Thermal-Sustainable Power Budgeting for Dynamic Threading

Xing Hu[†‡], Yi Xu[§], Jun Ma[†‡], Guoqing Chen[¶], Yu Hu[†] and Yuan Xie[£]
[†]State Key Laboratory of Computer Architecture, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, Beijing, China
[§] Macau University of Science and Technology, Macau, China
[¶]Advanced Micro Devices, Inc., Beijing , China
[£]Pennsylvania State University, University Park, USA
huxing@ict.ac.cn, yixu@must.edu.mo, majun@ict.ac.cn,guoqing1.chen@amd.com,
huyu@ict.ac.cn, yuanxie@cse.psu.edu

ABSTRACT

As technology scales, thermal management for multi-core architectures becomes a critical challenge due to increased power density and higher integration density. Existing power budgeting techniques focus on maximizing performance under a given power budget by optimizing the core dynamics. However, in multi-core era, a chip-wide power budget is not sufficient to ensure thermal constraints because the thermal sustainable power capacity varies with different *threading strategies* and *core configurations*. In this paper, we propose a model which estimates the thermal sustainable power capacity considering these two run-time factors. The model converts the thermal effect of threading strategies and core configurations into power capacity, which provides a context-based power budget for the power budgeting. Based on this model, we introduce a power budgeting framework aiming to optimize the performance within thermal constraints, named as **TSocket**. Compared to the chip-wide power budgeting solution, TSocket shows 19% of performance improvement for the PARSEC benchmarks by reducing thermal violations and providing extra power budget for performance improvement.

1. INTRODUCTION

With the increasing transistor density, power dissipation of chips is growing rapidly. Power and thermal budgets become the major limitative factors of performance optimizationin multi-core processors [1]. Consequently, it is really important to explore power and thermal management. Power budgeting techniques have been proposed to achieve optimal performance under a chip-wide power budget. However, they may be risky and occasionally incur thermal violations. Although there are many thermal management schemes, these works can hardly support performance optimization for multi-threaded applications.

Power budgeting techniques use dynamic threading [2] (i.e. changing thread number and thread to core affinity in run-time) and DVF-
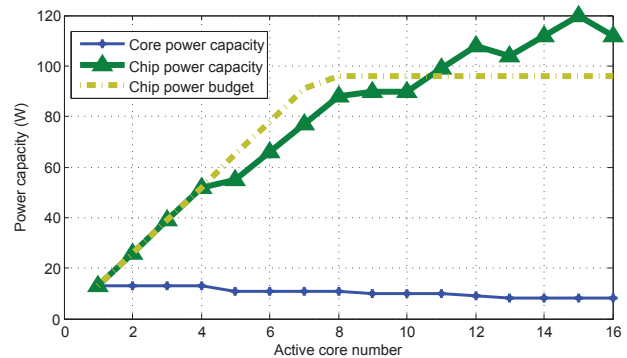
Figure 1: Power capacity varies with the number of active cores

S (dynamic voltage and frequency scaling) to cap power in adaption to the application characteristics. Both of these two techniques, dynamic threading and DVFS [3][4], manage core dynamics of chips. These techniques have been proposed to achieve optimal performance or energy efficiency under a chip-wide power budget [2][5]. However, we observe that the power capacity varies with the core dynamics. The power capacity is referred to as the thermal sustainable power limit of chips or cores without violating thermal constraints. Therefore the chip-wide power budgeting techniques cannot ensure the optimal core dynamics for performance improvement, because they are not aware of the gap between the power budget and the power capacity. Taking a sixteen-core processor as an example, we use hotspot simulation [6] to obtain the core power capacity with different active core number (the detailed descriptions of the experimental setup are in subsection 4.1). Fig. 1 shows that the chip power capacity varies with the number of active cores. We refer to the scenarios as over-budgeting when the chip-wide power budget is larger than the chip-wide power capacity. In these cases, the chip-wide power budgeting may result in thermal violations. When the chip power budget is less than the chip power capacity, which is referred to as under-budgeting, thermal headroom is wasted and the processor misses the chance of obtaining more computing capability. Both over-budgeting and under-budgeting may occur with the chip-wide power budget.

