Emerging Directions for High Performance Computing Software & Tools Research and Development

Dr. Frederica Darema

DARPA

Presented externally at the *DSWA Workshop on HPC Requirements* Feb 18-20, 1998

Outline

- **Motivation**
  - **Application trends**
  - **Computing platform trends**
- **Overview of Existing Application Support SW**
  - **Language, Compilers, Tools, Libraries**
- **New Technology Needed (NOTE: This portion of the presentation is included under the title: A Distributed Computing Support Environment)**
  - **application programming technology**
  - **application composition technology**
  - **system analysis technology**
- **Summary**

Application Directions

Past

- Mostly monolithic
- Mostly one programming language
- Computation Intensive
- Batch
- Hours/days

Present / Future

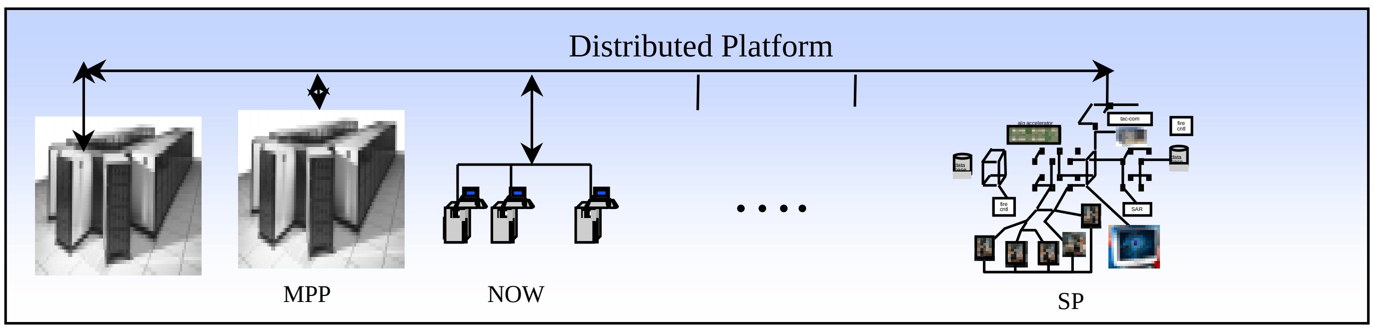
- Multi-Modular
- Multi-Language
- Multiple Developers
- Computation Intensive
- Data Intensive
- Real Time
- Few Minutes/hours
- Visualization (real time)
- Interactive Steering

Platform Directions

Past
- Vector Processors
- SIMD MPPs

Present
- Distributed Memory MPs
- Shared Memory MPs

Future
- Distributed Computers, Heterogeneous Platforms
  - Heterogeneity
    - architecture
    - node power (supernodes, PCs)
  - Latencies
    - variable (internode, intranode)
  - Bandwidths
    - different for different links
    - different based on traffic



Systems Software/Hardware Architectural Framework

Applications/Users — **Application**

Languages
Compilers
Libraries
Tools

Visualization
Collaboration Environments
Scalable I/O Data Management Archiving/Retrieval Services
Authentication/ Authorization Dependability Services
Other Services . . .

API & Runtime Services

Systems Management (OS) — **Global Management**

Distributed, Heterogeneous, DynamicComputing Platforms and Networks — **Computing Engine**

Memory Technology | *CPU Technology* | *Device Technology* | . . . — **Components Technology**

Language & Compiler Technology

USER/ APPLICATION

PROGRAMMING ENVIRONMENT

LIBRARIES

TOOLS

COMPILERS

OPERATING SYSTEM

HARDWARE

Programming Languages/Environments

Based on SPMD Model

- **HPF (Fortran D, CM-Fortran, Vienna Fortran)**
- **MPI (PVM)**
- **HPC++, CC++(C*, data parallel C, pC++)**

Language & Compiler Technology

Compiler Technology

- data independence
- task independence
- program transformation
- interprocedural analysis
- data locality/blocking
- techniques for latency tolerance & management
- run-time compilation
- inspection-execution approach

