High Performance Mathematical Libraries (MLIB) for Itanium™ 2 Clusters

Hsin-Ying Lin, Kevin Wadleigh
Leonardo Borges

Math Library (MLIB) Team
HPTCD
Hewlett Packard Company
June 25, 2003

invent

Introduction

- Target hardware: Itanium 2 processors and their systems
  - Explicitly Parallel Instruction Computing (EPIC)
- Improve performance of key algorithms by using HP MLIB
  - Functionality
  - Usages
  - Performance comparison PA-RISC vs. Itanium™ 2
    - Architecture features used to design high performance algorithms
  - Performance results
    - BLAS, FFT, LAPACK, ScaLAPACK and SuperLU_DIST
- Summary

Today's Architecture Challenges:

- Memory Latency
- Branch Misdirects

- Too Few Registers
- Hardware-Based Instruction Scheduling

- Memory Addressing Efficiency
- Hardware, I/O Capacity

So, What is Itanium™ 2?

Most Significant Architecture Since 80386

- 64-Bit Architecture (Post RISC, 32-Bit)
- Explicitly Parallel Instruction Computing (EPIC)
- Comprehensive Predication
- Enhanced Speculation

Architecture

- EPIC (Explicitly Parallel Instruction set Computing) contains features for high performance including
  - Instruction level parallelism: instruction bundling, templates and dispersal
  - Large register files: 128 general and 128 floating point registers
  - Rotating registers: software pipelining uses register rotation. Acts like short vectors with each iteration of a loop, data in rotation registers moves to the next register in the set.
  - Instruction predication: removes branches. Predicated instructions are either normally executed or they do not affect the architectural state.
  - Advanced and speculative loads: data is loaded from addresses regardless of whether or not the data will be used, the address will be written to in the meantime, or the address is known to be valid.

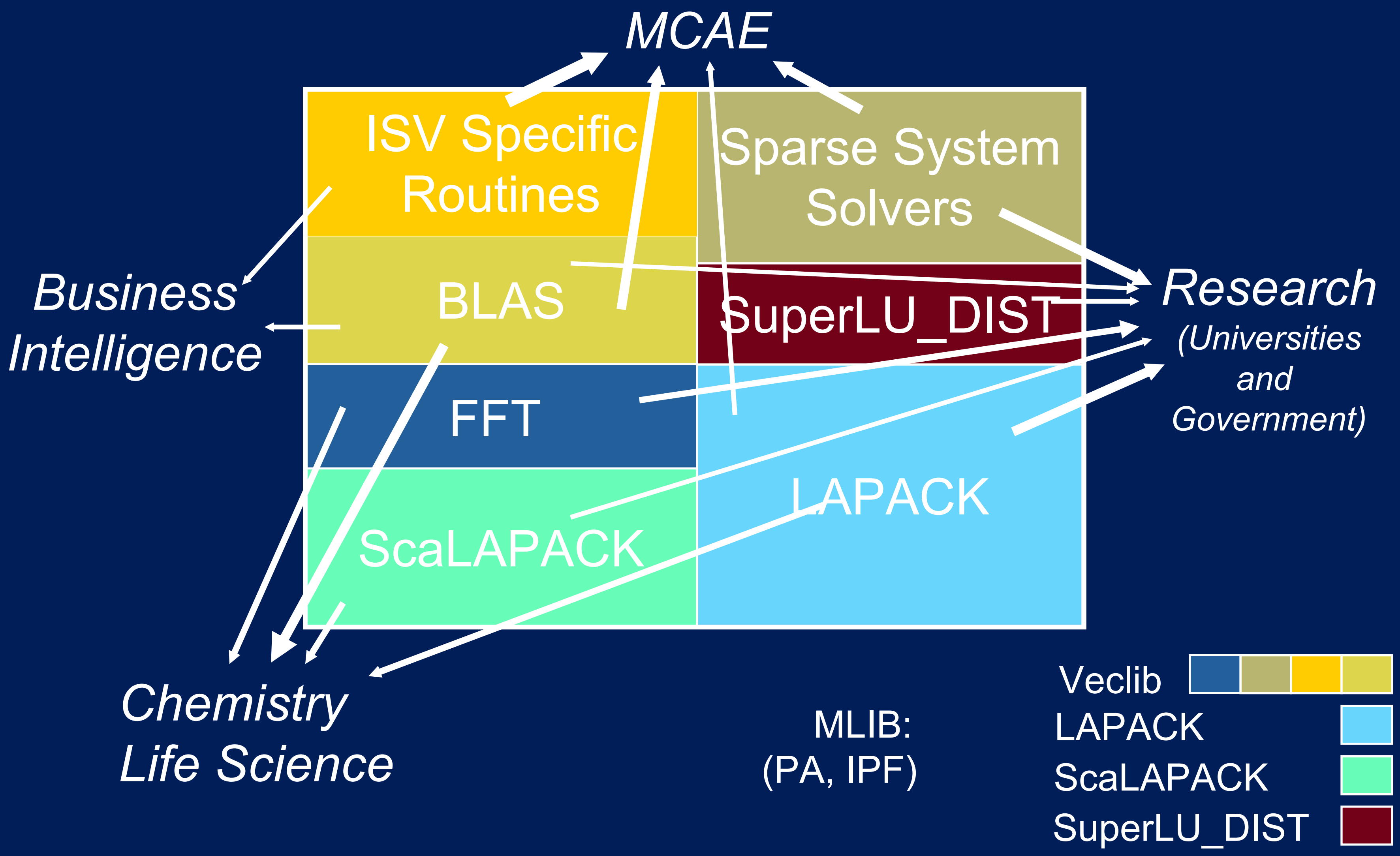
Performance Libraries

http://www.hp.com/go/mlib

HP MLIB (Math LIBrary)

- HP's Mathematical Software Product – MLIB
- Collection of subroutines providing frequently used mathematical software for scientific and technical computing
- Optimized for high performance on PA-RISC 2.0 and IPF processors
- Includes 32-bit addressing, 64-bit addressing, 64-bit addressing with 64-bit integers
- Includes shared and archive libraries
- Evaluation copies of MLIB available at http://www.hp.com/go/MLIB
- Documentation is also available at the web site
- Contact: mlibcore@rsn.hp.com