Existing thermal-power management work can predict power capacity based on thermal headroom estimation [7–9]. These work [7, 10] use a feedback loop to take the temperature and power in-

formation of chips as input parameters and predict the power limit for the next time interval. They utilize the thermal headroom and controlling the temperature of chips smoothly [11], but hardly support performance optimization for multi-threaded applications due to the following limitation. The thermal management caps power without taking the characteristics and demands of multi-threaded applications into consideration. For example, memory-intensive applications may not benefit from large power budget or high frequency. Or, some thread phases prefer threading while others prefer frequency boosting. It is difficult for thermal management works, which allocate power based on thermal headroom estimation, to meet diverse application demands.

In viewing the limitations of both power budgeting and thermal management techniques, we propose a *simple/concise yet accurate* model to correlate the relationship between power capacity and thermal constraints under different dynamic threading strategies. With assistance of this model, the thermal sustainable power capacity of each core can be determined for different threading strategies. Based on the power capacity estimation, the power budgeting work [5, 12] can select the optimal and safe core dynamics. Overall, we make the following contributions in this paper:

- We observe that the thermal-sustainable power capacity is varying with different dynamic threading strategies. We formulate the thermal coupling effect of these threading strategies in a quantitative way.

- We propose a thermal sustainable power capacity model, which converts the thermal constraints to power capacity under dynamic factors including threading strategy and chip temperature. This model can provide power capacity estimation for power budgeting to avoid thermal violations and achieving better performance.

- Based on the model, we introduce a framework, TSocket, to do optimal power budgeting under thermal constraints. The experimental results for PARSEC benchmarks [13] demonstrate that TSocket can effectively avoid thermal violations and outperform the chip-wide power budget approach by 19% on average.

The rest of paper is organized as follows: The details of the proposed model are described in Section 2 and the implementation of the TSocket framework is explained in Section 3. The corresponding experimental setup and results are given in Section 4, followed by the conclusions in Section 5.

2. THERMAL SUSTAINABLE POWER CAPACITY MODEL: TSOCKET

In this section, we formulate the thermal coupling effect of threading strategies based on active core distribution density. After that, we study the relationship between the thermal constraints and power capacity under different threading strategies and come up with a model. Finally, we introduce the thermal sustainable power budgeting framework **TSocket**.

For ease of analysis, we make the following two assumptions: 1) The active core distribution is determined by the threading strategy. Every active core hosts a thread for execution and the idle cores without thread running on it can be gated for power saving. 2) All active cores are running in the same working status. It is a practical assumption given that the present product processors [14], have the same design for the concern of design complexity and cost, and similar assumptions are in some relative works [5].
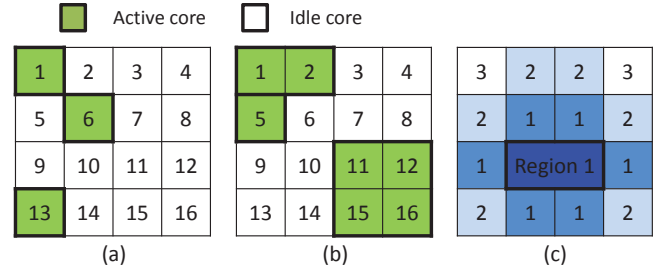

Figure 2: Region partition and region distance

2.1 Active Core Distribution Density

The core power capacity is strongly correlated with the distribution density of active cores. When there are less active cores in chip and these cores are distributed in a loose way, the power capacity of these active cores is larger. On the other hand, when there are more active cores in a processor and the active cores are distributed close to each other, the power capacity of active cores is smaller.

We use a metric of $DisF_c$ (Distribution Factor) to quantify the distribution density of the whole chip under certain threading strategies. This factor is determined by the largest thermal-effect region size (TS) of active regions in this chip, expressed as:

$$DisF_c = \max_{1<=i<=num_r} \{TS_i\}, \qquad (1)$$

where num_r is the number of active regions in the chip and the TS_i is referred to as the TS of active region i. The definition of active regions and thermal-effect region size is as follows.

Active region Active region is referred to as a rectangle area of chip where all cores are active. In order to calculate TS, first we should identify active regions in the chip. In this process, we follow the rule of maximizing region size and minimizing region counts during partitioning. Taking Fig. 2(a) as an example, there are three active cores: core1, core6, and core13, each of which forms a region. In Fig. 2(b), there are three regions: one consists of core1 and core2, one is core5, and the other consists of core11, core12, core15, and core16.

