Data-Driven Structured Thermal Modeling for COTS Multi-Core Processors

Seyedmehdi Hosseinimotlagh, **Daniel Enright**, Christian R. Shelton, and Hyoseung Kim
University of California, Riverside

Motivation

- Unexpected processor throttling
  - Interrupted or slowed execution of safety-critical tasks
- System design space exploration
  - Design workloads around thermal conditions and processor thermal characteristics
- Overly pessimistic or optimistic thermal budgets with inaccurate models
  - Over or under committing resources, missing deadlines.

Related Work (1/2)

- High computational cost simulations[18, 27, 11, 21, 39]
  - Require proprietary microarchitectural information
  - Require detailed power traces and floor plans
  - Hotspot-based, IR imaging based, and performance counter-based estimations[17, 33, 6]
- Application and processor fingerprinting with linear control systems[31, 32, 8, 1, 24, 23]
  - Thermal impulse response
    - Matrix pencil approach[8, 15]
  - Convolving utilization traces with impulse response vector
    - Requires detailed information about runtime behavior of tasks
    - Model changes with each application
    - Impossible to model thermal behavior with preemptive tasksets
    - Does not work with multiple cores with several tasks

Related Work (2/2)

- Thermal parameter estimation[23, 24]
  - Finding parameters for a solved LTI thermal system model
  - Use the calibrated model to estimate temperature
  - Usually high order and time/resource consuming to simulate

[1] R. Ahmed, P. Huang, M. Millen, and L. Thiele. On the design and application of thermal isolation servers. ACM Trans. Embed. Comput. Syst., 16(5s):165:1–165:19, Sept. 2017
[6] A. Das, A. Kumar, and B. Veeravalli. Reliability and energy-aware mapping and scheduling of multimedia applications on multiprocessor systems. IEEE Transactions on Parallel and Distributed Systems, 27(3):869–884, 2015
[8] T. J. A. Eguia, S. X. . Tan, R. Shen, E. H. Pacheco, and M. Tirumala. General behavioral thermal modeling and characterization for multi-core microprocessor design. In 2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010), pages 1136–1141, 2010.
[11] Y. H. Gong, J. J. Yoo, and S. W. Chung. Thermal modeling and validation of a real-world mobile ap. IEEE Design Test, 35(1):55–62, Feb 2018.
[15] Y. Hua and T. K. Sarkar. Generalized pencil-of-function method for extracting poles of an em system from its transient response. IEEE transactions on antennas and propagation, 37(2):229–234, 1989.
[17] P.-S. Huang, Q.-C. Chen, C.-W. Huang, and S.-L. Tsao. An efficient thermal estimation scheme for microprocessors. In 2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications, pages 1–10. IEEE, 2014.
[18] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. Hotspot: A compact thermal modeling methodology for early-stage vlsi design. IEEE Transactions on VLSI systems, 14(5):501–513, 2006.

[21] O. Kwon, W. Jang, G. Kim, and C. Lee. Accurate thermal prediction for nans (n-app n-screen) services on a smart phone. In 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES), pages 1–10, 2018
[23] Y. Lee, E. Kim, and K. G. Shin. Efficient thermoelectric cooling for mobile devices. In 2017 IEEE/ACM International Symposium on Low Power Electronics and Design, pages 1–6. IEEE, 2017.
[24] Y. Lee, K. G. Shin, and H. S. Chwa. Thermal-aware scheduling for integrated cpus–gpu platforms. ACM Transactions on Embedded Computing Systems (TECS), 18(5s):1–25, 2019.
[27] P. Michaud and Y. Sazeides. Atmi: analytical model of temperature in microprocessors. In Third Annual Workshop on Modeling, Benchmarking and Simulation (MoBS), volume 2, page 7, 2007.
[31] D. Rai and L. Thiele. A calibration based thermal modeling technique for complex multicore systems. In 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 1138–1143. IEEE, 2015.
[32] D. Rai, H. Yang, I. Bacivarov, and L. Thiele. Power agnostic technique for efficient temperature estimation of multicore embedded systems. In Proceedings of the 2012 international conference on Compilers, architectures and synthesis for embedded systems, pages 61–70, 2012.
[33] M. Rapp, O. Elfatairy, M. Wolf, J. Henkel, and H. Amrouch. Towards nn-based online estimation of the full-chip temperature and the rate of temperature change. In Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD, pages 95–100, 2020.
[39] A. Ziabari, J.-H. Park, E. K. Ardestani, J. Renau, S.-M. Kang, and A. Shakouri. Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices. IEEE Trans- actions on Very Large Integration (VLSI) Systems, 22(11):2366– 2379, 2014