How MLIB Products Are Used



VECLIB Contents

- Legacy BLAS – Basic Linear Algebra Subprograms
  - Level 1 – vector operations
  - Level 2 – matrix-vector operations
  - Level 3 – matrix-matrix operations
- Sparse Linear Algebra
  - Sparse BLAS
  - Real symmetric sparse linear equations
  - Real symmetric sparse eigenvalue problems
- FFTs
  - 1,2,3-dimensional, simultaneous 1-d
  - Radix-2,3,5
  - Real-to-complex and complex-to-complex
- Convolutions
- Selected custom routines from ISVs

BLAS, FFT focus

- Highly optimized BLAS-1,2,3 implementations
- FFTs – novel algorithms created to utilize features of PA and IPF instruction set architectures and systems
- The most important computational kernels used by BLAS and FFTs are written in assembly code
- Many ISVs use MLIB BLAS and FFT routines

Real symmetric sparse solver focus

- Based on ORNL sparse symmetric solver
- Supports MMD (Multiple Minimum Degree) and METIS (multilevel nested dissection) orderings
- Computational kernel routines highly optimized
- Thread parallelism for improved performance
- Solver and/or orderings used by several ISVs

LAPACK V3.0 –
Linear Algebra PACKage

- Linear equations
- Least squares problems
- Eigenvalue problems
- Orthogonal and singular value decompositions
- Supercedes LINPACK, EISPACK
- Better designed than LINPACK, EISPACK since it is based on BLAS-3 routines
- HP’s LAPACK is more optimized than public domain software

ScaLAPACK 1.7 –
Scalable Linear Algebra PACKage

- High performance linear algebra library for distributed message passing computers
- It's LAPACK for clusters

Super_LU DIST 2.0 –
General Sparse Solver using MPI

- General sparse solver for linear algebra using message passing for parallelism developed by Xiaoye S. Li at NERSC-Lawrence Berkeley national Laboratory
- Designed for clustered computers
- First official release shipped in AR0902 (Summer 2002)
- Working on V2.0 currently

MLIB Parallelism

- SMP parallelism
  - *GEMM, *GEMV, *GER
  - Some LAPACK routines have higher parallelism that BLAS
  - Symmetric sparse linear equations
  - All FFTs
  - Convolution routines sconv, dconv

- Executables using VECLIB (always parallel enabled)

  efc foo.o –openmp –lveclib

  ecc foo.o –openmp –lveclib –lcl

  setenv MLIB_NUMBER_OF_THREADS #

  Default is one thread

MLIB Usage for HP-UX

- Non-parallel executable using VECLIB
  - f90 foo.o –lveclib
  - cc foo.o –lveclib –lcl –lm
  - aCC foo.o –lveclib –lcl –lm
  - To use 64-bit libraries link with +DD64 or +DA2.0W
- Parallel executable using VECLIB
  - f90 foo.o +O3 +Oparallel –lveclib
  - cc foo.o +O3 +Oopenmp –lveclib –lcl –lm
  - aCC foo.o +Oopenmp –lveclib –lcl –lm
  - setenv MLIB_NUMBER_OF_THREADS #
  - Default is one thread
- Same usage for LAPACK
- Compile/link Fortran routines with +i8 for 64-bit integers
- A few 64-bit address routines are dependent on f90 internals, so calling MLIB from C may require linking with f90
- Since the libraries exist in both archive and shared form, you should link with –Wl,-aarchive_shared for optimal performance

MLIB Usage for Linux

- Executables using VECLIB (always parallel enabled)
  - efc foo.o –openmp –lveclib
  - ecc foo.o –openmp –lveclib –lcl
  - setenv MLIB_NUMBER_OF_THREADS #
  - Default is one thread

- Same usage for LAPACK
- Compile/link Fortran routines with –i8 for 64-bit integers
- When you use the C compiler for linking you may need to link with some of -lPEPCF90 -lIEPCF90 -lF90

MLIB on Itanium™ 2-based

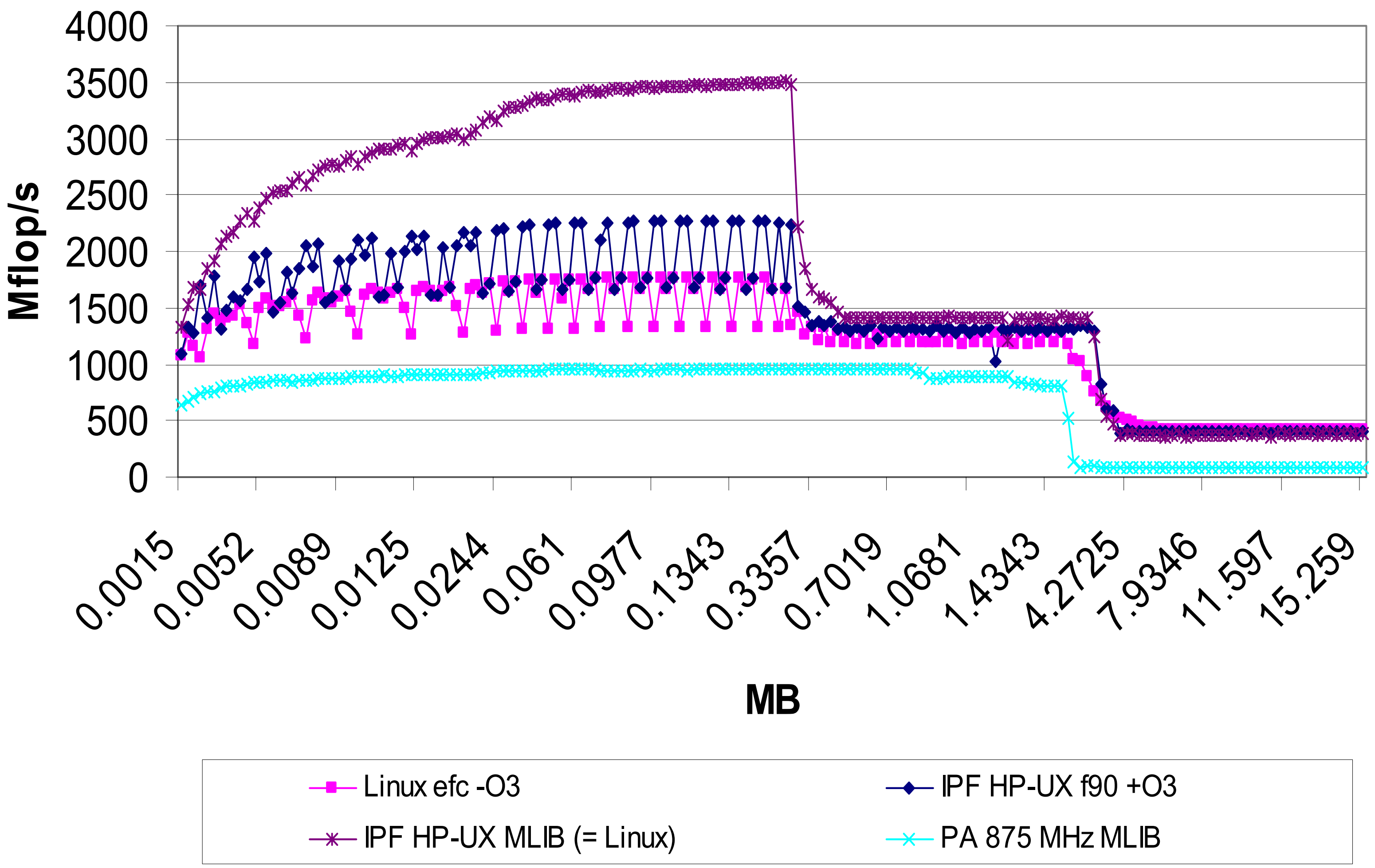
- Highly tuned library for Itanium 2 computers
- Contains all the functionality of VECLIB and LAPACK libraries
- SMP parallelism implemented using OpenMP
- Being used at PNNL, BP America, OSC.
- First release shipped in October 02 and is working on July 03 11.23 release