Region distance Region distance is an important factor for thermal-effect region size calculation. Before introducing the distance between two regions, first of all, we will explain the distance between two cores. The distance between core a and core b is the manhattan distance between them. Supposing that the coordinate of these two cores are (x_a, y_a) and (x_b, y_b) respectively, their distance is

$$d_{a,b} = |x_a - x_b| + |y_a - y_b|, \qquad (2)$$

and then the distance between two regions is the smallest distances of cores in these two regions, expressed as follows:

$$D_{i,j} = \min d_{a,b}, (a \in i, b \in j), \qquad (3)$$

where a refers to the cores in region i and b refers the cores in region j. The unit of the distance is the core size. The distance of a region to itself is defined as 1, *i.e.* $D_{i,i} = 1$. As shown in Fig. 2(c), the dark blue region is Region 1. The numbers in the neighboring cores represent the distance between Region 1 and the corresponding core.

Thermal-effect region size Thermal-effect size of region i, TS_i, reflects the thermal coupling effect of this region. The metric is the combination of its own region size and the external supplementary size of other regions. The supplementary size is determined
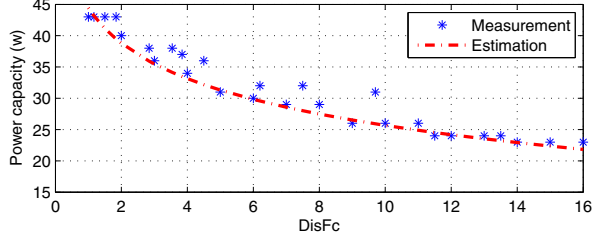
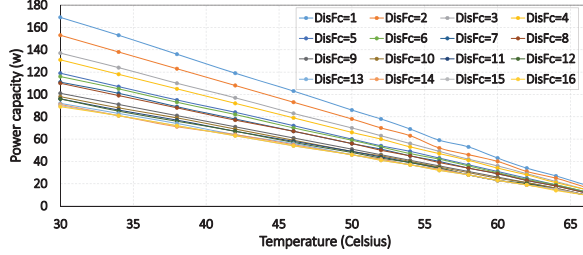**Figure 3: Active core power capacity for different $DisF_c$**

Figure 4: Impact of chip temperature on P_{core}
(core size is 8mm*8mm)

by both the distance and the size of other regions. Overall, the thermal-effect size of region i is expressed as follows:

$$TS_i = \sum_{j=1}^{num_r} S_j/D_{i,j}, \qquad (4)$$

where S_j is the size of region j. Taking Fig. 2(b) for example, TS_2 (for the region which consists of core5) can be obtained as: $TS_2 = S_1/D_{1,2}+S_2/D_{2,2}+S_3/D_{3,2} = 2/1+1/1+4/3 = 4.33$. Following the same way, there are $TS_1 = 4.33$, and $TS_3 = 5$. Therefore the $DisF_c$ of this scenario is 5. Larger $DisF_c$ implies smaller power capacity.

2.2 Power Capacity Estimation

Power Capacity and $DisF_c$. After characterizing the distribution density with $DisF_c$, we analyze the relationship between the power capacity of active core (P_{core}) and the metric $DisF_c$ under thermal constraints ($T < T_{limit}$). To obtain the thermal sustainable power capacity under a specific core configuration, we sweep the power of the active cores from low to high in Hotspot simulation [6]. During a specific time interval, once the peak chip temperature reaches the temperature limit (T_{limit}), the corresponding power of the active core is set as the P_{core}. The simulation time interval is determined by the monitor-actuation interval of thermal control mechanisms. The P_{core} with different $DisF_c$ is shown in Fig. 3. Curve fitting method is used to analyze the relationship between $DisF_c$ and P_{core}, and it is observed that the P_{core} is approximately a logarithmic function of $DisF_c$. The relationship can be described as

$$P_{core} = -C_1 * \ln(DisF_c) + C_2, \qquad (5)$$

where C_1 and C_2 are fitting parameters.