Challenges

- Modeling Challenges
  - Dynamic thermal system
  - Many varying parameters
  - High computational cost to solve
  - Requires detailed information like power traces and detailed floorplans
  - Thermal conduction between cores
- Accuracy Challenges
  - Sensor imprecision (quantization)
  - Sensor impulse response
  - Sensor location on core
  - IP blocks
  - Changes in ambient temperature

Contributions

- Low computational cost thermal modeling scheme
  - Estimating thermal parameters of a well known LTI model
- Errors are considered in the model and eliminated in the temperature estimation
- CPU floorplan estimation scheme
- Accuracy enhancement via the ensemble of multi-frequency thermal profiles
- Case study on ARM embedded platform

System Model

- Homogeneous multi-core processor
  - Each core has a dedicated temperature sensor
- No access to:
  - the chip floorplan
  - the exact locations of on-chip temperature sensors
  - the power traces of the processor

Power Model

- Standard power model
  - summation of static and dynamic power consumption.

$$P(t) = P_S(t) + P_D(t)$$

Dynamic Power

$$P_D(t) = k_0 f(t)^s$$

Static Power

$$P_S(t) = k_1 \theta(t) + k_2$$

Temperature Model

- Common thermal model for real-time multi-core systems.
- Cannot directly find **B** or **b**
- Instead, find **B** x **P**

$$[\theta'(t)]_{n\times 1} = A_{n\times n}[\theta(t)]_{n\times 1} + B_{n\times n}[P(t)]_{n\times 1}$$

Power consumption per core

Operating Temperature

System-specific parameter matrix

Captures the effect of power consumption on temperature

Diagonal parameter matrix that considers power consumption

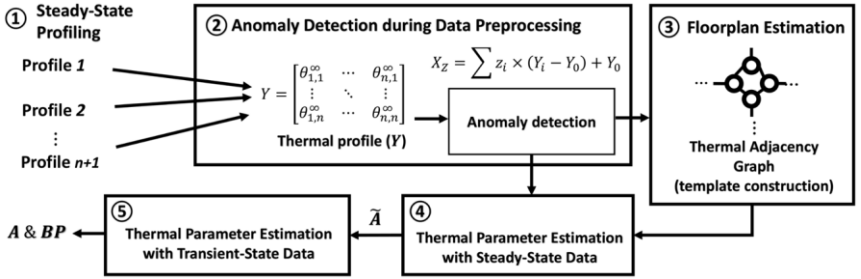
$$B_{n\times n} = b_{1\times n} I_{n\times n}$$

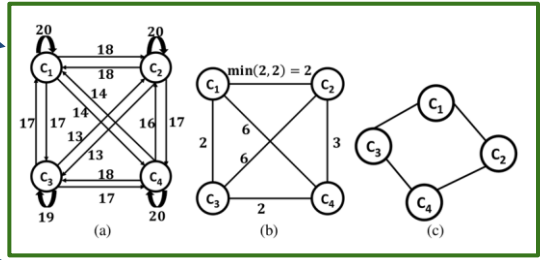
So, what is the problem?

- Given a multi-core CPU equipped with on-chip temperature sensors, construct an accurate and fast thermal RC model by estimating **A** and **B** × **P** of the CPU from a limited number of temperature profiles, without requiring prior knowledge of the cpu floorplan, information about the cooling package, and detailed power traces.