Performance Comparison

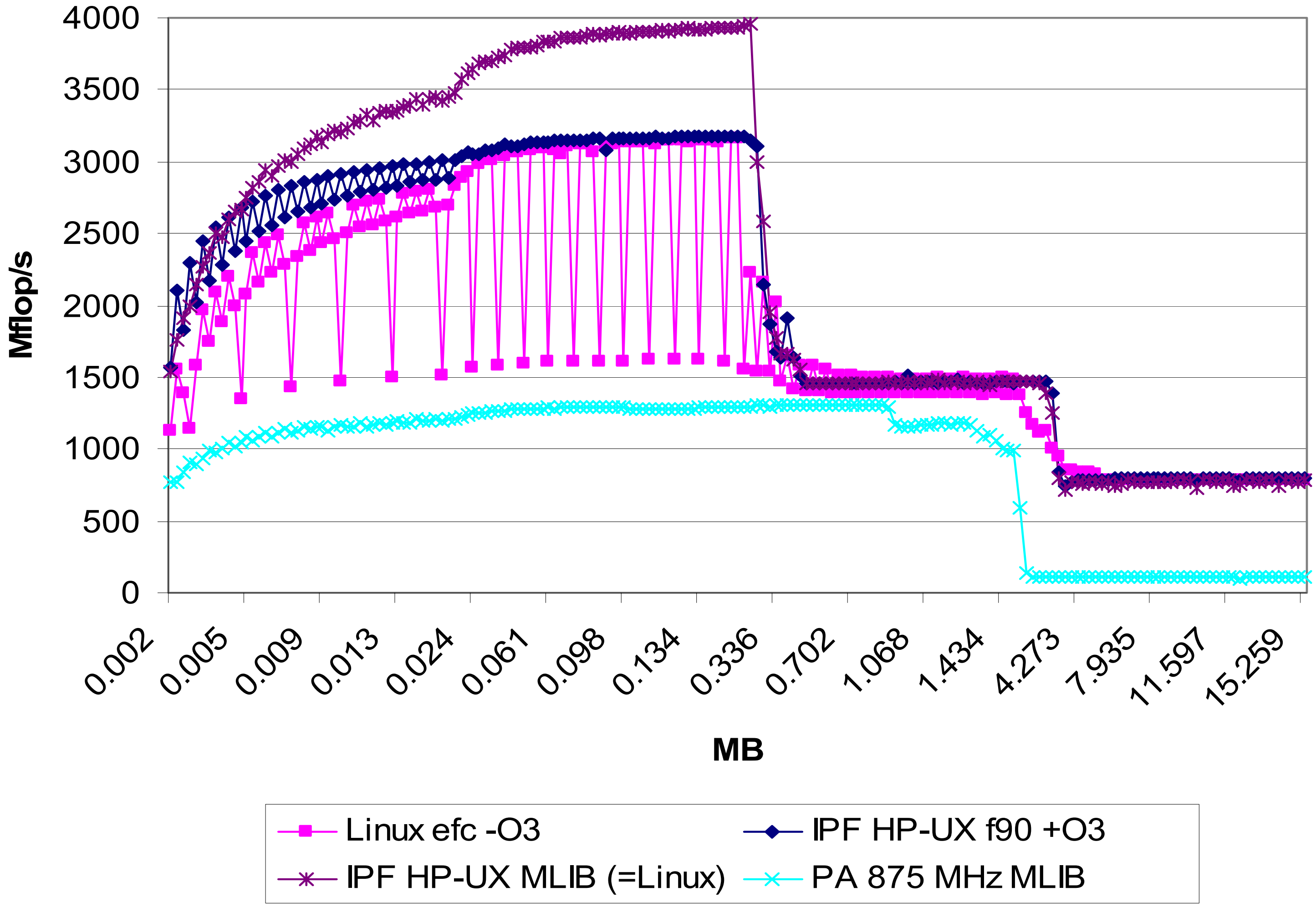
Processors Featured

Processor	Available	Clock (MHz)	64-bit theo. flt-pt peak (Gflop/s)	cache size
IPF Itanium 2	now	1000	4.0	L2/L3: 256 KB/3 MB
PA-8700+	now	875	3.5	1.5 MB