Impact of chip temperature. The hotspot simulation derives expression (4), which uses a fixed initial temperature of 60°C . In reality, the chip temperature is dynamically changing, which will affect the power capacity of the cores. To evaluate this chip temperature impact on P_{core}, we apply different initial temperatures
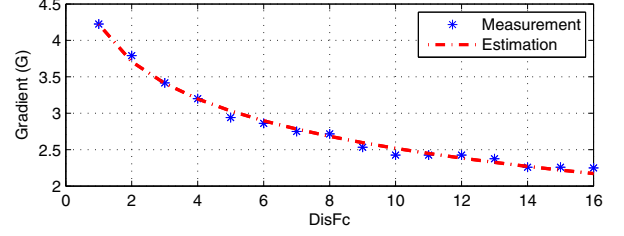

Figure 5: Correlation between gradient (G) and $DisF_c$

in the hotspot simulations, and the results are shown in Fig. 4. In this figure, different curves are corresponding to the scenarios with different $DisF_c$. The gradient of the curve is denoted as G. We can see that the P_{core} is decreasing approximately linearly with increasing chip temperature. Similar to P_{core}, G also changes with $DisF_c$, which is shown in Fig .5. With curve fitting, G can be expressed as,

$$G = -G_1 * \ln(DisF_c) + G_2, \qquad (6)$$

where G_1 and G_2 are fitting parameters. By including the dynamic temperature impact, equation (5) can be rewritten as

$$\begin{aligned} P_{core} &= P_0 - \Delta T * G \\ &= -C_1 * \ln(DisF_c) + C_2 - \Delta T * G \\ &= (G_1\Delta T - C_1)\ln(DisF_c) + C_2 - G_2\Delta T, \quad (7) \end{aligned}$$

where $\Delta T = T - T_0$, which is the temperature difference between current temperature and reference temperature (60°C in this work). P_0 is the core power capacity at T_0. G_1, G_2, C_1, and C_2 are related to the fixed chip features such as package cooling capability, T_{limit} and core size, which can be calibrated after the chip is fabricated, with the similar techniques for TDP (Thermal Design Power) measurement. The detailed steps to determine these coefficients are given in Section 3. Note that $DisF_c$ and chip temperature are dynamically factors impacting the thermal effect, which change power capacity. While the four coefficients (G_1, G_2, C_1, and C_2) depend on the static chip characteristics and are fixed during running time. When temperature limit (T_{limit}) rises or chip size gets larger, these coefficients become bigger or vice versa.

We validate the accuracy of expression (7) with a 16-core processor under different core dynamics and chip temperature. The core size in this processor is 8mm*8mm. The accuracy of expression (7) is illustrated in Fig. 6. The X axis is the actual power capacity obtained from hotspot simulation, and the Y axis is the power capacity predicted by expression (7). The distance between each marker ($\diamond\&\square$) and the diagonal dash line indicates the deviation of the estimated value from the actual value. The average deviation is about 5.8%. Note that thermal violation may occur when over-estimation happens, i.e. predicted power capacity is larger than the actual power capacity. According to the results, the chance of over-estimation appearance is 11%, but the largest positive deviation of over-estimation is about 2%. Because the difference between two neighboring power states of active core is around 20%, a 2% of deviation usually will not change the decision of power state selection. Hence, the possibility of incurring thermal violation with our power capacity prediction is very low. Consequently, the model can identify the power capacity accurately and conservatively in most of the time. With the proposed model shown in expression (7), we simplify the computation of power budget and only need small storage space to keep these four coefficients.
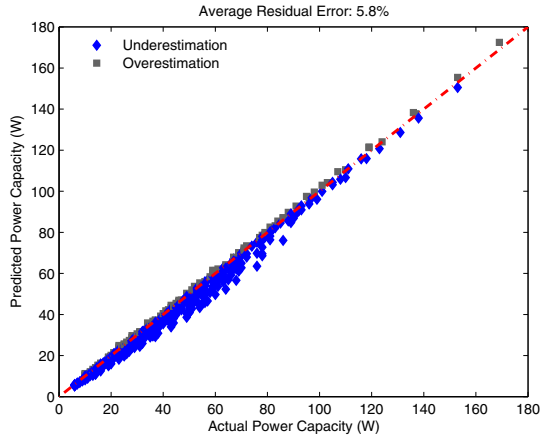
Figure 6: Accuracy of the proposed power capacity model

(a) Baseline: Chip-wide power budgeting

(b) TSocket: Thermal aware power budgeting

Figure 7: Chip-wide power budgeting vs. TSocket

2.3 TSocket

Based on the thermal-sustainable power capacity model, we propose a framework to ensure thermal constraints and meanwhile maximize the performance. TSocket is the vital nexus between thermal control and performance optimization. Rather than chip-wide power budget, TSocket power budgeting technique uses the 'context power budget' based on power capacity estimation under different threading strategies. Hence it is much safer and capable of avoiding thermal violations.

