

Goal

We propose to explore the power-performance tradeoffs of emerging multi-core platforms. In a multi-core processor, application mapping, scheduling and load balancing will be carefully addressed to optimize the power dissipation. Application of dynamic frequency voltage scaling (DVFS) also has significant potential for power savings.

Background

Modern large-scale computing systems, such as data centers and High Performance Computing (HPC) clusters are severely constrained by power and cooling costs for solving extreme-scale (or exascale) problems. The increasing power consumption is of growing concern due to several reasons, e.g., cost, reliability, scalability, and environmental impact.

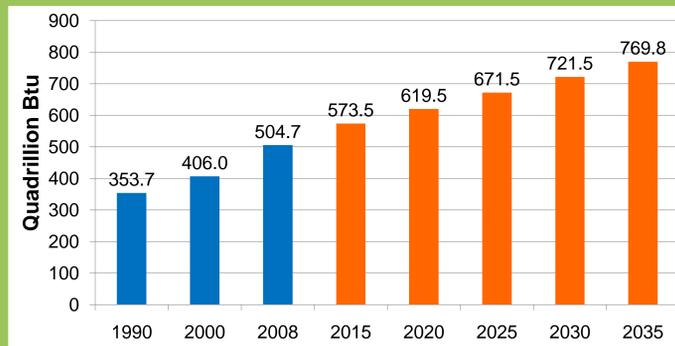


Figure 1: World marketed energy consumption, 1990-2035
(Source: U.S. Energy Information Administration: <http://www.eia.gov/forecasts/ieo/index.cfm>)

Method

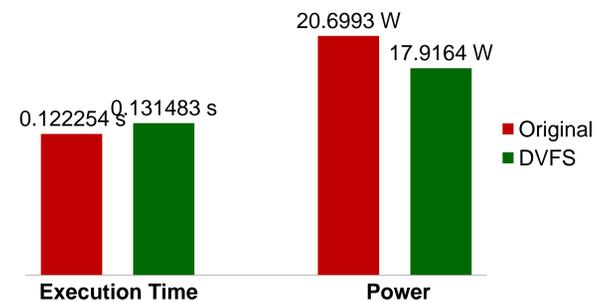
- **Dynamic Frequency and Voltage Scale (DVFS) based on Compiling Info.**
 - Use static information to direct dynamic Scheduling
 - Aiming at figuring out where in the code the processor would be in low utilization
 - Develop flags (e.g. instruction, hardware trigger) to invoke proper actions of processors
- **Static Analysis of Instruction Flow**
 - Analyze the execution pattern of various benchmarks
 - Identify the time frames when the processors are in low utilization
 - Study the power and performance tradeoff by reducing frequency and voltage during the identified time frames

Data Collection

DVFS Based on Compiling Info.

