Application of High Performance FORTRAN

Edward Bogucz  Geoffrey Fox
Tom Haupt  Kenneth Hawick
Kevin Roe

CRPC-TR95619
July 1995

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Also available as SCCS-727 from the Northeast Parallel Architectures Center at Syracuse University.
NPAC Technical Note SCCS-727
http://www.npac.syr.edu/
Syracuse NY 13244-4100
111 College Place
Syracuse University
Northeast Parallel Architectures Center

Edward Bogacz and Kevin Roe
Geoffrey Fox, Ken Haryck, Tom Haught

Performance Fortran
Application of High
Contents:

- HPF Applications Web Package
- Conclusions on HPF applicability for CFD problems
- Selected NPAR Projects
- Sparse Matrices in HPF - Conjugate Gradient Algorithm
- Panel Method and Full Matrix (Direct) Algorithms
- ADI Algorithm - conflict in optimal data decomposition
- Issues for Solver algorithms in HPF
- Brief HPF Overview
HPF Overview:

- Many details given elsewhere (see Web package)
- Particular architecture
- Additional parallel intrinsic functions - optimised on particular architectures
- Explicit parallel constructs like FORTAL:
  - by allowing various directives to specify data decomposition
- High performance Fortran (HPF): adds to Fortran 90
- Including tutorial on HPF Fortran page

MADIC Code Evaluation Workshop, NASA LARC
memory-efficiency and compute-efficiency occurs.

As for all parallel solver algorithms, tradeoff of
has good communication support for run time library.

Like solvers like ADI can work well if implementation

dense vectors and arrays.
many intrinsics functions do BLAS like operations on

propagate values in from boundary.
although some data-parallelism was lost during time to
regular mesh boundary value problem still good

methods for initial value problems
HPF's data-parallel model excellent for mesh/stencil

Solver Issues for HPF(T):

MADIC Code Evaluation Workshop, NASA LARC
Avoid ill-in problems.

Iterative method used instead of direct method and can store diagonals as vectors etc., providing space storage can be efficient in storage structured sparse systems can be implemented using unstructured sparse systems can be implemented using

Features planned for HPE-2 (private data for processors) INDEPENDENT but are inefficient without new

A good run time library module message passing can achieve (eg ScaLAPACK) could be but are not as efficient as the task parallelism that pure direct (full matrix) methods can be expressed in HPE

Solver Issues for HPE(2):

MADIC Code Evaluation Workshop, NASA LARC
Application of HPE, Slide: 6

Iterative methods (Conjugate gradient).
Panel Method (dense linear algebra).
Formulation.
Alternate Direction Implicit Problem.

Some Case Studies:
Formulation:

Alternate Direction Implicit Problem

\[
\begin{align*}
\forall \Omega, & \quad \nu \frac{\partial \phi}{\partial n} = 0 \\
(1), (2) & \quad (0, 0, t) = \Omega, \quad u_0 = f \Delta
\end{align*}
\]

• Dirichlet boundary conditions on \( \phi \)
• \( h^N_{\phi} \) intervals in \( x \) and \( N + 1 \) intervals in \( \phi \)

Consider Equation: 

Application of HPE, Slide: 7
END
END DO

... check convergence ...

CALL THOMAS(NX,C,BCFL,BCRL,PSI(0:NX+1,J))
CALL THOMAS(NY,C,BCBT,BCRT,PSI(I,0:NY+1))

... do not change these ...

DO J = 1, NY \& sweep

DO I = 1, NX

DO IER = 1, IERMAX | begin iteration loop

... deassertions, interfaces and initializations ...

