Performance Programming: Theory, Practice and Case Studies

Rajat Garg, Ilya Sharapov
Sun Microsystems, Inc.
rajat.garg@sun.com, ilya.sharapov@sun.com

Alistair Rendell
Dept. Computer Science, Australian National University
alistair.rendell@anu.edu.au
Introduction

Module I: Measuring Program Performance

Module II: Serial Optimization

Module III: Parallel Optimization

Module IV: Case Studies
Focus and Goals

- Target audience: developers of technical and HPC applications
- Focus on C, FORTRAN 77, FORTRAN 95 on UNIX/Linux platforms
- Can be applied to applications in
  - Mechanical computer-aided engineering (MCAE)
  - Electronic design automation (EDA)
  - Computational chemistry
  - Bioinformatics
  - Operations research
  - Financial modeling
  - Reservoir simulation and seismic modeling
  - Mechanical computer-aided design (MCAD) modeling
  - Graphics rendering and imaging
  - Climate and weather
Why Program for Performance

- Optimized Code
  - Solve a problem faster
  - Solve a larger problem
  - Solve problem of the same size but with fewer system resources
Performance Components

- **Hardware**
  - CPU
  - Memory system
  - Disks and peripheral devices

- **Software**
  - Compilers
  - Libraries
  - Monitoring tools

- **Parallelization Support**
  - Multiple processors
  - Memory and OS support for MP
  - Parallelizing compilers
  - Clustering software
Performance Programming

Serial Optimization Stage

Unoptimized Correct Code

Measure Baseline Performance

Perform Compiler Optimization

Correct Results?

Yes

Check Applicability

No

Link Optimized Libraries

Correct Results?

Yes

Verify APIs

No

Identify Performance Bottlenecks

Apply Selective Compiler Optimization

Modify Source Code

Correct Results?

No

Check Applicability

Yes

Measure Performance

Satisfactory?

Yes

Optimized Serial Code

No