Compilers and Parallel Computing

Dr. Clayton S. Ferner
Associate Professor
Department of Computer Science
University of North Carolina Wilmington
Outline

- Compilers
- Parallel Computing
- Automatic Parallelizing Compilers
Compilers

- Computers understand Machine Code (instructions in binary)

```
0000000 457f 464c 0102 0001 0000 0000 0000 0000
0000010 0002 003e 0001 0000 0440 0040 0000 0000
0000020 0040 0000 0000 0000 0da8 0000 0000 0000
0000030 0000 0000 0400 0038 0008 0040 001d 001a
0000040 0006 0000 0005 0000 0040 0000 0000 0000
0000050 0040 0040 0000 0000 0040 0040 0000 0000
```

- Humans can understand Machine Code, but is it extremely difficult to work with
Compilers

- Humans understand natural languages (English, French, German, etc.)
- Computers do not understand natural languages
- Natural languages are too ambiguous to be useful to program a computer:
  - “Add a pinch of salt.”
  - “I'll meet you at 4ish.”
  - “How about a cup of coffee?”
  - “What's the matter?”
Compilers as translators

- A high-level language is one that humans can easily work with but is unambiguous

```c
int i;
for (i = 0; i < 10; i++)
    array[i] = 2*i;
if (a < b)
    a = b;
return a;
```

- Compilers are essentially translators
A compiler is a software program that translates a high-level programming language to assembly language.

```c
if (a < b)
a = b;
```

Assembly code:

```
movl %esi, -8(%rbp)
movl -4(%rbp), %eax
cmpl -8(%rbp), %eax
jge .L2
.L2:
movl -8(%rbp), %eax
movl %eax, -12(%rbp)
```
Compilers as translators

- Assembler is software that translates a program in assembly language to machine code

```
movl %esi, -8(%rbp)
movl -4(%rbp), %eax
cmpl -8(%rbp), %eax
jge .L2
.L2:
movl -8(%rbp), %eax
movl %eax, -12(%rbp)
```
Courses related to compiler technology

- CSC 360 Formal Languages and Computability
- CSC 434 Programming Languages
- CSC 457 Compiler Construction
What is parallel computing?

- Parallel computing is the use of multiple processors to solve a single problem or to work on a single program.
- The use of multiple processors to work in separate independent programs is not parallel computer.
What is parallel computing?

- Human beings are very good at parallel work:
  - Cutting grass
  - Building a house
- The more processors there are to work on a problem the faster it gets done
- Linear Speedup – if you have a program that takes $T$ seconds to complete on one processor, with $N$ processors dividing up the work, it could take as little as $T/N$ seconds
Parallel computing is a challenge

- Hindrances to real speedup - synchronization and communication
- The processors must be programmed correctly to perform their share of the work and to synchronize and communicate properly
- Writing a correct program to run on multiple processors is much more challenging than writing one to run on a single processor
Parallel computing is important

- We've pushed the limits of processor speeds
  - Speed of light
  - Heat
- In order to increase the speed of computers they need to use multiple processors
Example: Matrix Multiplication

\[ \begin{pmatrix} 
\vdots 
\end{pmatrix} \begin{pmatrix} 
\vdots 
\end{pmatrix} \begin{pmatrix} 
\vdots 
\end{pmatrix} \]
Example: Matrix Multiplication

```
for (i = 0; i < N; i++) {
    for (j = 0; j < N; j++) {
        c[i][j] = 0.0;
        for (k = 0; k < N; k++) {
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
        }
    }
}
```
Example: Matrix Multiplication

Performance Results
Matrix Multiplication of 512x512 Arrays

Execution Time (seconds)

Number of Core Processors
Speedup Results
Matrix Multiplication of 512x512 Arrays

Multiple Cores (Parallel)
Ideal Speedup

Speedup (Sequential Time / Parallel Time)

Number of Core Processors
Parallel computing is important

- Many machines nowadays have dual- and quad-core processors
- Intel will soon be making machines with hundreds of processors
- What do we do with all those processors?
Courses Related to Parallel Computing

- CSC 337 Parallel Computing
- CSC 437/537 Parallel Computing II
- CSC 446/546 Grid Computing
- CS Department has a cluster of 11 computers with 44 processors
Automatic Parallelizing Compilers

- Wouldn't it be great if you could just write a program to run on one processor, then have a compiler figure out how to make it work correctly for multiple processors?
- Automatic Parallelizing Compilers attempt to do just that
- Problem: it is even more challenging to figure out how to get a compiler to automatically parallelize a program than to have a human do it.
- Very limited success
Paraguin Compiler

- The Paraguin Compiler is a compiler I built using the SUIF Compiler from Stanford Univ.
  - [http://people.uncw.edu/cferner/Paraguin/](http://people.uncw.edu/cferner/Paraguin/)
- Automatic parallelizing compiler
- Produces MPI code for execution on Distributed Memory systems
Discussion

Questions?

http://people.uncw.edu/cferner/papers/CSC10010F.pdf