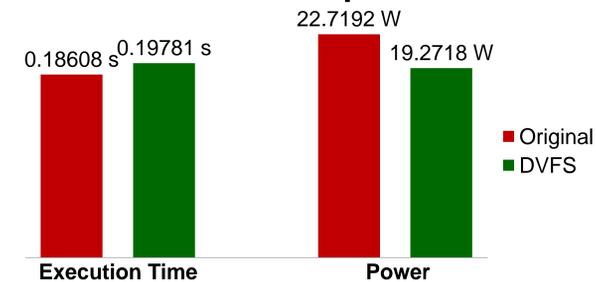
Splash-2 Benchmarks

LU Comparison



Latency penalty: **7.55%**
vs.
Power savings: **13.44%**

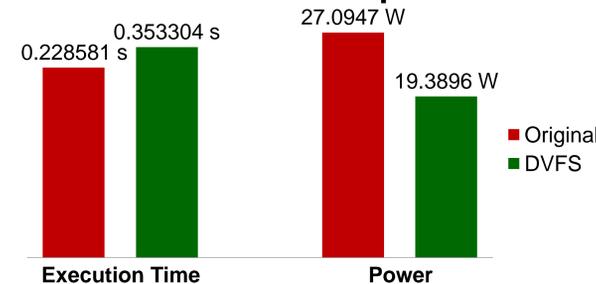
RADIX Comparison



Latency penalty: **6.30%**
vs.
Power savings: **15.17%**

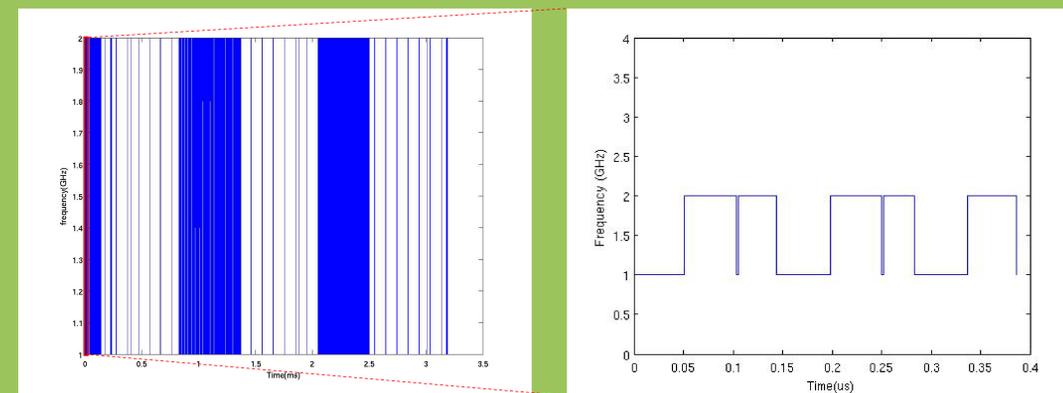
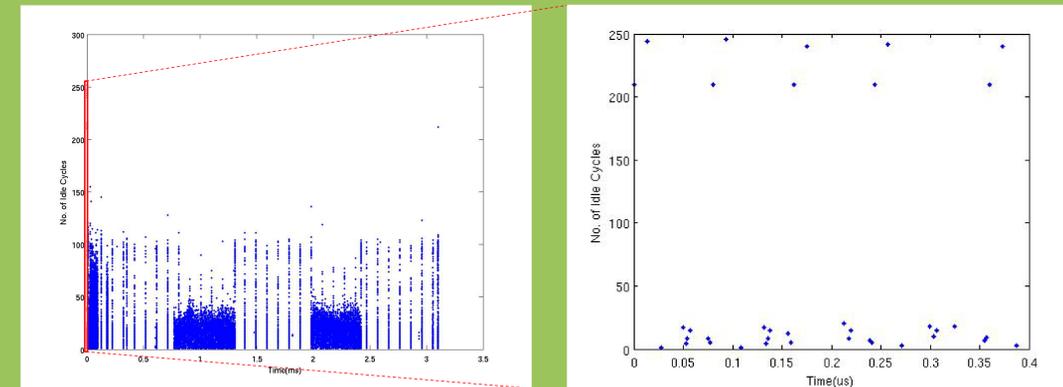
Parsec Benchmarks

CANNEAL Comparison



Latency penalty: **10.82%**
vs.
Power savings: **28.44%**

Flow Prediction



Conclusion

In our research, we insert flags during compiling time identifying the potential idling part of processors and during execution we scale voltage and frequency based on the flags. Using this approach we observed around 15% power savings with 7% latency penalty for Splash-2 benchmarks and around 30% power savings with 10% latency penalty in Parsec benchmarks.

We also studied the program execution pattern and were able to identify time frames of low processor utilization. It provides us opportunity to save power. Thus we can come up with a model to predict the low utilization periods.

Future

- **For DVFS based on Compiling Info.:** Build new compiler with such power oriented flags.
- **For Regression Flow Prediction:** Build models to predict low utilization periods.
- **Other Approaches:**
 - OS level scheduling
 - Theoretical research: to develop a scheduling model in order to normalize scheduling policy