Proposed Approach

1. Gather thermal profiles
   a. n+1 profiles
2. Perform anomaly detection
   a. Superposition law for MC processors
3. Estimate CPU floorplan
   a. Reducing a fully-connected, weighted, graph
4. Estimate thermal parameters with with steady-state data
5. Calibrate thermal parameters with transient-state data





$$[\theta'(t)]_{n\times 1} = A_{n\times n}[\theta(t)]_{n\times 1} + B_{n\times n}[P(t)]_{n\times 1}$$

Thermal Profiling

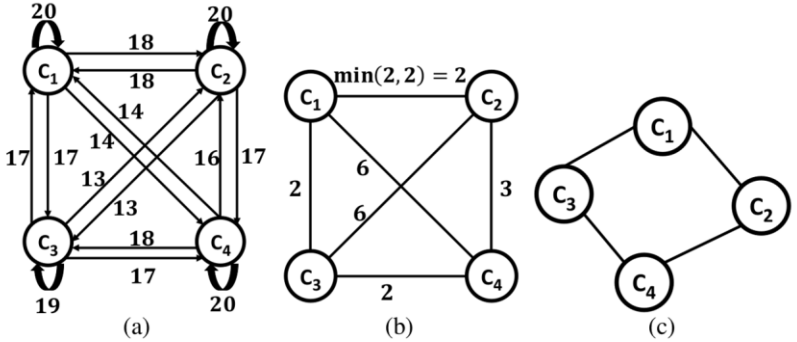
- Custom designed application
  - Executes on any combination of *n* cores
  - Utilizes a customizable percentage of cpu time
  - Logs frequency and temperature data
- Profiling needed
  - For n cores, n+1 profiles
    - One profile for each core fully utilized
    - One profile when all cores are idle
  - Can be enhanced with profiles at multiple frequencies

Data Preprocessing and Anomaly Detection

- Low-pass filter on raw sensor data
  - Eliminates the effects of temperature sensor imprecision and quantization
- Use of additional steady-state profiles
  - Apply the thermal superposition theorem validate data
  - Discard erroneous or invalid data from the thermal profile

Floorplan Estimation

- Greedy algorithm to estimate the geography of the CPU cores on the CPU die
- Reducing the graph
  - Eliminating self loops
  - Subtract directed edges from self-loop values
  - Take the minimum of that result to find new undirected edge weight



Solving the Model for Steady-state Profiles (1/2)

Equation 1: $$\left[\theta'(t)\right]_{n\times 1} = A_{n\times n}\left[\theta(t)\right]_{n\times 1} + B_{n\times n}\left[P(t)\right]_{n\times 1}$$

Solving equation 1: $$\theta(t) = \theta_{chip} + e^{(t-t_0)A}\theta_0 + \int_{t_0}^{t} e^{(t-s)A}BP(s)\,ds$$

Thermal response due to the initial temperature difference from the ambient

Non-homogeneous solution caused by the input power signal.

Steady state simplification: $$\theta(t) = \theta_{chip} + e^{(t-t_0)A}\theta_0 - A^{-1}\left(I - e^{At}\right)BP$$

$$\theta_\infty = \lim_{t\to\infty}\theta(t) = \theta_{chip} - A^{-1}BP$$

Solving the Model for Steady-state Profiles (2/2)

$[Y_i]_{n\times 1}$ ← Operating temperature matrix of the cpu cores when the i-th core is fully utilized

Y_0 ← The temperature of the CPU when all CPU cores are idle

$[Y]_{n\times n} = [Y_1 Y_2 \dots Y_n]^T$ ← The matrix of temperature profiles of the CPU in the steady state.

$$\theta_\infty = \lim_{t\to\infty} \theta(t) = \theta_{chip} - A^{-1}BP$$

$$Y - [Y_0 Y_0 \dots Y_0]^T_{n\times n} = P_D(-A^{-1}B)$$

Steady state temperature matrix

Idle Temperature matrix

$$A = -P_D b\left(Y - [Y_0 Y_0 \dots Y_0]^T\right)^{-1}$$

$$\tilde{A} = \left(Y - [Y_0 Y_0 \dots Y_0]^T\right)^{-1}$$

$$\gamma = bP_D$$

$$A = -\gamma\tilde{A}$$

Thermal Parameter Estimation with Steady-state Profiles

- **Ã** is constructed according to the estimated floorplan
- Inverse of the temperature increase due to a utilization pattern
- Gradient descent algorithm to estimate *r* parameters.

$$\tilde{A} = \left(Y - [Y_0 Y_0 \dots Y_0]^T\right)^{-1}$$

$$\underset{a_i:\ i \in [1,r]}{\operatorname{argmin}} \ ||\tilde{\mathbf{A}}^{-1} - Y||_F^2$$

Thermal Parameter Calibration with Transient-state Profiles

- Transient state profiles are key to estimating matrix **B**
- To estimate the value of **A**, we only need **γ**.
- Find **γ** by curve fitting on transient state temperature.

