

Challenges in compiler research

Alex Ramirez

Compiler & Architecture Seminar

Haifa, IL

December 2004

Position statement

- **Single processor domain**
 - There's still some work to do ...
 - ... specially for the new processor architectures
- **Multiprocessor / multithreaded domain**
 - That's where the real challenges are found ...
 - ... but it's also where no previous success is known
- **New explicitly parallel programming models**
 - We complain about the architecture second-guessing the compiler ...
 - ... but the compiler also second-guesses the programmer :-)

Superscalar / VLIW Processors

- Well known targets
 - Scalar optimizations
 - Control flow optimizations
 - Code scheduling
 - Code layout
 - ... not much else to do
- Maybe the only target is to reach the full potential
 - Extensive use of run-time data
 - Dynamic optimization
 - Profiling information
 - Better tools are needed for this!
 - Add compile-time speculation
 - Requiring hardware support

Data parallel architectures

- Namely: Vector processors
- We know how to compile scientific applications
 - Cray / Convex compilers were quite good at that
- We don't have good "C" compilers for vector code
 - Data dependencies / alias analysis is harder
 - Extensive use of pointers
- Again, there may be a use for compile-time speculation

Dataflow architectures

- Distributed processor architectures require explicit routing of data elements between functional units
 - Not load/store + register-to-register operation anymore
 - Not 3-address code anymore
- The compiler must build explicit data dependence chains ...
 - ... and then map those chains to functional units
- Data dependence analysis comes back to haunt us again
 - Only this time it's really critical: the runtime won't save us

We need to help the compiler

- Bring back the good old co-design days
 - Ask the compiler people:
 - what code can you generate?
 - which support do you need in the architecture?
 - and which NOT :-)
- Teach people to code for the new architectures
 - Nobody said piloting a 747 was easy ...
 - ... so, why should programming a multiprocessor be easy?