The working process of a chip-wide power budgeting is described in Fig. 7(a). The core configurations are selected based on chip-wide power budget, according to the performance-power model. This model can identify the thread and frequency sensitivity of applications and allocate power based on these two factors. Chip-wide power budgeting works are not aware of the thermal effect brought by the selected core dynamics. Thermal throttling may be occasionally introduced during the system execution. Alternatively, TSocket will first estimate context power budget for different core dynamics, and select the optimal solution, as shown in Fig. 7(b). T-Socket gives a larger design space for budgeting. Moreover, it considers the thermal effects of applications by quantifying the power capacity for frequency boosting and threading. Thermal violations can be avoided with the context power budget provided by TSocket.

3. IMPLEMENTATION

The model of TSocket is concise and easy to be implemented as well. In this section, we introduce the model calibration to determine the coefficients, followed by the practicality for implementation and compatibility with thermal-control works.

3.1 Model Calibration

The TSocket calibration is completed after the chip is fabricated. In a real processor with N cores, we first measure the power capacity when all the cores are on with various initial temperatures. This measurement can be supported by practical product developing procedures since the TDP is measured using the similar techniques. The core power capacity at temperature T_0 is $P_{a\_T_0}$, and the gradient of the power-capacity/initial-temperature curve is G_a. Afterwards, we measure the power capacity when only a single core is active under different temperatures. The core power capacity at temperature T_0 is $P_{s\_T_0}$, and the gradient of the power-capacity/initial-temperature curve is G_s. With these two sets of
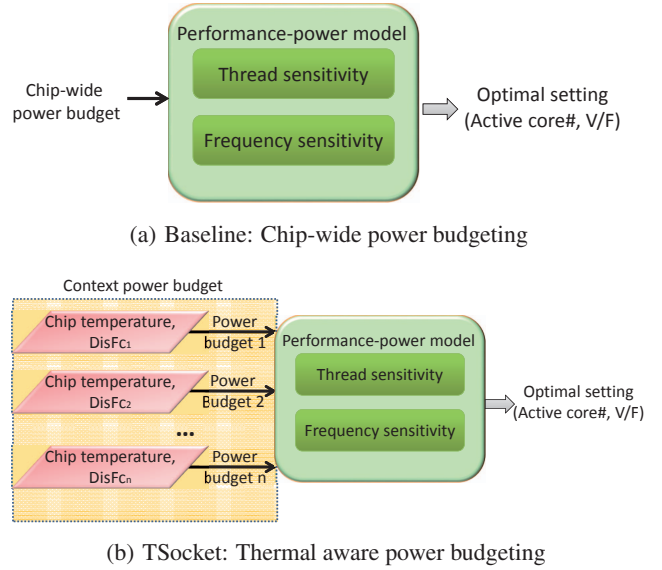
data, the coefficients C_1, C_2, G_1, and G_2 in expression (7) can be derived as: $C_1 = (P_{s\_T_0} - P_{a\_T_0})/\ln(N)$, $C_2 = P_{s\_T_0}$, $G_1 = (G_s - G_a)/\ln(N)$, $G_2 = G_s$.

3.2 Hardware Support

TSocket can be implemented as a firmware in chips. Nevertheless, it needs extra hardware supports for power budgeting, which includes the power control ability and interfaces of processors.

First, modern processors support a wide range of working points, which is the foundation for power budgeting. In AMD Trinity, there are seven HW- and SW-managed DVFS states, and the maximum power is about five times of the minimal power [15]. Hence it is possible to do power management with TSocket model.

Second, there are power control interfaces in product processors, such as power control registers. Using AMD microprocessors as example, they cap power under TDP that is recorded in performance registers, which are both readable and writable. The TSocket can modify these TDP registers to set up the power capacity of cores. After TSocket power budgeting techniques selecting the optimal configuration, the thermal control mechanisms such as BAPM [11] can do the fine-grained tuning to further exploit thermal headroom without thermal violation.