PROGRAM ADI-SEQUENTIAL
END
RETURN
Z(0) = Z0
END DO
Z(K) = D(1,K)*Z(K+1) + D(2,K)
DO K=NK,1,-1
Z(NK+1) = ZN
END DO

(c(2,K) + c(3,K)*d(1,K-1))

D(2,K) = (c(4,K)-c(3,K)*d(2,K-1))/c(2,K)
D(1,K) = -c(1,K)/(c(2,K)+c(3,K)*d(1,K-1))
DO K=1,NK
O(0) = Z0
O(0) = 0.0

SUBROUTINE THOMAS (NK,C,Z0,ZN,Z)

Thomas algorithm for ADI code:
solve a set of tridiagonal system of equations

\[ \text{END DO} \]

\[ \text{DY2INV} \times (\text{PSI}(1:NX, j+1) + \text{PSI}(1:NX, j-1)) \]

\[ c_{4, j} \times j = f_{j} \]

\[ c_{3, j} \times j = \text{DX2INV} \]

\[ c_{2, j} \times j = -2.0 \times f_{j} \times \text{PSI}(j) \times \text{DY2INV} \]

\[ \text{DX2INV} \]

\[ \text{DO J = 1, NY} \]

\[ \text{solve a set of tridiagonal system of equations} \]

\[ \text{END DO} \]

\[ \text{DX2INV} \times (\text{PSI}(1:NY, I+1) + \text{PSI}(1:NY, I-1)) \]

\[ c_{4, I} \times I = f_{I} \]

\[ c_{3, I} \times I = \text{DX2INV} \]

\[ c_{2, I} \times I = -2.0 \times f_{I} \times \text{PSI}(I) \times \text{DY2INV} \]

\[ \text{DX2INV} \]

\[ \text{DO I = 1, NX} \]

\[ \text{solve a set of tridiagonal system of equations} \]

\[ \text{DO J = 1, NY} \]

Modelled x- and y-sweeps for data parallel execution.
Optimal Data Decompositions for Matrix Equations in ADI

Solver (2D problem)
Application of HPF, Slide: 12

- Time library is implemented.
- Performance depends on how well intrinsics and run
  3D and higher dimensional problems
- REDISTRIBUTE is used instead of TRANSPOSE
  REDISTRIBUTE
  (multiplications)
  MATMUL intrinsic replaces all dense array
  Fortran 90 intrinsics. (eg. SAXPY is one line of code,
  many of BLAS type operations are in-line equivalents of
  ADI can be neatly expressed in HPF.

ADI Conclusions:
source strength at \( \rho \)

Ellipses in uniform incident flow \( \bar{U} \) inc. showing Kth panel of

Panel Method - Ellipse Example:

MADIC Code Evaluation Workshop, NASA LARC
determined, velocity field obtained from potential. Once vector of source densities panel and x-axis, RHS vector is $\varphi_0 = \varphi \sin \varphi$, where $\varphi$ is angle between and RHS vector is $\varphi$.

\[
\int_{\varphi_0}^{\varphi} \left[ \frac{\nu \varphi}{\xi} \right] + \frac{Z}{1} = \varphi \mathcal{V}
\]

\[
\mathcal{V} = \Phi \text{ with } \varphi = \varphi_0 \mathcal{W}
\]

Surface: $\mathcal{W}$ generates system of linear equations from boundary condition of zero normal flow through body strength of the K'th panel. Source densities determined is distance between two panels, and $\varphi_0 = \varphi_0$ is the source.

\[
\int_{\varphi}^{\varphi} \left[ \frac{\nu \varphi}{\xi} \right] + \frac{Z}{1} = \varphi_0 \mathcal{W}
\]

\[
\int_{\varphi}^{\varphi} \left[ \frac{\nu \varphi}{\xi} \right] + \frac{Z}{1} + \varphi_0 \mathcal{W} = \varphi \mathcal{W} \Phi
\]

Distribution of $N$ source produced potential:

Panel Method Formulation: Body in a uniform velocity potential.
Application of HPF, Slide: 15

Multiplication factor.

Partial matrix row, column requires broadcast of
requirements no communication. Row requires broadcast of
pivot, whereas for column this is poorly balanced - but
Row distribution requires a distributed global test for the
Row distribution across (4) processors, memories. (Column

Cyclic

Block

Full Matrix Decomposition: MADCIC Code Evaluation Workshop, NASA LARC

25 July 1995
REDISTRIBUTE may be costly.

Substituting the mixed distributions is also possible but required to work on entire RHS at all stages of back substitutions are poor since a single processor is

Cyclic row distribution provides best decomposition for both

**Cyclic**

**Block**

Higher dimension Matrix Distribution
on programmers behalf. May be part of run time library invoked by the compiler.