Industry Efforts

- Portland Group, IBM, APR, PSR

Compiler Research Example

Matrix Multiplication

Without Blocking

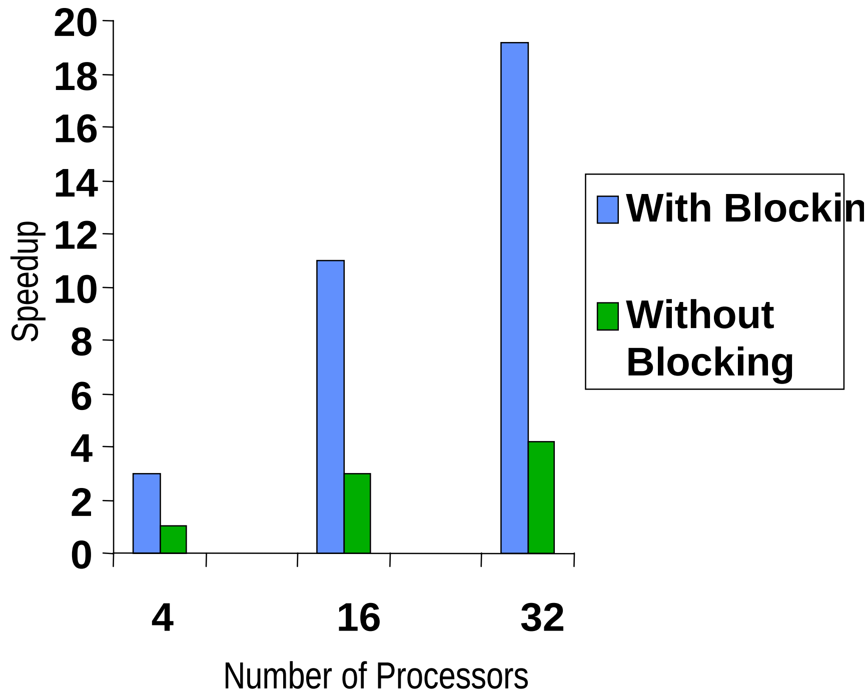
1 Processor code

```
for (i=0; i<N; i++) {
    for (k=0; k<N; k++) {
        r =X[i,k];
        for (j=0; j<N; j++)
            Z[i,j]=Z[i,j] + r*Y[k,j];
    }
```

With Blocking

1 Processor code

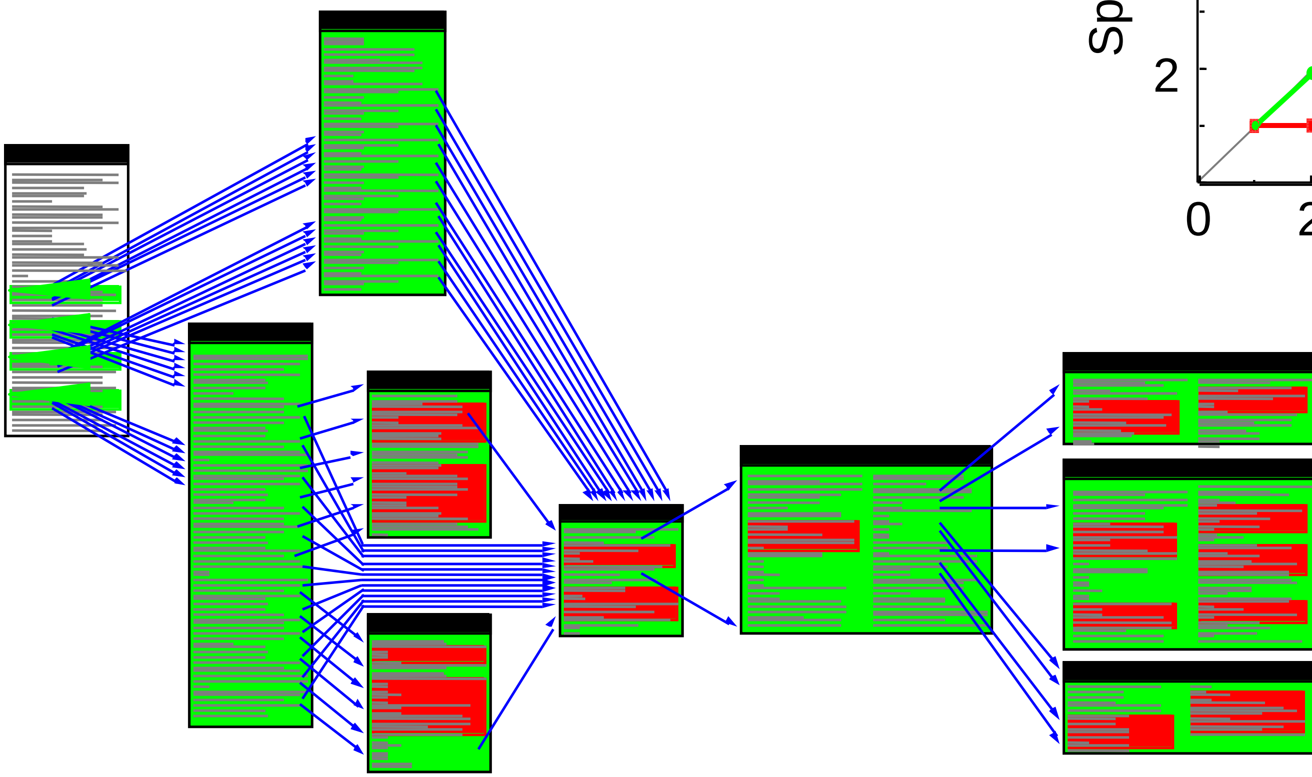
```
for (kk=0; k<N; k+=B)
    for (jj=0; jj<N; jj++=B)
        for (i=0; i<N; i++)
            for (k=kk; k<min(kk+B-1,N); k++) {
            r =X[i,k];
            for (j=jj; j<min(jj+B-1,N); j++)
                Z[i,j]=Z[i,j] + r*Y[k,j];
        }
```