In addition, the TSocket model can be implemented with very little hardware overheads. This model comprises addition and logarithm operations. The logarithm operations can be extended as Taylor's series that consists of multiplications and additions. Hence TSocket is practical and can be applied in real processor designs.

4. EXPERIMENTAL RESULTS

This section first introduces the experimental setup, and then the effectiveness of TSocket is validated with informative statistics.

4.1 Experimental Setup

The thermal-sustainable power capacity model derives from thermal simulation with Hotspot 5.0 [6]. We evaluate TSocket on a customized multi-core processor simulated with GEM5 [16]. It contains 16 homogeneous cores, each of which is 4-issue out of or-

der, and has size of 3.65mm*3.65mm (estimated by McPAT [17] in 32nm). A distributed banked last level cache (LLC) is shared by all cores. Cache coherence is maintained by directory-based MOESI. As demonstrated before, DVFS is supported, with frequency range from 0.6GHz to 1.6GHz by step of 0.2GHz. The power trace derives from McPAT. We use PARSEC as the base workload, which is a set of typical shared memory multi-thread applications and shows diverse sensitivities to both thread number and frequency. The runtime management interval is set to 10 ms. The system can adopt OpenMP [18] for thread distributing or packing technique [5] to change thread to core affinities.

For comparison, the baseline is a runtime management with a chip-power budget, as shown in Fig. 7(a). For a fair comparison, we use *Oracle* power-performance model, which knows the relationship between performance and power of applications in the next time interval. In this case, the power-performance model always gives the optimal configurations with the maximum throughput under given power budget. Therefore, the difference between the baseline and TSocket is only determined by the strategy of power budgeting. Furthermore, both baseline and TSocket are equipped with thermal throttling mechanism to avoid thermal violations. The on-chip temperature sensors detect the temperature of each core, and throttling is triggered if the temperature of any core exceeds the limit. We adopt the thermal throttling method proposed by Indrani *et al.* [15], which ensures the thermal reliability by cutting down frequency immanently when throttling is triggered.

In our evaluation, we set the initial power budget of baseline to 96W, which is the TDP of our customized multi-core processor, and is defined as the typical power without incurring thermal violations in most of the time. For TSocket, power budget shows a large spectrum from 12W to 105W, which provides more potential for performance improvement (with provision of more power) and thermal reliability (with realistic power capacity).

4.2 The Effectiveness of TSocket

TSocket not only avoids thermal violation based on the power capacity estimation, but also can distinguish additional power budget in some cases.

Thermal Violation Mitigation. We *blackschole* for example. The execution traces of *blackschole* are shown in Fig.8. It includes threading strategies (thread number), working frequency, power, maximum temperature in chip, and throughput traces, in two scenarios: TSocket power budgeting (TSocket) and baseline power budgeting (baseline). The X axis covers a hundred of timing intervals. The TSocket scenario is marked in solid curve and the baseline scenario is marked in dotted line.

The baseline selects the configuration with active core number of six and frequency of 1.6GHz. Although the baseline seems to find the optimal solution with larger throughput, the thermal throttling triggered by the high temperature introduces performance penalty, and leads to overall low throughput. As shown in Fig.8, the baseline continuously incurs thermal violations and makes frequency throttling, which is harmful to both performance and reliability of the processor. TSocket, however, could avoid thermal violations with the assistance of power capacity estimation model.

Additional Power Budget Sniffing. With the assistance of TSocket, system can safely identify additional power budget during run-time. Fig. 9 illustrates the execution phases for benchmark *bodytrace*, and shows that TSocket and baseline select different threading strategies and core configurations. Although the chipwide power of TSocket is larger than the baseline, the temperature traces of TSocket always remain in a safe region.
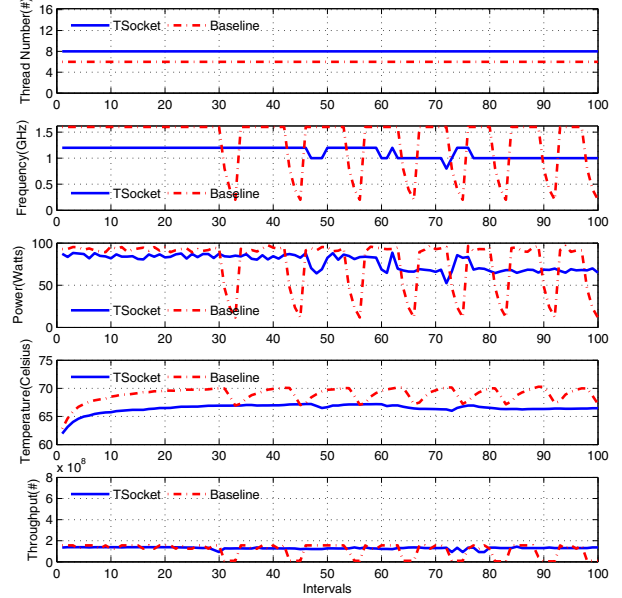