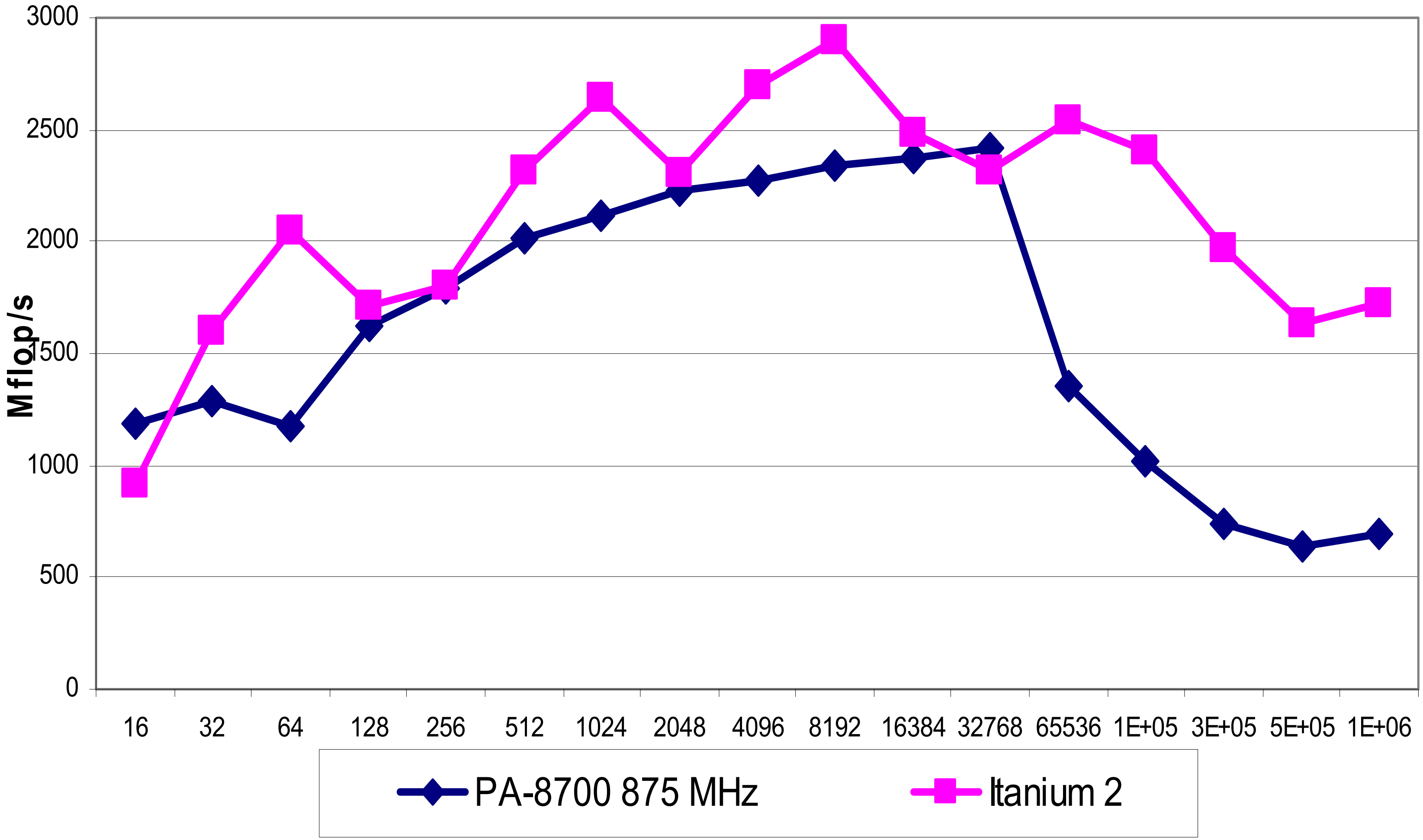
DAXPY performance



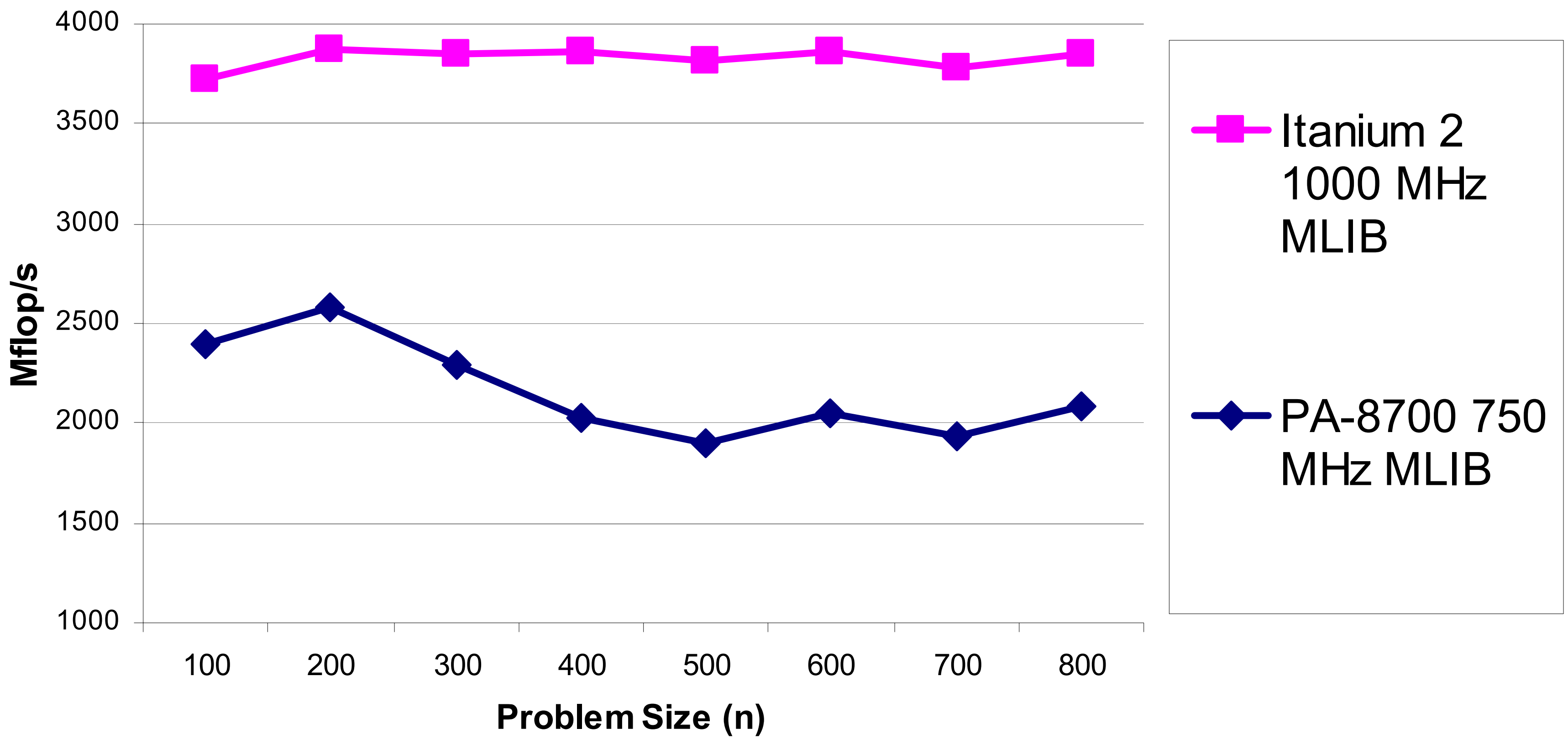
DDOT performance



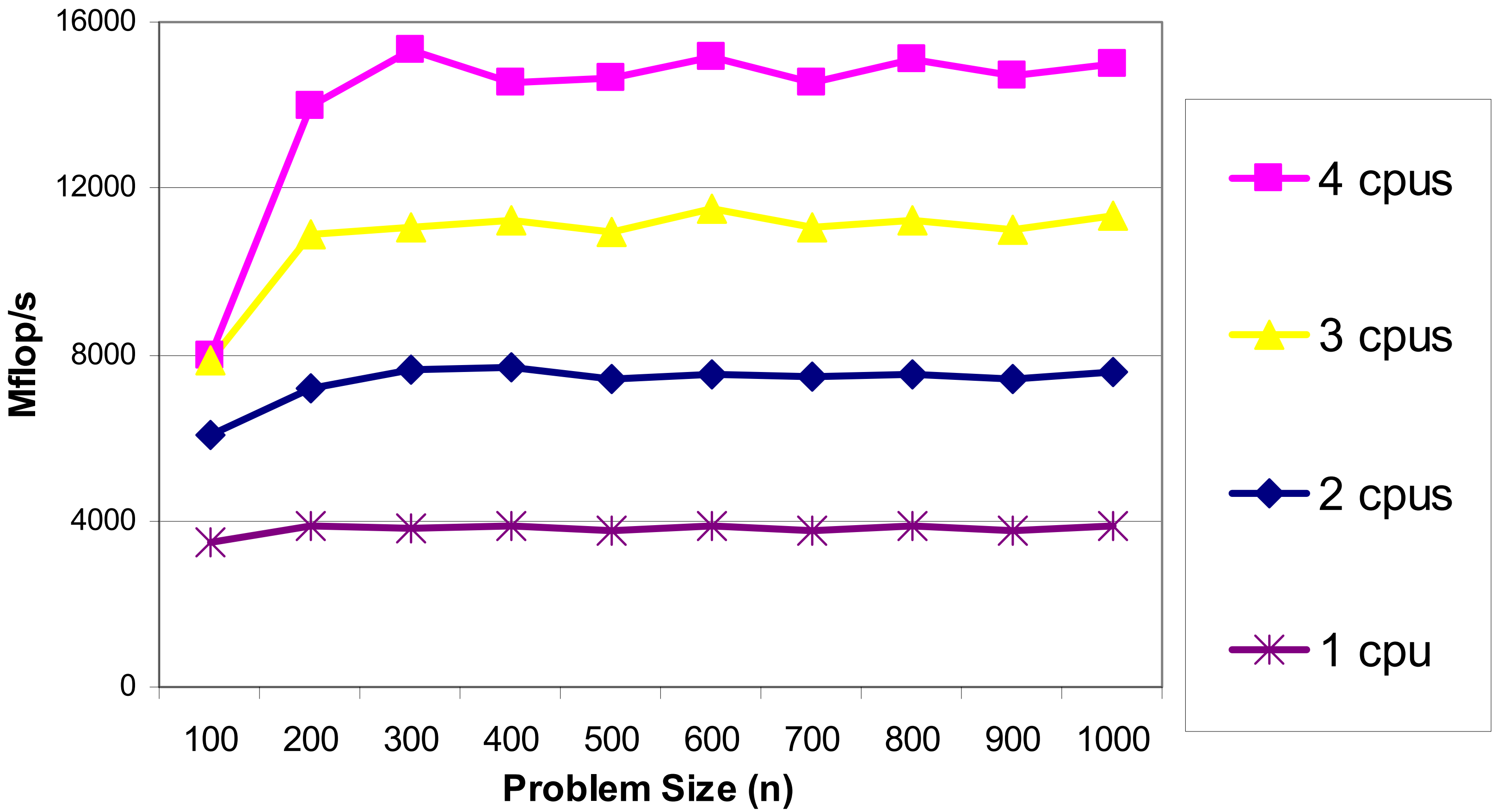
C1DFFT (1-d complex*8 FFT) Performance



DGEMM Performance

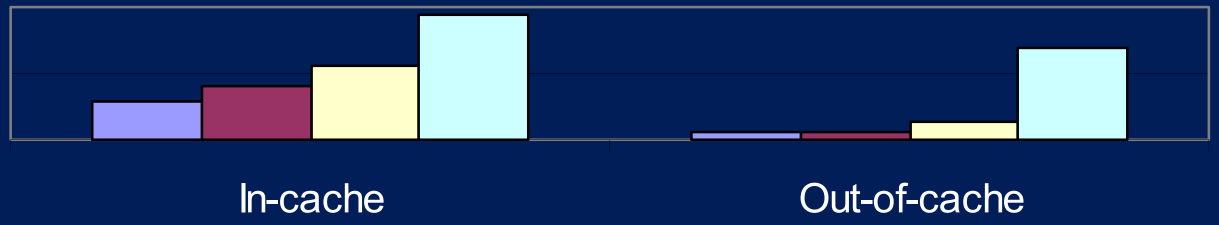


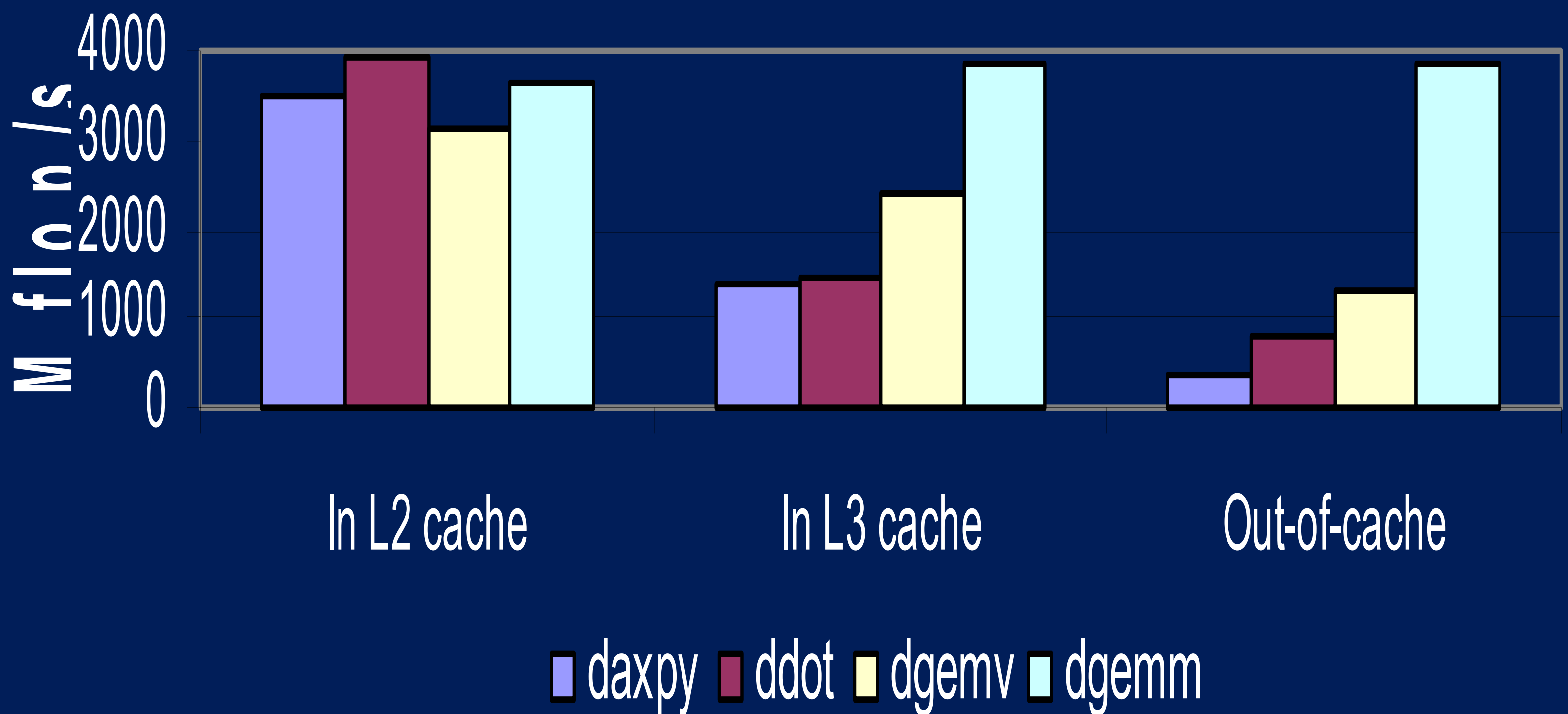
Itanium 2 DGEMM Performance – Linux