$$A = -\gamma\tilde{A}$$

$$\theta(t) = \theta_{chip} + e^{(t-t_0)A}\theta_0 - A^{-1}\left(I - e^{At}\right)BP$$

$$\theta(t) = \theta_{chip} + e^{(t-t_0)-\gamma\tilde{A}}\theta_0 + \left(\gamma\tilde{A}\right)^{-1}\left(I - e^{At}\right)bIP$$

$$\theta(t) = \theta_{chip} + e^{-(t-t_0)\gamma\tilde{\boldsymbol{A}}}\theta_0 + \tilde{\boldsymbol{A}}^{-1}\left(I - e^{\boldsymbol{A}(t-t_0)}\right)\boldsymbol{H}$$

$$\theta(t) = \theta_{chip} + \mathbf{V}\, e^{-(t-t_0)\gamma\boldsymbol{\Lambda}}\, \mathbf{V}^{-1}\theta_0 + \tilde{\mathbf{A}}^{-1}\left(\mathrm{I} - \mathbf{V}\, e^{-(t-t_0)\gamma\boldsymbol{\Lambda}}\, \mathbf{V}^{-1}\right)\mathbf{H}$$

H ← *n* x 1 control signal $h_i = 1$ when the i^{th} core is fully utilized

$\boldsymbol{\Lambda}$ ← Eigenvalues of $\tilde{\mathbf{A}}$

V ← Eigenvectors of $\tilde{\mathbf{A}}$

Accuracy Enhancement

$$[Y]_{n\times n} = [Y_1 Y_2 \dots Y_n]^T$$

The matrix of temperature profiles of the CPU in the steady state.

$$[D]_{m\times n} = [Z_1 Z_2 \dots Z_m]^T$$

Predicate of CPU settings.

$$U = [X_{z_1} X_{z_2} \dots X_{z_m}]^T$$

Steady state temperatures of all cpu cores with setting Z

$$Y = \left(\left(D \times D^T \right)^{-1} \times D \times U^T \right)^{-1}$$

Augmented steady state temperature profile

Use of Multi-Frequency Data Ensembles

- At a frequency f_i ,we can estimate $\tilde{\mathbf{A}}_i$ with additional thermal profiles
- **A** is invariant despite changes in frequency, hence $\boldsymbol{\gamma_i}$ can be easily determined from $\tilde{\mathbf{A}}_i$

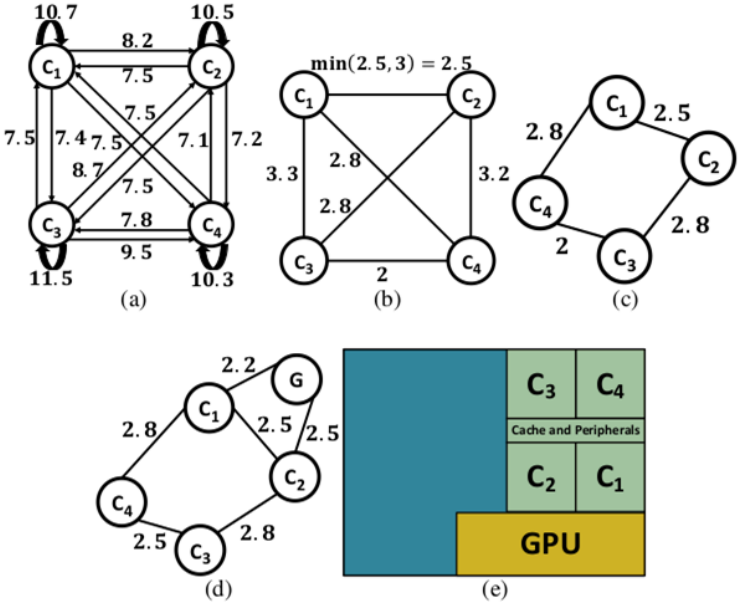
Y^i ← Temperature increase matrix at frequency f_i. γ_i denotes its power effect on temperature at that frequency.

$$A = -\gamma \tilde{A}$$

$$\underset{\substack{a_j:\, j \in [1,r] \\ \frac{\gamma_i}{\gamma_1}:\, i \in [1,|f|]}}{\operatorname{argmin}} \sum_{1}^{|f|} ||\tilde{\mathbf{A}}_1{}^{-1} - \frac{\gamma_i}{\gamma_1}\mathbf{Y}^i||_F^2$$