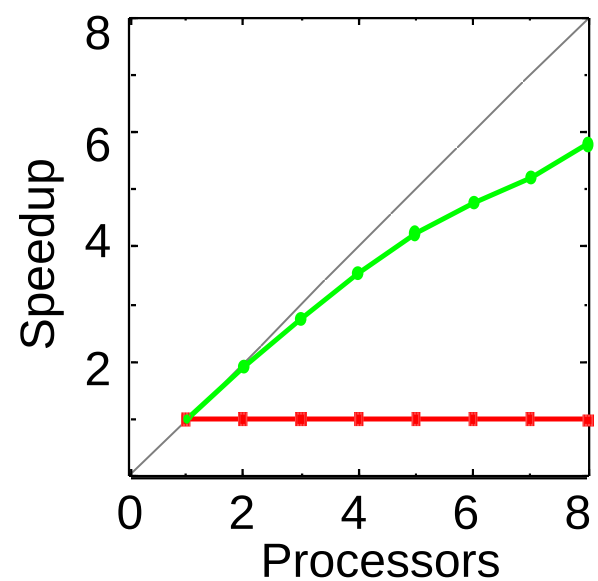


TURB3D: Whole Program Analysis
Turbulence Simulation

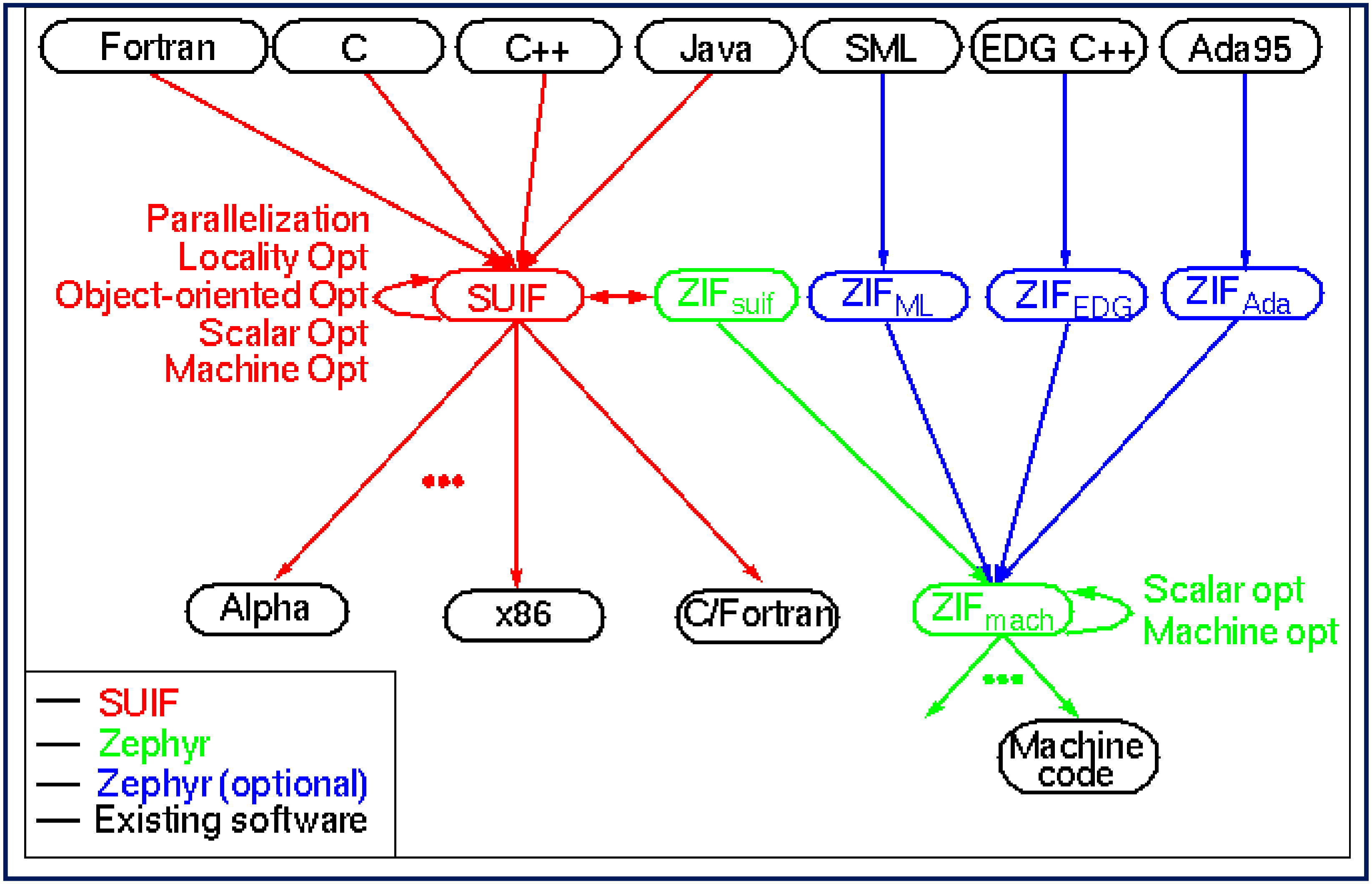
Parallelized Regions

- without interprocedural analysis
- with interprocedural analysis





National Compiler Infrastructure (NCI)