Although in future SCALAPACK SYTML library (e.g. for example passing for this sort of application) present implementations cannot outperform message REDISTRIBUTED are implemented.

Computer system (how well TTRANSPOSE and problems sizes and characteristics of particular which of the decompositions performs best depends on BLOCK, * easily changed to (*CYCLIC))

Once code is in HPF, it allows different decompositions code sections as important to remove serial code as to add parallel HPP allows problem to be expressed quite neatly.

Full Matrix Conclusions:

MADIC Code Evaluation Workshop, NASA LaRC
scalars $a$ and $b$ and as well as the matrix $A$ and working vectors $x$, $t$, $d$, $q$ and as $b \cdot d/dx = x$

ENDDO

DO $k = 2, NIter$

$\langle b \cdot d \rangle / d = x, q = t = d$

end
Matrix A.

Compressed Sparse Column (CSC) Representation of Sparse Matrix A.

Compressed Vector:

Sparse Matrix A:

Compressed Storage Format:
Multiplication where A is distributed in a (BLOCK, * )

Communication requirements of (sparse) Matrix Vector

Matrix A

Vector b

Vector p

Communications for Matrix Vector Multiply (ROW)
Matrix multiplication where A is distributed in a (\textbullet, \text{BLOCK})

\text{Multiplication requirements of (sparse)} Matrix \text{Vector}

\text{Communications for Matrix Vector Multiply (Column)}:
HPP version of sparse storage CG (CSR format):

\begin{verbatim}
 HPF $DISTRICT\text{E}\text{R} Row\text{C}\text{Y}\text{C}\text{I}\text{C}(n+1)/4)
 HPF $DISTRICT\text{E}\text{R} Col\text{B}\text{L}\text{O}\text{C}(k)
 HPF $DISTRICT\text{E}\text{R} A(Block)
 HPF $DISTRICT\text{E}\text{R} (Block) X, P, I, X
 HPF Processor : Proc(Smp(Np))
 REAL, dimension(i:n) X, I, P
 INTEGER, dimension(j:n) I, J
 REAL, dimension(k:n) A
\end{verbatim}
END DO
IF ( stop criterion ) EXIT
   x = x - alpha * saxpy
   x = x + alpha * p
  alpha = rho / DOT_PRODUCT(p, q)
END FORMAL

END DO
( b ) = ( A(t) * p(t) + a(t) ) + ( b(t-1) )
DO t = row(j), row(j+1)-1
   FORALL ( j=1 : n )
      sparse matrix-vector multiply
      c = 0.0
      b = beta * c + x
      beta = rho / rho0
      rho0 = rho
      rho = rho0
   END IF, NITER
DO K=1, NITER

Call version of sparse storage CG (CSR format) -
Conclusions:

CG (Sparse Methods)

25 July 1995
NPAC Projects Exploring HPF Applicability:

- Blackhole Binaries
- Weather/Climate Optimal Data Interpolation / Assimilation
- Parallel Shear Flow Code and Acoustic Equations implementation in HPF
- Work with ICASE on TLNS3D code
difficulties now being uncovered by applications.

HPF-2 language definition may address many of

(Digital, Portland, APR, ...) - HPF compilation systems now actually available

parallel algorithms.

 disadvantages (in common with any parallel

additional temporary data-store requirements of

implementation) over serial implementations are

Disadvantages (in common with message-passing implementations.

additional code portability and ease of maintenance by

and distributed computers; HPF (potentially) allows faster computation on parallel

Overall Conclusions:

MADIC Code Evaluation Workshop, NASA LARC
Talks and Lectures on HPF.

Answers to commonly asked questions on HPF.

On-line HPF Tutorial:

Associated parallel computing issues:

List of papers and books on HPF, Fortran90 and HPF+.

Discussion of issues of relevance to HPF and HPF+.

List of generic example application applications codes with suitability of high performance Fortran.

An indication of appropriate software including

List of Industrial and Academic application areas with

List of available Compilers, Translators, ...

Contents of the HPFA Package:

MADIC Code Evaluation Workshop, NASA LARC
email: hpfram@upac.syr.edu

The National HPCC Software Exchange:
http://www.netlib.org/nse/home.html

Project material at NpAC:
http://www.npaci.syr.edu/NpAC

Online Internet Resources: