Extending the Sparc-Sulima Computer Simulator

Peter Strazdins,
(with Bill Clarke and Andrew Over)

- **see** [http://cs.anu.edu.au/~Peter.Strazdins/postgrad/ExtSulima.html](http://cs.anu.edu.au/~Peter.Strazdins/postgrad/ExtSulima.html) for project description and relevant links

- including slides for background talk (COMP3800 s8 2002) and this document
1 About the Sparc-Sulima software

- It's a big software artefact, written in C++ and Python
- It has lots of ‘cool’ features
  - Heavy use of OO - modularity, some re-use
  - Use of SLED for automatic generating of instruction decoding code
  - SWIG/Python scripting interface
  - ‘Annotation’ framework for flexible modification of simulator behaviour
  - Nifty makefiles
  - Heavily use of optimization techniques – speed matters!
- It has a significant history (5 developers over 2002–2002), developed under the ANU-Fujitsu CAP Program
- It has a long way to go still! (under the CC-NUMA Project, 2003-2006)
  - Must support the performance analysis of a huge Gaussian quantum chemistry application, running on a CC-NUMA UltraSparc III SMP
2 What’s Involved in Extending Sparc-Sulima?

could tackle 1–2 of these per BSEng student:
1. extend for portability: run on many platforms
   • integrate Marcus Watts’ extensions back into main code tree
2. extend for configurability: easy change of architectural parameters
   • may involve some re-design; re-design for maintenance may also be valuable
3. add event-gathering infrastructure (essential for CC-NUMA!)
4. systematic (regression) testing (challenging: test space is enormous!)
5. advanced debugging support, eg. gdb interface

Also there are the following more research-related projects
• check-pointing in a complex OO system:
  • important for simulating large applications
• extend to simulate (and run on) cluster computers
3 What’s Interesting/Different about Working on Sparc-Sulima?

- become part of a ‘crack’ team, itself part of a moderate-sized research project (CC-NUMA)
  - we believe we have some good SE practices, *but*
  - we could benefit from new team members with different (more rigorous) approaches
- the problem domain has some real depth
  - state-of-the-art computer architecture and how large-scale applications run on them
- the software artefact has interesting features and a significant history
- clearly established links with Sun Microsystems!
- opportunities for follow-up (e.g. postgrad study) exist