Comparing XT5 Compilers

Lee Higbie
Arctic Region Supercomputing Center
higbie@arsc.edu +1-907-450-8688
www.arsc.edu/~higbie
Outline

• Compiler Rating System
• Sample results
• Problems with PAPI
• Recommendations & Desires
Caveats

• These ratings were based on specific compiler versions.
• PAPI timing is suspect
• Assembly language misses memory hierarchy management
Rating a Loop

• Only two options -O2 and “-fast”
• Compare execution time
  – -fast to -O2
  – compare to PGI (ARSC default)
• Looked at assembly code
  – only a few (extremal ratio) loops
  – hides memory hierarchy management
Intra-Compiler: Simple Loops

DO I = 1, nSomeDat * k
   is1(I) = is2(I) + is3(I)
enddo

DO I = 1, nDat * k
   XL1(I) = sqrt(XL2(I))
enddo

2x faster

2x slower
Intra: Subscr, Jumps, Rcrsn

\[ T_1(I, J, K) = T_2(I, J, K)^3 \]

\[
\text{DO } L=1, IS2(9) \\
F_1(I, J, K, L) = (F_2(I, J, K, L)/T_1(I, J, K))^2 \\
\text{DO } M=1, IS2(15) \\
F_{V1}(I, J, K, L, M) = F_{V2}(I, J, K, L, M)^2 - 2.0
\]

\[
\text{DO } I=1, \text{NPARHD, NPAR2} \\
\text{DO } J=1, I \\
\text{\quad } \text{XP}_1(\text{NPARHD}-I+1, I) = 1.144 \times \text{XP}_2(I+J, I-J) \\
\text{\quad } \text{endo}
\]

2x slower

2x faster

10x faster
Inter-Compiler: Simple Loops

\begin{verbatim}
DO I=1, nDat * k
  XL1(I) = sqrt(XL2(I))
enddo

DO I=1, nDat * k
  XL1(I) = XL1(I)/v1 + xl1(i-1)/(36.0*v2) + xl1(i-2)*v3/372.8 + XL2(I + 1)
enddo

DO I=1, nLongDat
  XL1(I) = XL2(I)
enddo
\end{verbatim}

5x slower

2x faster
Inter: Subscr, Jumps, Rcrsn

Logs of Time Ratios of Loops with Messy Subscripts, Branches or Recursion

5x slower

loops with branches (or masked vector operations)

non-linear indexing

2D arrays with unusual indexing

2x faster

10x faster
Problems with Loop Timing

• PAPI provided the only high-res, portable timer we found
  – MPI timers ~ 0.1 μs on one machine

• Repeatable errors
  – bizarre but reproducible results
  – error depends on total job time???

• PAPI library has bad Fortran
Compiler Conclusions

• Compiled code quality varies widely

• For best speed: Experiment
  – “fast” options speed operation, usually
  – “better” compilers => faster code, usually

• Industry cannot and should not expect users to experiment
Timing Conclusions

• We need high-resolution timers
• PAPI makes O/S call
  – Variable overhead
  – Makes timing small code sections difficult
  – Cray has better routines
    • Need them in Fortran Standard
    • Need them user-available
Thank you

Questions

• **Linked from www.arsc.edu/~higbie**
  – Complete program code

• **Spreadsheets of results and graphs**
  – additional families:
    • structures & F90 constructs
    • function or subroutine calls in loop
    • formatting
    • ....... (10 total)
Comparing XT5 Compilers

Lee Higbie
Arctic Region Supercomputing Center
higbie@arsc.edu  +1-907-450-8688
www.arsc.edu/~higbie