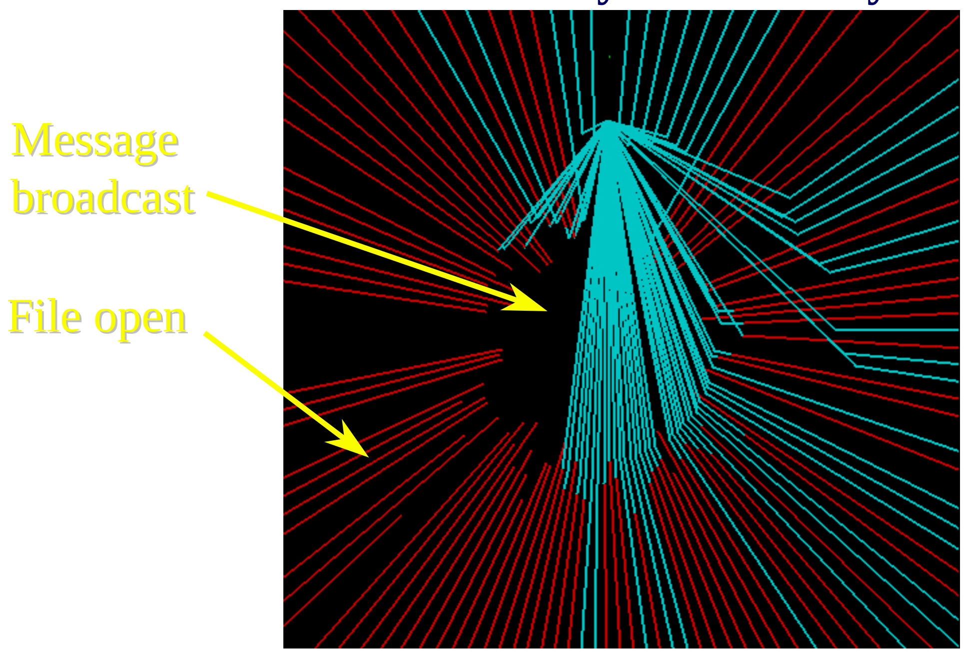
Performance Tools

- Emerging applications are
  - Dynamic, with adaptive behavior
  - Distributed, network-based computations
  - Long-lived, with evolving characteristics
- Very difficult to understand execution behaviors
  - New generation of tools needed
  - Deep compiler/tool integration
  - Real-time adaptive resource control
  - Direct manipulation via virtual environments
  - Multi-level analysis (hardware and software)

Performance and Debugging Tools

- **Debugging Tools**
  - FORGExplorer (Applied Parallel Research)
  - Total View (Dolphin Inc.)
