**SLIDE 1: TITLE**

Thank you for the introduction. My talk is on “Optimal Reliability-Constrained Overdrive Frequency Selection in Multicore Systems”. This is joint work with professor Kahng at UCSD.

**SLIDE 2: OUTLINE**

My talk is structured as follows. I begin with motivation followed by previous work, contributions of our work, the problem formulation, flow of our optimal (discretized) solution, and our experimental setup and results. Finally, I conclude.

**SLIDE 3: RELIABILITY IN MULTICORE SYSTEMS**

Modern multicore processors, such as the Intel Xeon, AMD Athlon, etc. operate at multiple operating modes to meet different performance and power requirements. For example, nominal, supply voltage scaling and turbo.

Reliability is a key design consideration at leading-edge technology nodes to meet a prescribed system lifetime.

Task scheduling affects how each core in a multicore processor is used. A subset of cores can fail before others.

**SLIDE 4: SCHEDULING IN MULTICORE SYSTEMS**

Applications have different requirements of the number of cores to use. The operating system scheduler packs tasks from applications using some or all of the available processing cores.

The figures show how Applications A on the top and B at the bottom have different requirements for the number of cores during its execution phase.

The figure on the right shows percentage active time on the Y-axis and number of active cores in the X-axis for Application A in blue and Application B in maroon. The green bars show how cores are used when the scheduler packs tasks in a eight-core system. The core usage is roughly a Gaussian distribution with the mean being the total of the available cores divided by two.

**SLIDE 5: CORE WEAROUT**

Cores wear out when they are active. Mean time to failure or MTTF is a measure of a core’s lifetime. Reliability mechanisms such as, electromigration, etc., degrade a core’s MTTF. We focus on MTTF degradation due to EM. When all cores are not simultaneously active, the OS scheduler can schedule tasks such that cores wearout in a balanced manner.

**SLIDE 6: IMPACT OF OVERDRIVE FREQUENCY**

To meet performance and throughput requirements, cores are overclocked.

The overdrive frequencies cause faster MTTF degradation due to elevated temperatures.

And results in two challenges.

The system can violate an “acceptable throughput”, that is, cores fail before all assigned tasks are completed. OR

The system can violate a minimum “acceptable performance” , that is, cores operate at less-than-desired frequencies.

**SLIDE 7: TERMINOLOGY**

Before I proceed with the remaining talk, I will introduce terminology used in our work.

Power-on-hours is the effective number of lifetime hours consumed and is a measure of a core’s lifetime degradation due to operating conditions.

Nominal temperature is the temperature at which MTTF degradation is the same as the number of hours a core is active.

Acceleration factor is the ratio of original MTTF at nominal temperature to the actual MTTF at higher-than-nominal temperature.

**SLIDE 8: OUTLINE**

Let us examine previous works that study the impact of task scheduling on reliability.

**SLIDE 9: CLASSIFICATION OF EXISTING WORKS**

We use the following labels to classify existing works. The letter “N” before any label signifies a Non or No. For example, RC is Reliability Constrained and NRC is Non-Reliability Constrained, LG is lifetime guarantee and PG is performance guarantee.

Reiss12 is NRC, NLG and NPG. They describe how to maximize throughout by stressing cores to operate at their maximum temperature. Karpuzcu09 is RC but NLG and NPG as they describe how to maximize performance by degrading a subset of cores.

Several works are examples of RC, LG and NPG. They provide various dynamic policies. However, none of them provide any performance guarantees.

**SLIDE 10: COUNTEREXAMPLE TO NRC POLICIES**

In our paper, we provide counterexamples to existing works with numerical examples. Consider a task schedule as shown in the table on the right for a system with four cores. There are nominal as well as overdrive tasks that require between one and four cores to be active.

NRC policies run cores at the max frequency of 3GHz. By calculating POH we can determine that cores fo not have sufficient lifetime to execute the overdrive tasks at m = 3. In addition, no task requiring m = 4 can be completed.

Therefore, NRC policies cannot guarantee “acceptable throughput”.

**SLIDE 11: COUNTEREXAMPLE TO RC-LG POLICIES**

For the same task schedule, RC-LG policies will initially execute cores at 3GHz and later drop to 1.6GHz to meet lifetime. Again by calculating POH, we observe that all tasks requiring three and four active cores will operate at 1.6GHz, that is, less than the acceptable 1.8GHz.

Therefore, RC-LG policies cannot guarantee “acceptable performance”.

**SLIDE 12: OUTLINE**

So, what do we do differently?

**SLIDE 13: WHAT DO WE DO DIFFERENTLY?**

We formulate a new Maximum-Value Reliability-Constrained Overdrive Frequencies or MVRCOF optimization problem. This is solved offline. Our formulation is important because the overdrive frequencies are the optimization variables and the value is the user experience when the overdrive frequencies are maximized.

We guarantee prescribed levels of “acceptable performance” and “acceptable throughput”.

**SLIDE 14: COMPARISON OF OURS VS. EXISTING WORKS**

Recall that existing RC-LG works provide no performance guarantees. In our work, we provide performance guarantee in addition to lifetime guarantee under reliability constraints.

**SLIDE 15: WHAT IS THE OPTIMAL SOLUTION?**

Our formulation can determine the optimal solution for the same scheduling problem for which existing NRC and RC-LG policies are suboptimal. Using exhaustive search we determine the overdrive frequencies as shown in the bottom table. Note that all our overdrive frequencies meet the minimum “acceptable performance” requirement as well as guarantees “acceptable throughput”.