Effect of data reuse on Itanium 2 (1000 MHz) MLIB routines

PA-RISC results

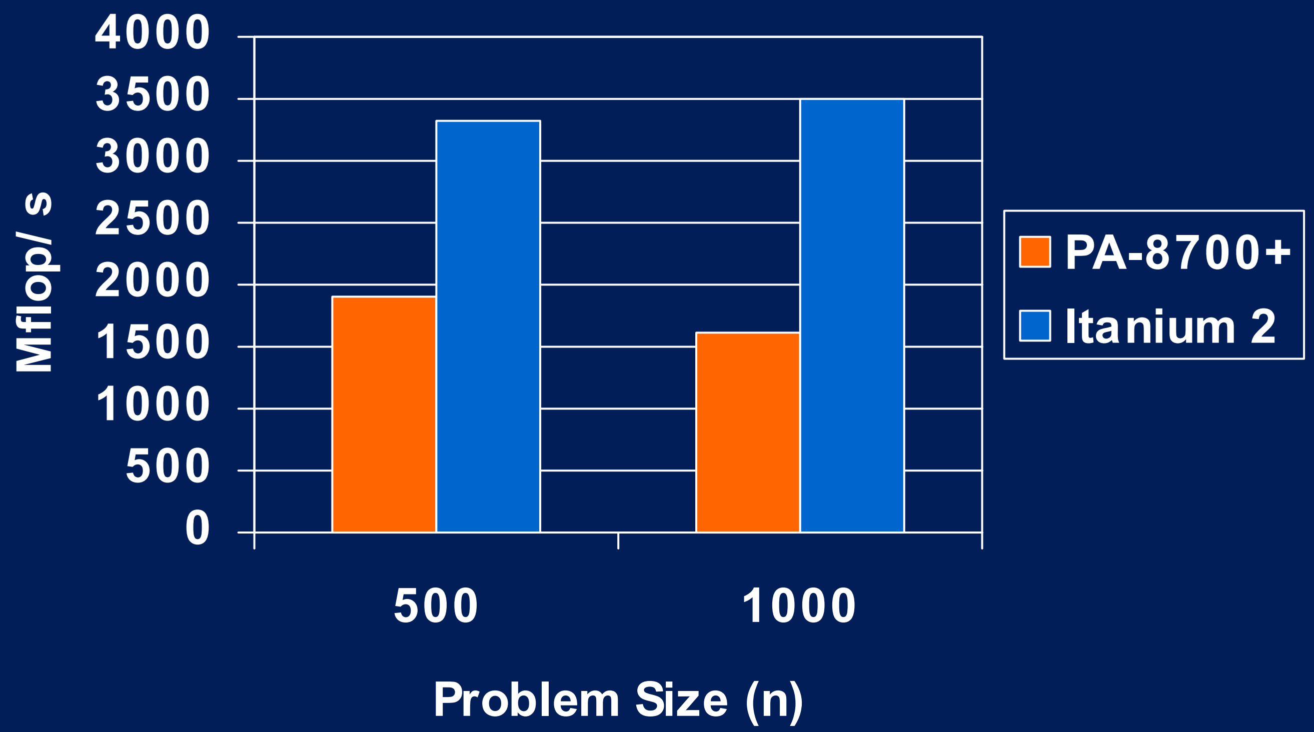




Architecture features used for high performance codes written using Assembly

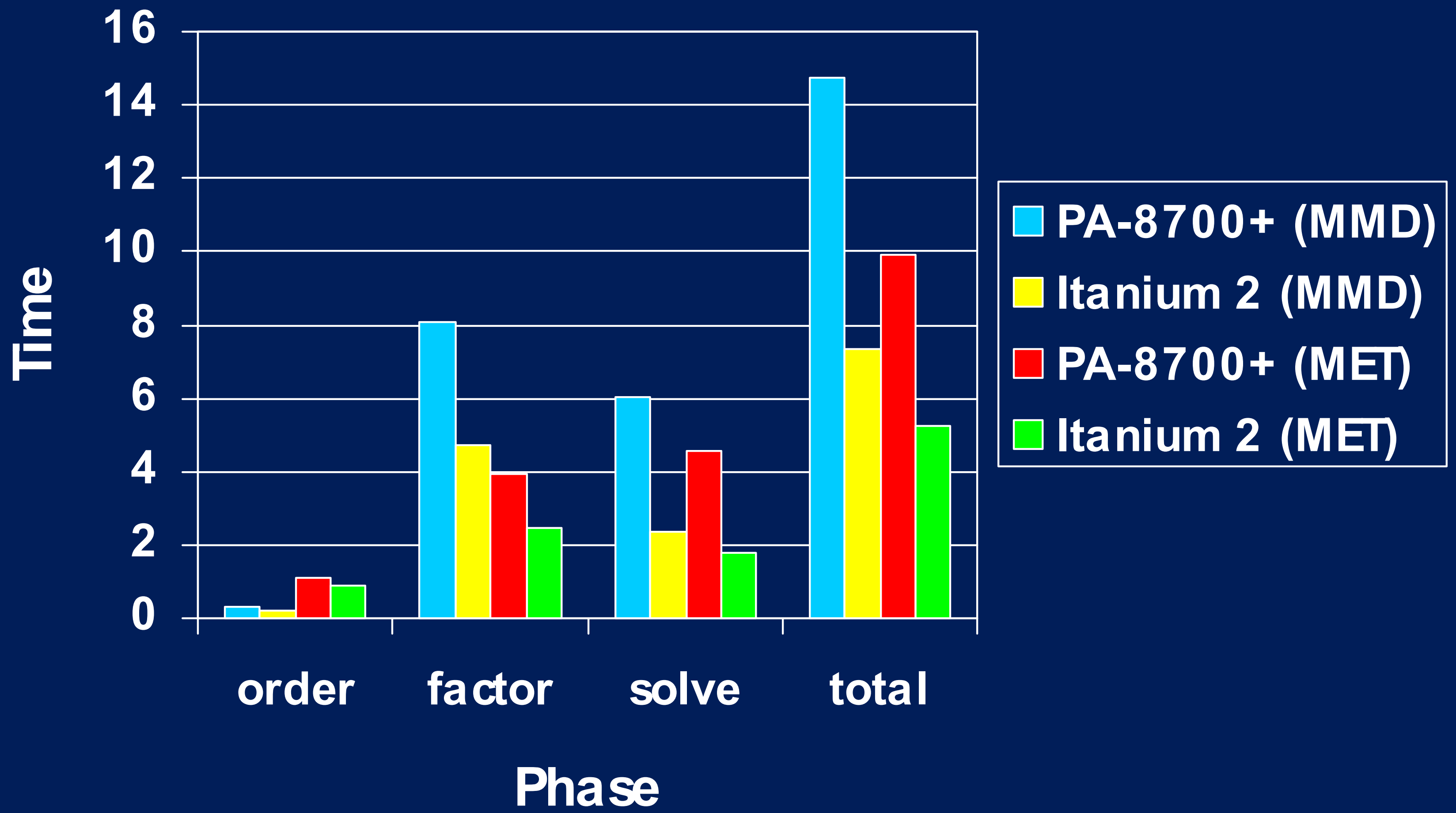
- Instruction level parallelism: instruction bundling, templates and dispersal
