

Gelato member report:  
Research on ILP (Instruction  
Level Parallelism) and  
TLP (Thread Level Parallelism)

Weimin Zheng

*Tsinghua University, China*

Oct. 14, 2003

# Outline

- Researches on optimization compiler technologies to enable high performance computing based on IA64 platform for scientific/engineering computation.
- Efforts on different granularities: ILP and TLP
- Utilizing IA64 features to enable aggressive optimization to exploit ILP and TLP
- ORC based, open source for Linux world.

# Compiler Optimization technologies for High performance computing

## ■ Research goals

- Software pipeline approaches to improve ILP
- OpenMP based approach to explore TLP

## ■ projects

- Software pipeline enhancement in ORC
- ORC based OpenMP compiler and tools

## ■ Supported by HP and Intel

# Software pipeline enhancement in ORC

## ■ Finished project

- SoftwarePipeline enhancement in ORC, by introducing more complex control on the code and register usage
  - **Loop Unrolling control in software pipeline**
  - **Using stack register to alleviate static register pressure in software pipeline**
- 2-3% performance gain on average for NPB (typical scientific computing)
- Has been integrated into ORC2.1

# Software pipeline enhancement in ORC: Ongoing projects

- Take IA64 architectural features into account to allow more loops to be software pipelined.
- Data speculation enhanced software pipeline
  - Hand-checking exhibits great opportunities for current ORC implementation
  - Handle ambiguous memory access and sparse data dependences across adjacent iterations
  - Compensate code handling is very complex

# Software pipeline enhancement in ORC(cont.)

- Software pipeline for loops with complex conditional branches
  - Predication
  - Must be cautious about the predicate register usage
- Accurate Cost model to evaluate the benefit of software pipelined loop
  - Selective software pipeline

# ORC based OpenMP research

## ■ Project goals

- Provide a research vehicle with necessary OpenMP functionality, high performance and related tools support
- Parallel programming environment for OpenMP users to ease the adaptation and tuning process:
  - **Profiling**
  - **overhead analysis and performance tuning support**
  - **Support for correction check**
- Using of OpenMP as a TLP exploitation paradigm for HPC on shared memory computers, CMPs and other multi-threading architectures

# The basic OpenMP research platform

- An OpenMP compiler for C/Fortran90 programs, supporting most constructs in current standards
- The OpenMP runtime system, based on pthread library on Linux, with basic profiling support
- As a patch to ORC1.1 and ORC2.0
- Available to open source society
  - <http://sourceforge.net/projects/orc-openmp>

# Ongoing researches on OpenMP

- OpenMP compiler implementation
- OpenMP optimization
- Tools for OpenMP programming
- Using OpenMP for cluster based computation
- OpenMP based approach for thread-aware architectures

# OpenMP compiler implementation

- To complete the left functionalities
  - Limited Threadprivate implementation
  - Workshare translation
  - Other checking and language specified compatibility
- improve the performance of the runtime system
  - NPTL(Native Posix Thread Library) instead of pThread for better efficiency and scalability
  - More flexible task-queue based model, for future extension
  - Well tuned synchronization structure
  - Integrated interface from perflib for instrumentation and profiling
- Will also be open sourced soon

# OpenMP optimization

- Stand alone optimization modules other than the translation module, optional
- Optimization for the directives
  - Eliminate non-beneficial directives
  - Static schedule for load balance
- Memory access optimization
  - Eliminate possible cache anomalies
- Synchronization elimination
  - Region merging and rule-based barrier elimination
  - A simplified implementation of fork-join to SPMD transformation

# Tools for OpenMP programming

- **OpenMP events profiling and monitoring**
  - Automatically enabled profiling support in Runtime library
  - Extended ORC profiling for OpenMP module interested events
- **A tool to support computation-error detection**
  - To provide the ability of checking the correctness of an OpenMP program adapted from sequential scientific/engineering computation programs
  - A facility to identify and group computation regions for correction check, and determine the region map between sequential/parallel versions
  - Intermediate computation results interception and check.

# Overhead causality analysis for OpenMP

- Extended Overhead analysis
- Layered overhead classification model
  - Higher level program structures induce multiple overheads across overhead categories
  - Middle level translation and optimizations are not necessarily visible to the stand alone analysis system
- Machine level performance events profiling and monitoring
- Using Bayesian network to infer high level causes of overheads observed at lower levels

# Using OpenMP for cluster based computation

- Motivation: The using of OpenMP paradigm for cluster based computation is still not clear
- Questions: How effective does OpenMP explore the parallelism on SMP?
  - Do OpenMP and MPI do the same to hide latency?
  - How effective do OpenMP and MPI synchronize the computation?
  - Do OpenMP and MPI introduce the same kind of overhead?
- Low level Comparison of OpenMP/MPI programs using IA64 PMU
- Intended to find the whether there is a way to compensate the bottleneck of both paradigms

# OpenMP based approach for Thread-aware architectures

## ■ Questions:

- Does OpenMP programs get extra benefits on thread-aware architectures?
- How much difference is there between the OpenMP's granularity and auto-explored TLP granularity?

- Basic idea: using OpenMP as one approach to express parallelism, and still heavily relies on more aggressive methods to automatically explore remain TLP. E.g., TLDS( Thread level data speculation) to handle sparse data dependence loops(DOACROSS)

# Summary

- Advanced software pipeline algorithms to improve the ILP
- Work to build an OpenMP platform for further research
- Research to
  - make OpenMP programming easy
  - use OpenMP for cluster based HPC
  - use OpenMP to explore TLP

# Questions and Comments

- *Any question?*
- *Any comment on further research?*
- *Thank you.*