Powerful Beyond Dissipation?

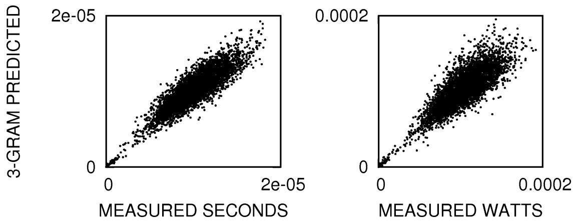
Virtually all the power that goes into a modern processor comes out as heat. Circuit densities have reached a point where the ability to dissipate that heat is a fundamental limit on performance. The cost of power and cooling also have become significant fractions of the total cost of ownership. We are developing systems technologies to deliver high performance without exceeding power and thermal limits.

Predictive Control. Power management is complicated by the fact that temperature of a system can continue to rise long after power consumption has been cut. Reactive control based on real-time sensor readings must allow for this hysteresis, which effectively means the control must be very conservative. To allow the system to safely operate closer to its limits, control must make use of predictions more tightly bounding future behavior. Over the past year, we have demonstrated new compiler technology that can predict power consumption over time intervals long enough to allow effective control.

The first step is prediction of energy consumption and execution time for each individual instruction. One might suppose this could be accomplished using detailed architectural implementation models, but most chip makers do not provide the information needed. Empirical methods suffer two fundamental problems: the inability to directly measure power at anything close to instruction granularity and the fact that modern processors have complex internal state (superscalar pipelining, caches, etc.) which may dramatically alter the power associated with execution of a particular instruction. We estimate instruction-level properties by:

1) Constructing and benchmarking a large number of synthetic benchmarks using code idioms employed in real applications
2) Analyzing each benchmark as a set of instruction N-grams (each representing an instruction within an N-instruction context); one benchmark generates one linear equation summing N-gram properties to obtain the overall properties measured
3) Using a genetic algorithm to solve the overspecified system of linear equations

As shown below, the resulting instruction N-gram energy consumption and time estimates generally are accurate to within 20%, with higher accuracy as N is increased.



The lookahead analysis is complicated primarily by two factors: it must be able to span all types of control flow, including recursive function calls, and "all paths" analysis tends to suffer exponential growth in complexity as path length is increased. Our approach converts the program's *executable object file* into a state machine representation that trivially allows spanning all types of control flow. Running on a PC, our fast lookahead algorithm can fully annotate a 10K-line program's object file with million-instruction lookahead results in about a minute.

We also can use runtime sampling to enhance the static annotations. However, keep in mind that the static annotations focus on *bounding the worst-path power* – dynamic sampling primarily enhances predictions of the *expected power*.

Runtime use of these predictions is fairly straightforward, however, these predictions only tell part of the story. Our runtime support will not only use predictions for scheduling each program in the system and setting parameters such as node voltages and clock speeds, but will combine this information with an online fluid dynamics model to anticipate heat flow and a variety of environmental sensors and other information to optimally manage a cluster supercomputer. For example, environmental sensors can determine the efficiency of air conditioning. Knowledge of utility electricity rate schedules can be used to shape usage so that electricity cost is minimized with minimal impact on job completion time.

Scheduling With Real-Time Constraints. As with scientific computing applications, real-time systems are increasingly turning to parallelism to improve performance. We have been looking at improved methods for reducing power consumption in real-time embedded parallel systems.

Modern real-time systems can no longer rely on static offline scheduling. In addition to power and energy constraints, they have dynamic task sets with new tasks arriving and departing while the system is running. We are currently developing power-aware, real-time scheduling algorithms to schedule dynamic task sets on multiprocessor systems.

Our algorithms can assign sub-tasks of an end-to-end task to processors, and distribute the end-to-end task's deadline. An end-to-end task is a chain of dependent sub-tasks with a common deadline and period. Each sub-task can be assigned to a different processor. An online acceptance test determines if a new end-to-end task can be scheduled with existing end-to-end tasks. Tasks are assigned to processors in such a way as to reduce power consumption due to the added task and due to communication with other tasks. Processor voltage and speed are controlled to reduce energy consumption.

This document should be cited as:

```
@techreport{sc06power,
author={William Dieter and Henry Dietz},
title={Powerful Beyond Dissipation?},
institution={University of Kentucky},
address={http://aggregate.org/WHITE/sc06power.pdf},
month={Nov}, year={2006}}
```

Aggregate.Org UNBRIDLED COMPUTING

SC06