Deane
- Nathan Garner
- George Ho
- Leelinda Parker
- Thomas Spencer
- Long Zhou
Tools currently using PAPI

» Deep/MPI
» Scalea
» SvPablo
» TAU
» Vprof
HARNESS & FT-MPI

Graham Fagg
320 Claxton
fagg@cs.utk.edu
http://icl.cs.utk.edu/harness

HARNESS = Heterogeneous Adaptable Reconfigurable Networked System

FT-MPI = Fault Tolerant MPI

HARNESS is a DOE funded, joint project with ORNL and Emory University.

UTK/ICL team, Edgar (soon), Graham, Tone.
Funding 3 years.
Whats HARNESS?

» Once upon a time.. We built s/w in a big block of modules. Each module did a different thing.. But they all got linked into a single executable.
  » Example PVM a message passing library.
  » So when we needed some new functionality we wrote the new code, and recompiled a new executable.

Whats HARNESS?

» HARNESS is a back-plane/skeleton
» Build parts as you need them, put them on a web repository or in a local directory.
» When you need something load them dynamically and then maybe throw them away...
» Think of kernel modules but for a distributed system that does parallel RPC and message passing.
» NOT JAVA, its faster C, C++, F90 etc
Whats FT-MPI

» MPI is the Message Passing Interface standard.
» FT-MPI is an implementation of that.
» But..
  » MPI programs were designed to live on reliable supercomputers.
  » Modern machines and clusters are made from many thousands of commodity CPUs.
  » MTBF_{total} = MTBF_{node} * number of nodes
    » MTBF_{total} < my large application simulating the weather
» In English, modern jobs on modern machines have a high chance of failure and as they get bigger it will just get worse...

What is FT-MPI

» FT-MPI extends MPI and allows applications to decide what to do when an error occurs:
  » restarting a failed node
  » continuing with a lesser number of nodes
  » Other MPI implementations either just abort everything OR they use check-pointing to "roll back" which is expensive.
Research stuff

- HARNESS
  - Distributed algorithms for coherency
  - Management of plug-ins
  - High speed parallel RPCs
- FT-MPI
  - Many2many [collective/group] communications, buffer management, new algorithms of numeric libraries
  - Fault state management
- Skills you would use:
  - networking (TCP/sockets), systems (threads/posix calls)

Contact info:

Graham Fagg
320 Claxton
Phone 974-5790

Email: fagg@cs.utk.edu
Web: http://icl.cs.utk.edu/harness
GrADS - Grid Application Development System

Jack Dongarra, Asim Yarkhan, Sathish Vadhiyar, Brett Ellis, Victor Eijkhout, Ken Roche

- Problem: Grid has distributed, heterogeneous, dynamic resources; how do we use them?
- Goal: reliable performance on dynamically changing resources
- Minimize work of preparing an application for Grid execution
  - Provide generic versions of key components (currently built in to applications or manually done)
    - E.g., scheduling, application launch, performance monitoring
  - Provide high-level programming tools to help automate application preparation
    - Performance modeler, mapper, binder
People in GrADS

- Principal Investigators
  - Francine Berman, UCSD
  - Andrew Chien, UCSD
  - Keith Cooper, Rice
  - Jack Dongarra, Tennessee
  - Ian Foster, Chicago
  - Dennis Gannon, Indiana
  - Lennart Johnsson, Houston
  - Ken Kennedy, Rice
  - Carl Kesselman, USC ISI
  - John Meller-Crummey, Rice
  - Dan Reed, UIUC
  - Linda Torczon, Rice
  - Rich Wolski, UCSB

- Other Contributors
  - Dave Angulo, Chicago
  - Henri Casanova, UCSD
  - Holly Dail, UCSD
  - Anshu Dasgupta, Rice
  - Sridhar Gullapalli, USC ISI
  - Charles Koelbel, Rice
  - Anirban Mandal, Rice
  - Gabriel Marin, Rice
  - Mark Mazina, Rice
  - Celso Mendes, UIUC
  - Otto Sievert, UCSD
  - Martin Swany, UCSB
  - Satish Vadhiyar, Tennessee
  - Asim YarKhan, Tennessee

