Roadrunner Programming Course: Intermediate Level

Dates: Jan 27-29, Feb 3-5, Feb 10-12 (all 2009)
Days: Tuesday, Wednesday & Thursday
Time: 1:30 - 3:30 PM
Location: Encantado Training Room, Suite 201, Room #219, Los Alamos Research Park.
Contact: sriram@lanl.gov

What it is:

Getting started with hands-on programming on Roadrunner
Simple optimizations: SIMD / unrolling
Data flow analysis
SPU instruction analysis

Prerequisites:

Required: basic knowledge of MPI
Required: basic knowledge of C
Required: comfortable with C pointers
Required: knowledge of how to compile and run C codes

Lecture 1: Introduction (Jan 27)
Instructor: Cornell Wright (Ben Bergen)

Introduction to Roadrunner development systems

Account request, sign in, modules, build/compile

Introduction to Roadrunner topology

Cell Processor (PPE/SPE), Opteron, Connectivity (DaCS/ALF, MPI), Endian conversion, Compiler options

Lecture 2: Process Launch (Jan 28)
Instructor: Cornell Wright (Ben Bergen)

Overview of process launch and communication paradigms

MPI hello world launch
Launch process on CBE using DaCS
Launch process on SPE using DaCS and libSPE / pthreads
Introduction to other ways of launching: ALF, ssh

Lecture 3: Basic Debugging Clinic (Jan 29)
Instructor: Cornell Wright (Sriram Swaminarayan)

printf debugging
gdb debugging over DaCS

Lecture 4: Control / Data Movement - I (Feb 3)
Instructor: Ben Bergen (Sriram Swaminarayan)

Use DaCS to Transfer the node data and slope computation to PPE (and then to the SPE)

Introduces:
```
posix_memalign()
To/From CBE with DaCS
Endian conversions
```
Demonstrates:
```
DaCS Put/Get Vs. Send/Recv
Blocking / non-blocking communication
Scalar code on the SPU
```

Lecture 5: Control / Data Movement - II (Feb 4)
Instructor: Sriram Swaminarayan (Ben Bergen)

Transfer compute from PPE to SPE using libSPE/DMAs

Introduces:
```
Understanding of DMA / libspe
```
Demonstrates:
```
Simple DMA transfers
Double buffered DMA and the 16k limit
Running on one SPU
Running on many SPUs
SPU Synchronization
```

Lecture 6: Advanced Debugging Clinic (Feb 5)
Instructor: Sriram Swaminarayan (Cornell Wright)

Brief introduction to totalview, if available
Writing functions in gdb
Looking at arrays in gdb

Lecture 7: Optimization-I: SIMD (Feb 10)
Instructor: Tim Kelley (Sriram Swaminarayan)

Strategies for SIMD-ization

Introduces:
```
SIMD vector types
spu intrinsics
```
Demonstrates:
```
Vector types
Gross timing using gettimeofday()
Comparison with SSE, perhaps using SAL
```

Lecture 8: Optimization-II: Unrolling, Branches (Feb 11)
Instructor: Sriram Swaminarayan (Cornell Wright)

Advanced Optimization Strategies

Introduces:
```
Loop Unrolling
Compiler pragmas
Branch elimination
spu_timing
spu_decrementer
```
Demonstrates:
```
Loop Unrolling
#pragma disjoint()
Eliminating if statements
Pipeline stalls
Simple profiling on the SPU
```

Lecture 9: CPC, Items From The Floor, Practice (Feb 12)
Instructor: Marcus Daniels (Cornell Wright)

Finding bottlenecks using Cell Performance Counter (CPC)
Items that have been parked during the rest of the lectures

Nota bene:
Descriptions, discussions, and examples will be in C. While much of the information applies to Fortran, Fortran won't be used in the course.

Course materials will be on portal (http://rralgs.lanl.gov/portal):

Codes
Lecture Notes
Cell Documentation
Knowledgebase

Version 1.7, 1/21/09 5:36 PM