  - Mantis (Split-C parallel debugger)
  - …
- **Performance Monitoring/Display Tools**
  - Pablo
  - Paradyn
  - AIMS
  - Nupshot (Upshot)
  - VAMPIRtrace
  - VT
  - …

Example of Pablo Display
Hartree-Fock Chemistry Code I/O Dynamics



Libraries

USER/ APPLICATION
PROGRAMMING ENVIRONMENT
LIBRARIES
TOOLS
COMPILERS
OPERATING SYSTEM
HARDWARE

Increase Performance:

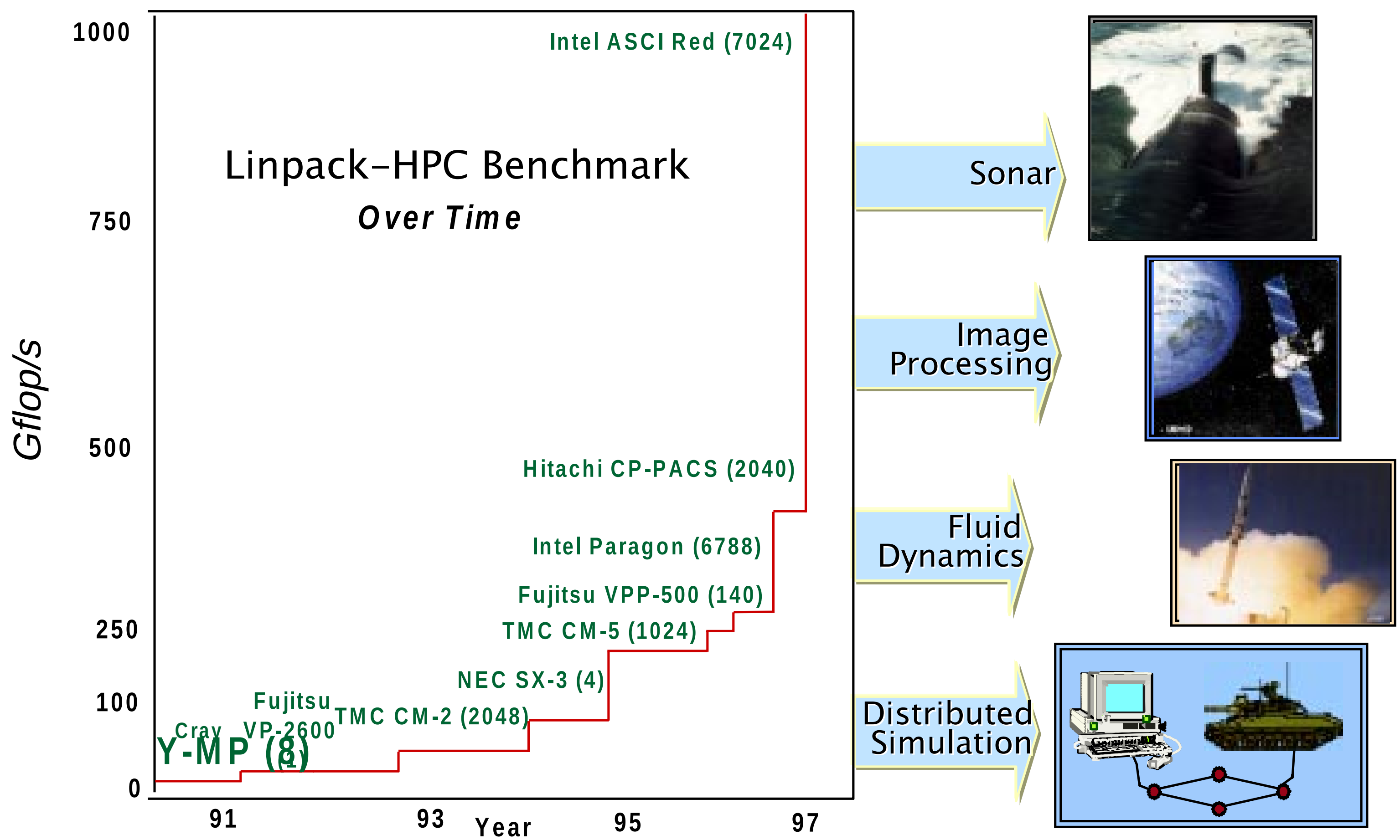
- Highly optimized architecture specific versions for common codes (BLAS)
- Lower barriers hide implementation so that not everyone has to be a specialist
- Increase portability (vendors implement standard libraries)