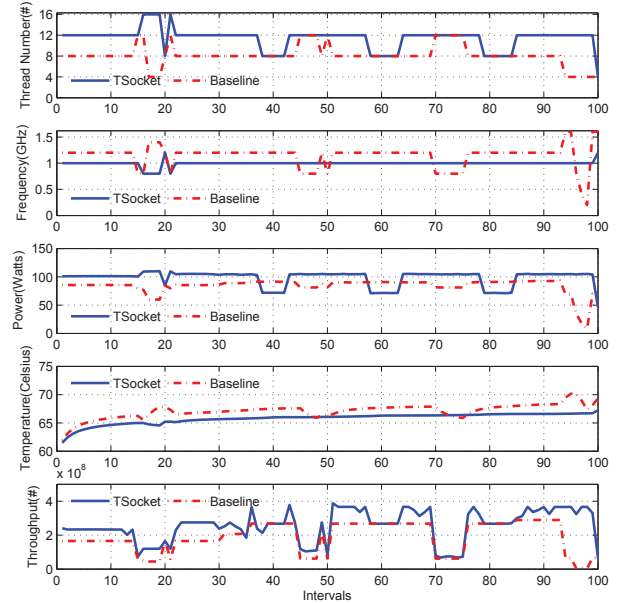**Figure 8: The blackschole traces: Baseline vs. TSocket**

Figure 9: The bodytrace traces: Baseline vs. TSocket

Performance Improvement. We apply the baseline and TSocket on all the applications in the PARSEC benchmark suit and compare the throughput between them, as shown in Fig. 10. In summary, TSocket has two advantages over the baseline: (1) Determine the safe power budgeting scheme and find the real optimal configurations. This phenomenon often occurs in scenarios when the applications are frequency sensitivity. Applications with high frequency sensitive are likely to select the configurations with higher frequency and fewer threads. Without guidance of TSocket, these applications may easily incur thermal violations because the core power is exceeding the affordable core power capacity; (2) Sniff more power budget which can be used for performance improve-
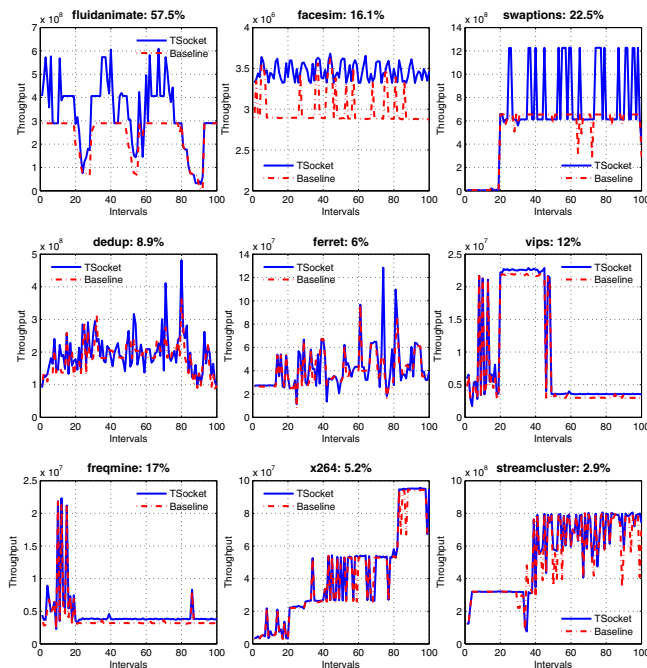
Figure 10: Performance comparison: Baseline vs. TSocket

ment. It often occurs in scenarios when the applications are both frequency sensitive and thread sensitive. In these cases, TSocket can offer more power budget for applications and result in better performance, such as the execution phases of *fluidanimate*. Therefore, the effectiveness of TSocket depends on the thread phases. There are some phases, such as interval 42 in *ferret*, TSocket performs slightly worse than the baseline. The reason is that TSocket makes conservative estimation about the power capacity which aims to ensure thermal constraints in a long time period. Sometimes a short interval of power overshoot does not incur thermal violations. Overall, our method is effective, which avoids thermal violations and achieves 19% of performance improvement.