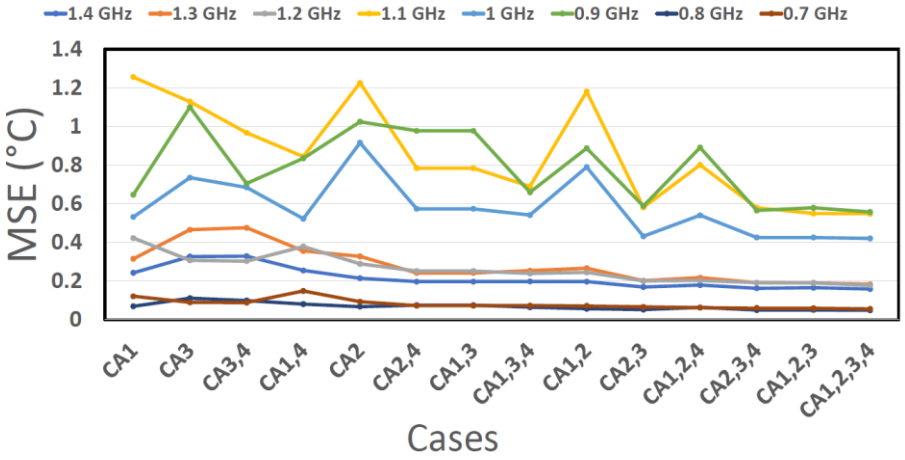
Evaluation (1/4)

- ODroid-XU4 development board equipped with a Samsung Exynos5422 SoC.
- Collected temperature profiles for all utilization configurations and multiple frequencies.
- Estimated the CPU floorplan and determined the thermal parameters **A** and **B** x **P**.
- Model estimates the maximum available utilization within 3% of ideal.



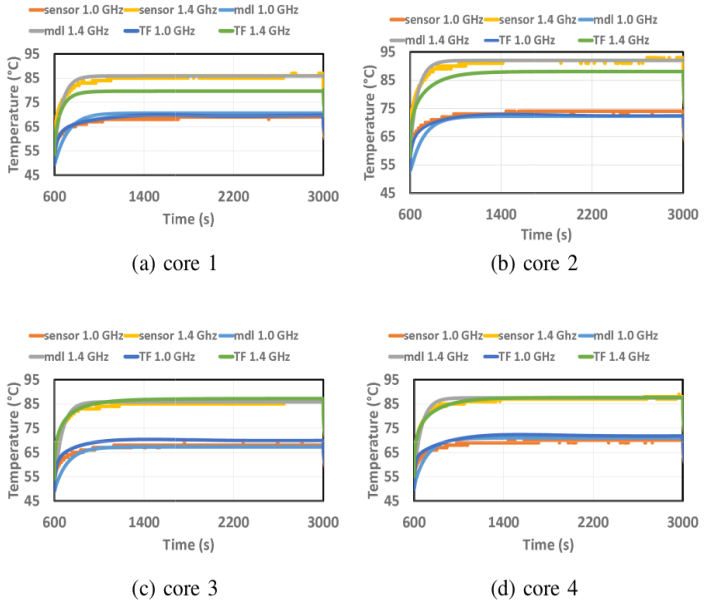
Evaluation (2/4)

- Average model error at various frequencies and with subsets of thermal profiles.
- Maximum error is below 1.5C



Evaluation (3/4)

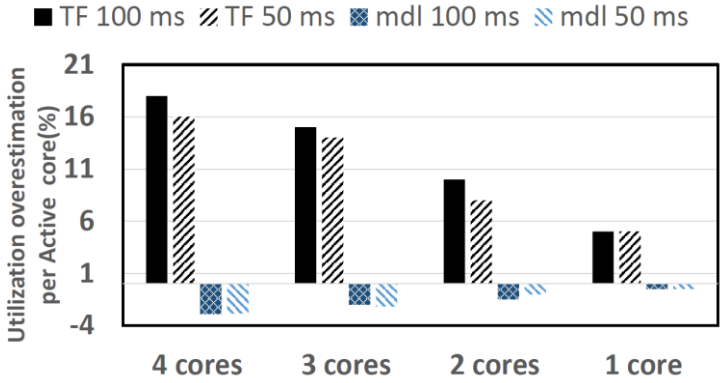
- Comparison to the state of the art[1] (referred to as TF)
  - Gathered thermal profile
  - Used Matlab's TFest function to estimate self-core transfer functions
    - Exhaustive search to optimize the number of poles and zeros
  - The modeling results of TF have the average goodness fit of 89.96% at 1.0 GHz and 88.31% at 1.4 GHz
- Our model outperforms TF in every case



