SGI Porting and Optimisation Seminar

Ruud van der Pas

SiliconGraphics

European High Performance Computing Team

Mode of Operation

This seminar is run in interactive "VCR mode":

- **Repeat parts**
- **Move faster or slower**
- **Pause**

Remember:

You (have to) control the buttons as well

Goals of this seminar

Topics

- *R10000 and Origin2000 Main Architectural Concepts*
- *RISC oriented Optimisation Techniques*
- *Compiler Environment*
- *Shared Memory Parallelisation*
- *Program Development Environment*

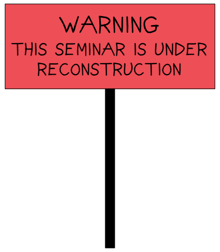
After the seminar you should have a good understanding of these concepts and be able to use them in practical situations on your own application(s).

How to get (more) information

It is strongly recommended to use:

- ***The UNIX man pages***
- ***SGI's Insight Documentation***
- ***On-line help within the products***

It is a good investment to really use these sources of information



Agenda – Monday November 24

- Intro
- Origin Products
- The Memory Hierarchy
- The R10000 Architecture
- The Origin Architecture
- The Origin Software Environment
- Porting Applications to IRIX
- Floating Point Exceptions
- Program Development Tools
- Developer Magic, part 1

Breaks will be held regularly

Agenda – Tuesday November 25

- Software pipelining
- RISC Optimisation Techniques
- The MIPSpro Compilers
- Case Study

Assistance with practical porting/performance issues

Breaks will be held regularly

Agenda – Wednesday November 26

Today you will be applying the course material to your own program:

- *Make sure you have a (short) validation run*
- *Port your program*
- *Find the optimal compiler options*
- *Use ssrun to find the time consuming parts*
- *Verify you get good sequential performance*

Agenda – Thursday November 27

- Introduction to Parallelisation
- Data Dependency
- Shared Memory Parallelisation
- Advanced Parallelisation
- Case Study
- Distributed Shared Memory
- Tuning for the Origin
- Developer Magic

Breaks will be held regularly

Agenda – Friday (morning) November 28

Today you will be applying the material to your own program:

- ***Make sure you have a (short) validation run***
- ***Port your program***
- ***Find the optimal compiler options***
- ***Use ssrun to find the time consuming parts***
- ***Verify you get good sequential performance***
- ***Then parallelise your application***