**SLIDE 16: OUR KEY CONTRIBUTIONS**

Our key contributions are

We develop a new MVRCOF formulation to maximize the value of operating multiple cores at overdrive frequencies.

Our solutions guarantee both “acceptable performance” and “acceptable throughout”.

We propose an optimal (discretized) solution flow using exhaustive search as well as an approximate heuristic flow

We empirically determine our optimal solutions improve the objective function value by up to 17.4% as compared to existing works.

**SLIDE 17: OUTLINE**

Now I describe our problem formulation.

**SLIDE 18: FORMULATION**

This slide presents our formulation in a formal manner

**SLIDE 19: FORMULATION IN ENGLISH**

We maximize the sum of the product of weight, frequency and execution times in nominal and overdrive modes across all m. The execution times multiplied by frequency determines the duration for which cores operate at a given frequency.

**SLIDE 20: FORMULATION IN ENGLISH**

Our first constraint ensures a lower bound on “acceptable performance”, that is, 30% above the nominal frequency.

The second constraint guarantees all tasks are completed within system’s lifetime and cores wearout in a balanced manner.

The third and fourth constraints are upper bounds on instantaneous power and temperature.

**SLIDE 21: MVRCOF INPUTS: TASK DESCRIPTION**

Our formulation has two kinds of inputs: Task description and system description.

The OS scheduler obtains core usage and performance requirements from applications and determines the total execution times in nominal and overdrive modes, the weights and the nominal frequencies.

**SLIDE 22: MVRCOF INPUTS: SYSTEM DESCRIPTION**

An SoC designer provides details on the number of available cores, maximum power of any core, maximum frequency achieved by a core, maximum die temperature, nominal temperature and the intial MTTF of each core.

**SLIDE 23: MVRCOF OUTPUTS**

The MVRCOF solver outputs the optimal overdrive frequencies, percentage time in each combination of active cores. For example, in a three-core system, two cores can be active in 3 ways. And the percentage of lifetime each core executes in nominal and overdrive modes.

**SLIDE 24: MVRCOF INPUTS AND OUTPUTS**

The inputs are provided to our MVRCOF solver to obtain the outputs.

**SLIDE 25: OUTLINE**

**SLIDE 26: OPTIMAL (DISCRETIZED) SOLUTON FLOW**

This slide shows our optimal (discretized) solution flow. It is discretized because we consider discrete overdrive frequencies within a range.

For each core and for each combination in which the core is active, we perform power and thermal simulations using discrete values of overdrive frequencies and generate a one-time lookup table of all possible values of overdrive frequencies, temperature and acceleration factors.

Now, we perform exhaustive search using values from this table to maximize the value of our objective function.

**SLIDE 27: HEURISTIC FLOW**

Recall our objective function. We maximize the overdrive frequencies in the order of maximum value of the product of weight and execution times of a set of active cores. For example, in a three-core system if the product of two active cores is greater than three and one active cores. We determine the overdrive frequencies for two cores, followed by three cores and one core.

**SLIDE 28: OUTLINE**

I describe our experimental setup, testcases and present results now.

**SLIDE 29: EXPERIMENTAL SETUP**

To resemble realistic vector processor-like core, we use 72 copies of the jpeg\_encoder from OpenCores. We perform synthesis, place and route using commercial tool flows and 45nm foundry libraries.

We perform power simulations with Synopsys PTPX. We increase voltage from 0.8V in steps of 10mV to 1.2V and frequency from 1.5GHz in steps of 50MHz to 3GHz.

We perform thermal simulations using HotSpot and our LP solver is lp\_solve. We use RC-LG policies as the baseline policy for comparisons.

**SLIDE 30: TESTCASES**

We develop our own testcases by configuring various parameters.

We develop eight testcases in total.

An example of a testcase with N = 4 is shown in the table below. The bold red numbers show that for four active cores, the nominal execution time is 2000h, overdrive execution time is 5000h and the corresponding weights are 0.4 and 0.6 respectively.

**SLIDE 31: OPTIMAL, HEURISTIC vs, RC-LG**

Here, I present only our key results. This figure shows the objective function value in Y-axis and the testcases in the X-axis. The figure compares the objective function values across optimal, heuristic and the baseline solutions.

Our optimal solutions achieve up to 17.4% higher value of objective function as compared to baseline solutions as shown by the arrows and percentage numbers in white. Our heuristic solutions can be up to 3.3% worse than our optimal solutions as shown by the arrow and percentage number in yellow.

**SLIDE 32: RUNTIME COMPARISON**

This figure shows normalized runtime in the Y-axis and testcase in the X-axis. Although our heuristic solutions can be up to 3.3% worse than our optimal solutions, the runtime can improve by up to 10x.

**SLIDE 33: OUTLINE**

I now conclude my talk.

**SLIDE 34: CONCLUSIONS**

We formulate and solve a new MVRCOF problem under lifetime reliability constraints.

We develop a MVRCOF solver that implements our optimal and heuristic flows.

Our solutions guarantee both “acceptable performance” and “acceptable throughput” and we empirically demonstrate that our optimal solutions can achieve up to 17.4% greater value in objective function as compared to baseline RC-LG solutions.

Our future works include application of our methods to traces from real server workloads, expand our methods to handle other objectives and achieve solutions that are temperature history-aware.

Thank you for your attention.