[1] R. Ahmed, P. Huang, M. Millen, and L. Thiele. On the design and application of thermal isolation servers. ACM Trans. Embed. Comput. Syst., 16(5s):165:1–165:19, Sept. 2017
[13] S. Hosseinimotlagh, A. Ghahremannezhad, and H. Kim. On dynamic thermal conditions in mixed-criticality systems. In 2020 IEEE Real- Time and Embedded Technology and Applications Symposium, pages 336–349, 2020.
[14] S. Hosseinimotlagh and H. Kim. Thermal-aware servers for real-time tasks on multi-core gpu-integrated embedded systems. In 2019 IEEE Real-Time and Embedded Technology and Applications Symposium, pages 254–266, 2019.

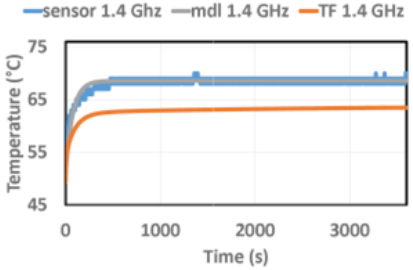
Evaluation (4/4)

- Experiment using thermal isolation servers[1, 14, 13] with replenishment period of 50ms and 100ms and the polling server policy are used.
- TF overestimates maximum available utilization (MAU) by up to 18% per core
- Our scheme underestimates MAU by a maximum of 3% per core

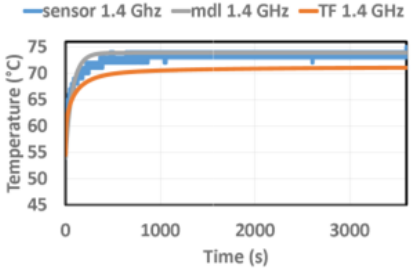


Case Study

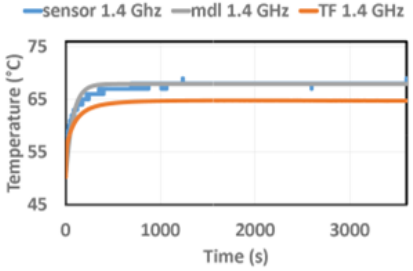
- Mixed-criticality flight management system (FMS) taskset
  - Polling server with 50ms budget and 50% utilization for low-criticality tasks on core 1
  - 4 thermal-aware periodic servers for high criticality tasks on cores 2 and 4 with replenishment budget of 50ms and utilization of 65%
- Temperature estimation
  - Our model achieves up to 2.53% error
  - TF achieves up to 23.07% error



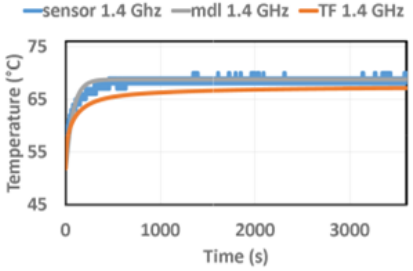
(a) core 1



(b) core 2



(c) core 3



(d) core 4

Wrapping it up

- Conclusion
  - Novel and accurate scheme for estimating thermal parameters of COTS multi-core processors for real-time embedded systems
    - Two stage estimation and calibration
      - Using steady state and transient state thermal profiles
    - Accuracy enhancement
      - Many thermal profiles with various CPU utilization configurations
      - Multiple frequency profiling
  - Model is robust and fast to converge
    - Low computational cost
    - Can be used in an event driven manner
      - Negligible memory and computational overhead
  - Estimation of power consumption
- Future work
  - Identifying thermal parameters of systems under various cooling conditions
  - Estimating thermal parameters for heterogeneous multi-core platforms
  - Statistical methods to estimating floorplan
  - Mathematical analysis of the modeling against noisy thermal profiles

Thank you!