  - This enables MLIB to have high ILP.
- Large register files: 128 general and 128 floating point registers
  - This allows MLIB to unroll loops a lot more
- Rotating registers: software pipelining uses register rotation. Acts like short vectors: with each iteration of a loop, data in rotation registers moves to the next register in the set
  - This is not your father’s Vector Processor
- Quadwords load: read two double words using one instruction
  - This allows program to access twice amount of data per instruction. This is very useful when high cache bandwidth is needed.

LAPACK DGETRF (LU Factorization)

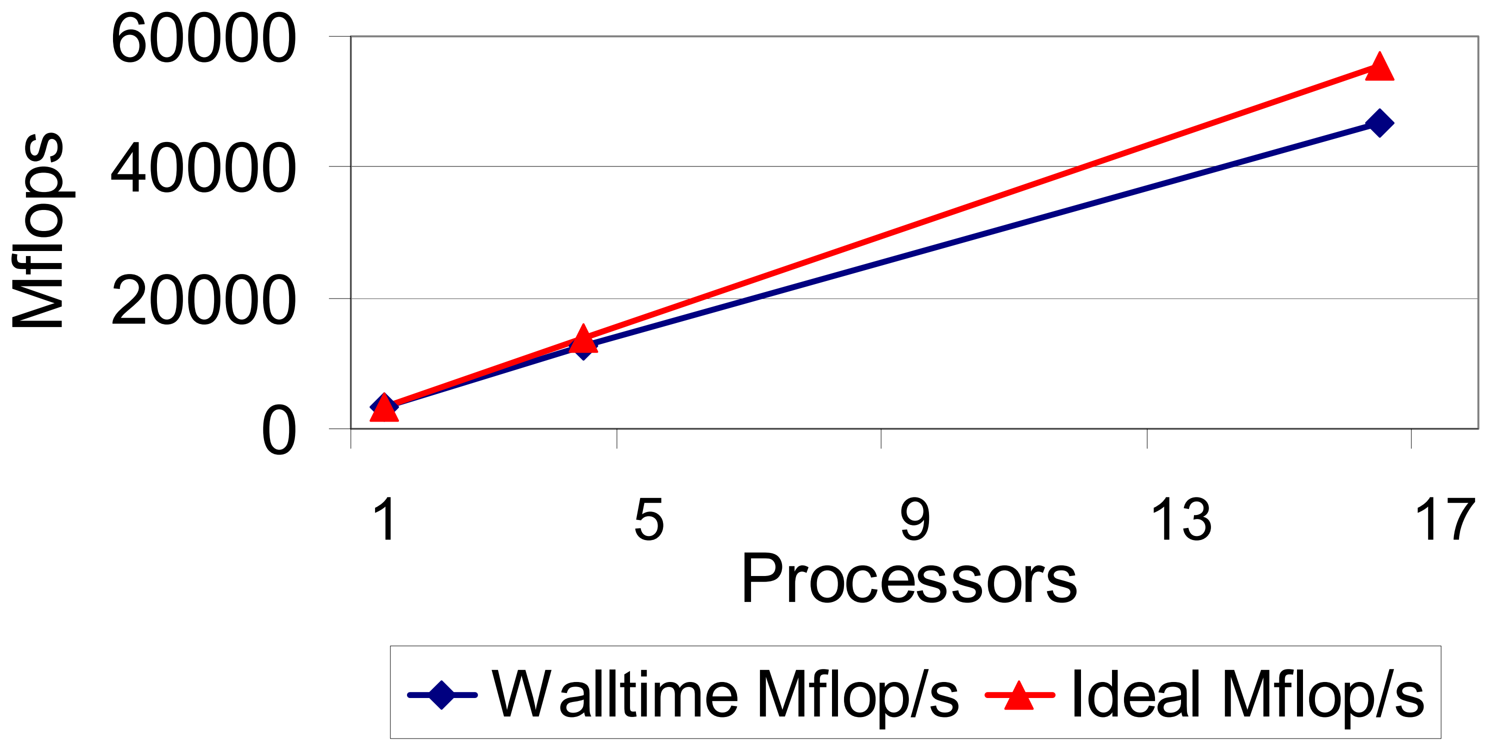


Sparse Solver (Times)

