The research of some key technologies of compiler optimization for on-chip multi-core processor

Introduction
Multi/Many-core processor is the inevitable trend of development of processor, and has been applied in desktop, server and mobile and embedded system. However, the power, memory and the parallelization of software limited the development of multi-core processor. In this project, we studied several key technologies of the compiler optimization for on-chip multi-core processor: Architecture of Multi-core Processor, Acceleration Model, Compiler Optimization, Low-power framework and methods, Task Stealing Scheduling etc.

Multi-core Processor Modeling

• Scratch Pad Memory Model
  - Characteristic: Small-Capacity; low-power Consumption; high-speed access
  - Allocation strategy: Static Allocation; Dynamic allocation

• Data Pipeline Model
  - Dynamic voltage scaling
  - Instruction-level speed modeling
  - Graphical data fetch modeling
  - Independence between DMA and CPU
  - Data Predictability from loop iteration

• Power Cost Model
  - Quantify static power consumption
  - Quantify command power
  - Quantify power of cache and memory access
  - Multi-core communication power modeling

Multi-core simulation tools (epcc.sjtu.edu.cn/EPCC_SESC.htm)

• Add DVFS instructions
• Calculate DVFS power using Watch model
• Provide three Benchmark examples
• Web-based visualization, user-friendly

Conclusion
We use compiler to analyze and optimize user programs automatically. In the popular multi-core architecture scenario, the optimized programs are scheduled among cores to save power with guarantee of no vital performance degradation. Compiler is our tool of achieving high performance and low power consumption. These technologies provide the support for high-performance computer, cloud computing and software parallelization.