Numerical Example:

BLAS → loops
LINPACK → 1st generation algorithms
LAPACK → 3rd generation algorithms
SCALAPACK → parallel algorithms

Examples of Library Performance

Radar Cross-Section Modeling Benchmark



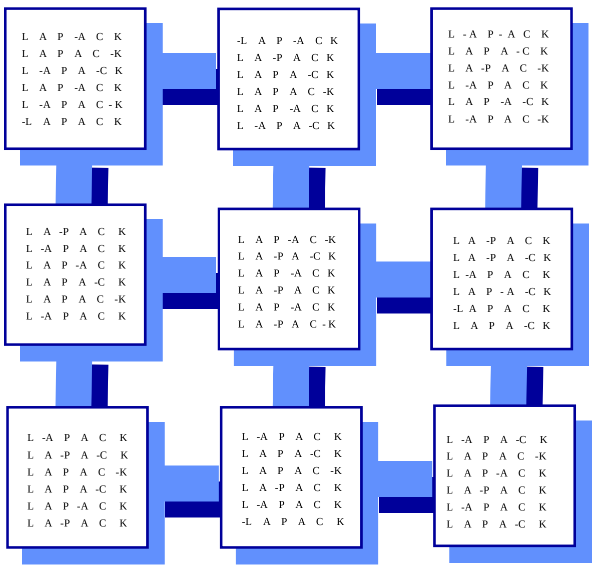
Examples of Libraries

- **LINPACK, EISPACK**
- **LAPACK**
- **ScaLAPACK**
  - **linear solvers, dense & sparse**
  - **eigen solvers, dense & sparse**
- **FFT-PACK**
- **P_ARPACK**
- **//ELLPACK**
- **NetLib**
- **PETSc**

ScaLAPACK and Distributed Memory Environment

ScaLAPACK: A Portable Linear Algebra Library For Distributed Memory Computers

Design Issues and Performance



Goal: Port LAPACK to distributed-memory environments

- Efficiency
  - Optimized compute and communication engines
  - Block-partitioned algorithms (Level 3 BLAS) utilize memory hierarchy and yield good node performance
- Scalability
  - As the problem size and number of processors grow
- Reliability
  - Whenever possible, use LAPACK algorithms and error bounds
- Portability
  - Isolate machine dependencies to BLAS and the BLACS
- Flexibility
  - Modularity: Build rich set of linear algebra tools: BLAS, BLACS, PBLAS
- Ease-of-Use
  - Calling interface similar to LAPACK