14012 eqns, 1052630 nonzeros, 50 rhs



PDGEMM Scalability using ScaLAPACK



SuperLU_DIST's Performance (Mflop/s) of Tuned version 1cpu vs. 4 cpus

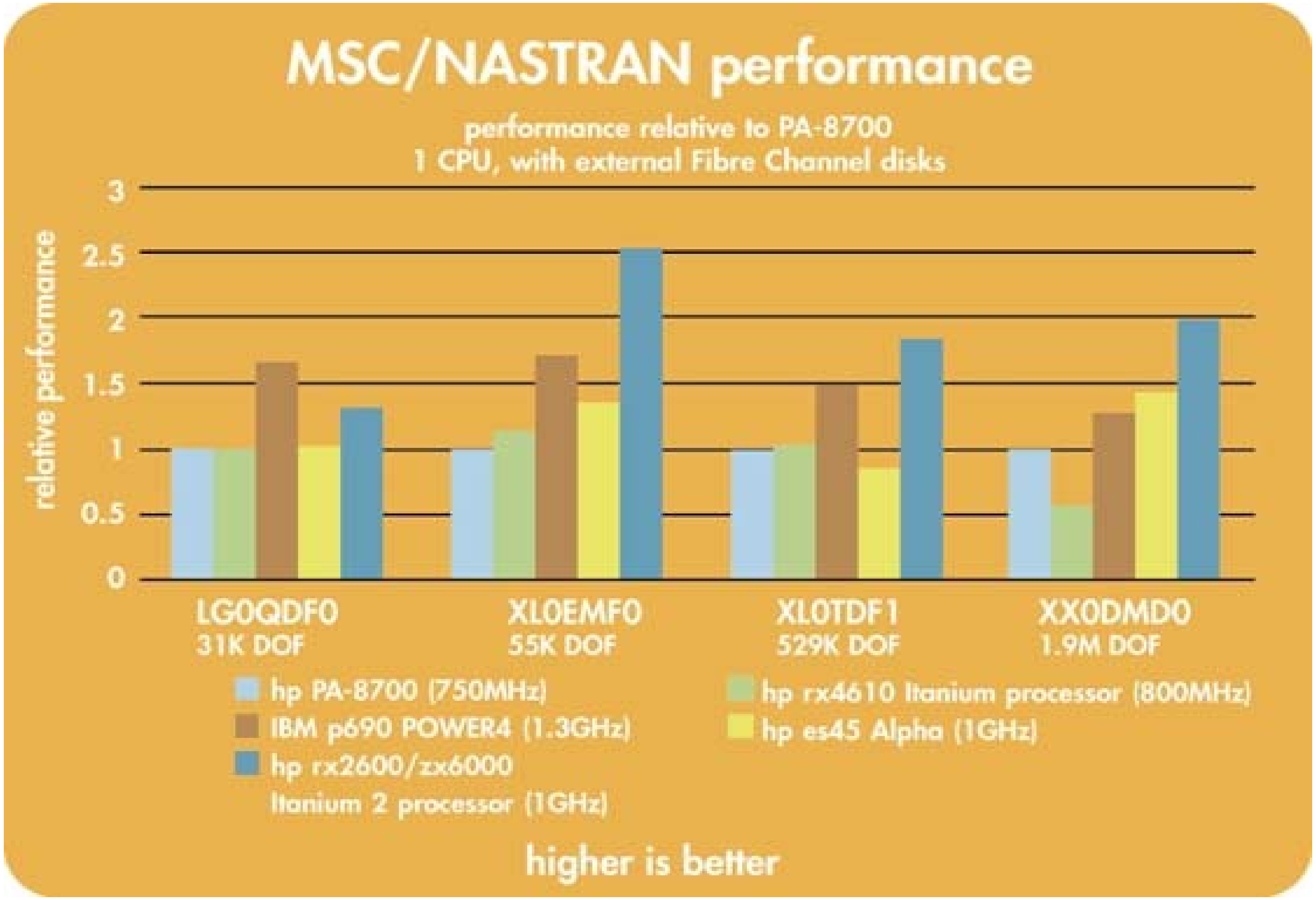
(1 CPU machine peak : 4000 Mflop/s)

Matrix	Size	Non-zeros	# of CPUs	Factor	Speed up	Solve	Speed up
af23560	23560	484256	1 4	3331 5763	1.73x	1105 1860	1.68x
ecl32	51993	380415	1 4	2320 4852	2.09x	1113 1753	1.58x
ex11	16614	1096948	1 4	2631 4953	1.88x	1153 1972	1.71x
twotone	120750	1224224	1 4	159 530	3.33x	467 677	1.45x
wang4	26068	177196	1 4	2649 5089	1.92x	1039 1742	1.68x

Linpack NxN results using HP MLIB

- Achieved 335.5 Gflop/s on a 64 Madison cpus (1.5 GHz) Superdome 87% peak
- Achieved 4.78 Tflop/s on a 1540 cpus Itanium2 (1 GHz) cluster at PNNL (78% of peak)
- Achieved 350 Gflop/s on a 118 cpus Itanium2 (1 GHz) cluster (84% of peak).
- Achieved 109.2 Gflop/s using 16x2 Itanium 2 (1GHz) cluster (32 cpus) 85% of peak

MSC/NASTRAN Itanium™2-based launch information



http://www.hp.com/products1/itanium/performance/technical/msc_nastran.html

HP MLIB for Itanium™ 2 Cluster summary

- Itanium™ 2 with MLIB achieves very high machine peak and is good for both floating point and I/O intensive applications
- MLIB group helps HP to achieve best NASTRAN benchmark results
- MLIB helps HP to be the leader in IPF Linux, i.e. PNNL, BP/America and OSC customers
- Evaluation copies of MLIB available at http://www.hp.com/go/mlib
- Contact us by e-mail at mlibcore@rsn.hp.com

hp

invent