UPC Applications

Parry Husbands
Roadmap

- Benchmark small applications and kernels
  - SPMV (for iterative linear/eigen solvers)
  - Multigrid
- Develop sense of portable UPC programming style (using T3E and Compaq AlphaServer)
- Motivate and evaluate compiler optimizations
- Move to larger applications
  - Candidates should be hard with current techniques:
    - Large N-body problems
    - Sparse Direct Methods
    - 3-D Mesh Generation
...
Sparse Matrix-Vector Multiplication

- $Ax=b$ with $A$ sparse
- Distributed Compressed Row Format Used for $A$
- Vectors distributed across processors
- Communication of elements of $x$ needed to compute $b$

\[
\begin{array}{cccc}
  x_0 & x_1 & x_2 & x_3 \\
  \hline
  A_0 & & & b_0 \\
  A_1 & & & b_1 \\
  A_2 & & & b_2 \\
  A_3 & & & b_3 \\
\end{array}
\]
Communication Strategies

• Need to send elements of x to processors that need them
  — Individual sends?
  — Pack?
  — Prefetch?

• Try to overlap communication with computation
  — Initiate communication
  — Do some local computation
  — Wait for remote elements
  — Compute on remote elements
T3E Results

SPMV on T3E in UPC

MFLOPS

Processors

Lawrence Berkeley National Laboratory
Compaq Results (1)

SPMV on Compaq in UPC

MFLOPS vs. Processors

- MPI (Aztec)
- UPC Small
SPMV on Compaq in UPC and MPI (4 procs/node)
Discussion

• Small message version required access to low latency messaging for performance
  – Manually done on T3E
  – Under investigation on Compaq
• Pack/Unpack version gives best portable performance
  – Relies on large messages (usually best performing)
  – Requires more source code
  – Investigating inspector/executor techniques
• Proposal
  – Make life easy for the compiler and add a pragma:
    #pragma prefetch(vector, indices)
Multigrid

- Taken from NAS Parallel Benchmarks
  - Hierarchy of grids ($256^3 \rightarrow 2^3$)
  - Project down to coarsest grid
  - Solve
  - Prolongate and smooth back up to finest grid
- Operators all involve nearest neighbour computations in 3-d and ghost region exchanges
- Code based on OpenMP version from RWCP
- Simple domain decomposition scheme used to map 3-d grid to a 3-d processor grid.
- On T3E computation compiled with CC (multidimensional array performance poor with gcc)
Data Structures

- For grid large, static distributed array not feasible
  - Difficult to change sizes at runtime
  - Need to access through local pointers for performance
    (avoid A[i] for pointer to shared A)
- Pointers to local regions (upc_local_alloc()’d) used instead
  - Can easily access any global element
  - Directory can be cached locally
T3E Results – Class B (256³)

MG Class B -- T3E

- Mops/s
- Processors

Lines represent:
- UPC
- MPI
- Linear
T3E Results - Class C (512³)

MG Class C -- T3E

Mops/s

Processors

UPC
MPI
linear
Outperform MPI Fortran version on T3E!
Single processor performance an issue
No speedups past 8 processors on Compaq
  - Spins to signal incoming variables
  - May need to reorganize communication
No small message version yet. Probably not worth it on Compaq.