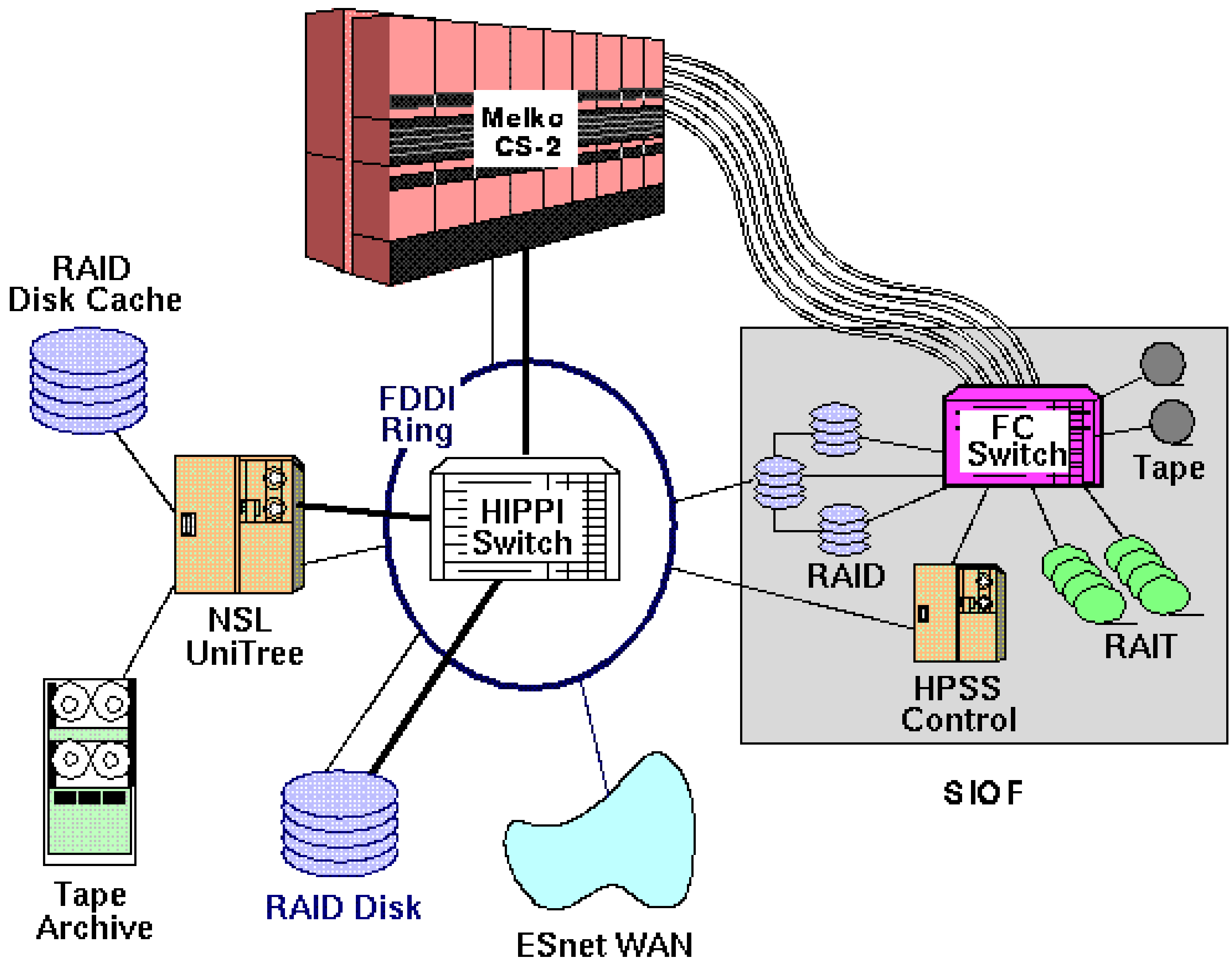
Many of these goals have been achieved through the promotion of standards for computation (BLAS, PBLAS) and communication (PVM, MPI)

Data Management

- **File Systems**
  - **SIO, MPI-I/O**
  - **AFS**
  - **DFS, NFS, Portable Parallel File System**
- **… but …**
  - **Flat files are not sufficient**
  - **Standard DBMS technology not sufficient**

  - **… important technology area, needs research and development**

Unitree Archival Data Storage System



Who's Doing What?

Development/Industry

- System Vendors
  - robust, implementations for standard computer languages
- Independent S/W Vendors
  - aggressive optimization front ends, customized back-ends

Research/Agencies

- NSF
  - few uncontrolled, unmanaged grants
- DARPA
  - large, focused efforts
- DOE, NASA, NSA, NOAA, EPA
  - Mission Driven

Performance Modeling