5. CONCLUSION

For future multi/many-core processor designs, the power delivery capacity and thermal implications have become two primary design constraints. It is important to maintain high performance without violating thermal constraints. Existing power budgeting techniques and thermal/power control mechanisms do not take into account the application characteristics for power limit estimation. Consequently, we propose a thermal sustainable power capacity model, *TSocket*, which takes into account both thermal and power implications. With the assistance of TSocket, the power-performance model can choose the optimal configuration without thermal violations. Experimental results show that the TSocket is effective in thermal violation mitigation and sniffing additional power budget, leading to the performance improvement by 19% on average.

6. ACKNOWLEDGEMENT

References

[1] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, 2012.

[2] G. Chadha, S. Mahlke, and S. Narayanasamy, "When less is more (limo):controlled parallelism for improved efficiency," in *International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES)*, pp. 141–150, 2012.

[3] J. Kim and M. Horowitz, "An efficient digital sliding controller for adaptive power-supply regulation," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 37, no. 5, pp. 639–647, 2002.

[4] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core dvfs using on-chip switching regulators," in *International Symposium on High Performance Computer Architecture (HPCA)*, pp. 123–134, 2008.

[5] C. Ryan, H. Can, A. K. Coskun, and R. Sherief, "Pack and cap: Adaptive dvfs and thread packing under power caps," in *International Symposium on Microarchitecture (MICRO)*, pp. 175–185, 2011.

[6] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan, "Hotspot: a compact thermal modeling methodology for early-stage vlsi design," in *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 14, no. 5, pp. 501–513, 2006.

[7] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, "A distributed and self-calibrating model-predictive controller for energy and thermal management of high-performance multicores," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, 2011.

[8] A. B., M. C., A. T., and L. B., "Adaptive Power Control with Online Model Estimation for Chip Multiprocessors," in *Ieee transactions on parallel and distributed systems (TPDS)*, vol. 24, no. 1, pp. 170–183, 2013.

[9] X. Zhou, Y. Xu, Y. Du, Y. Zhang, and J. Yang, "Thermal management for 3d processors via task scheduling," in *International Conference on Parallel Processing (ICPP)*, pp. 115–122, 2008.

[10] X. Wang, K. Ma, and Y. Wang, "Adaptive power control with online model estimation for chip multiprocessors," in *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 22, no. 10, pp. 1681–1696, 2011.

[11] "Bidirectional application power management," in *http://www.amd.com/us/Documents/AMD_SFF-Whitepaper.pdf*.

[12] C. Isci, A. Buyuktosunoglu, C.-Y. Chen, P. Bose, and M. Martonosi, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *International Symposium on Microarchitecture (MICRO)*, pp. 347–358, 2006.

[13] M. Bhadauria, V. Weaver, and S. A. McKee, "A characterizaton of the parsec benchmark suite for cmp design," Cornell University, Tech. Rep. CSL-TR-2008-1052, 2008.

[14] "Intel core i7-800 and i5-700 desktop processor series, datasheet." in *http://www.intel.com/content/www/us/en/intelligent-systems/piketon/core-i7-800-i5-700-desktop-datasheet-vol-1.html*.

[15] I. Paul, S. Manne, M. Arora, W. L. Bircher, and S. Yalamanchili, "Cooperative boosting: needy versus greedy power management," in *Proceedings of International Symposium on Computer Architecture (ISCA)*, pp. 285–296, 2013.

[16] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," in *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[17] S. Li, J.-H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi, "Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *International Symposium on Microarchitecture (MICRO)*, pp. 469–480, 2009.

[18] L. Dagum and R. Menon, "Openmp: an industry standard api for shared-memory programming," in *IEEE Computational Science Engineering (CSE)*, vol. 5, no. 1, pp. 46–55, 1998.