GrADSoft Architecture

Program Preparation System
- Source Application
- Libraries
- Software Components
- Whole Program Compiler
- Configurable Object Program
- Performance Feedback

Execution Environment
- Real-time Performance Monitor
- Resource Negotiator
- Scheduler
- Binder
- Grid Runtime System
GrADS Program Execution System

COP
Perf Model
Mapper

Application
Manager
(one per app)

Scheduler
Resource
Negotiator

Binder
Dynamic Opt
Perf Mon Setup

Contract
Monitor

Launch

Grid
Resources
And
Services

GrADS Information
Repository

Application

Research Areas

- Automatically generating performance models (e.g. for ScaLAPACK) on Grid resources
- Evaluating Performance “Contracts”
- Near Optimal Scheduling (execution) on the Grid
- Rescheduling for changing resources
- Checkpointing and fault tolerance
- High-latency tolerant algorithms (SANS ideas)
- Porting applications/libraries to GrADS framework
- Developing generic GrADSoft interfaces (API’s)
How To Be A Mathematician In A CS Department And Still Have Fun

Victor Eijkhout
eijkhout@cs.utk.edu

We get \( D'_{ij} i a, G_{m_{ij}} \) \( f, \mu, H_{m_{ij}}, \) \( h, \) where \( a, g, h, \) satisfy the limit equations
\[ a = a - \sum_{i} b_{i}a_{i} \]
\[ g = g - \sum_{i} b_{i}g_{i} \]
\[ h = h - \sum_{i} b_{i}h_{i} \]

Now let \( p = \max_{i,j} \) \( g \) \( = 0 \) for \( f, \) \( c \leq p \) and \( k, \) \( = 0 \) for \( f, c \leq p \) Then
\[ a_{i} = \frac{\sum_{j} b_{i} - \sum_{j} b_{i}}{\sum_{j} b_{i}} \]
\[ g_{i} = \frac{\sum_{j} b_{i} - \sum_{j} b_{i}}{\sum_{j} b_{i}} \]
\[ h_{i} = \frac{\sum_{j} b_{i} - \sum_{j} b_{i}}{\sum_{j} b_{i}} \]

gives that
\[ \left( \sum_{i} b_{i} - \sum_{i} b_{i} \right) = \left( \sum_{i} b_{i} - \sum_{i} b_{i} \right) \left( \sum_{i} b_{i} - \sum_{i} b_{i} \right) \]
\[ = \left( \sqrt{\sum_{i} b_{i} - \sum_{i} b_{i}} \right) \left( \sqrt{\sum_{i} b_{i} - \sum_{i} b_{i}} \right) \]

If \( \alpha \) is increased by a small amount, the line which is positive increases.
From the recursion formula it follows that \( \alpha \) will also increase, whereas all \( a, c, \) and \( h, \) will decrease. Hence both factors of the ratio will decrease as their product increases and they are of equal sign they must both be positive.

9.3 Some elementary estimates for Toeplitz matrices

In the case \( A = \text{conv} (m_{i}) \) the coefficients introduced in (36.2.1) are readily estimated. From that we estimate \( a_{i} \)
\[ a_{i} = a_{i} = \sum_{j} b_{i}a_{i} + \sum_{j} b_{i}a_{i} = a_{i} - a_{i} \]
\[ = a_{i} - a_{i} \]
The SALSA Project

» Self-Adaptive Linear Solver Architecture
» Traditional approach: user picks library routine, calls.
   » All decisions up to user
» Need for intelligent middleware to assist the user in picking the best library call
   » One extreme: use as black box
   » Less extreme: the user supplies hints, wishes, annotations
» Intelligence is developed over time: feedback of results into a database
   » Tuning of heuristics.

To Contact Us:

» Send email to dongarra@cs.utk.edu
» http://icl.cs.utk.edu/