

Towards Fast and Accurate Parallel Chip Thermal Simulations with PACT

Zihao Yuan¹, Prachi Shukla¹, Sofiane Chetoui², Carlton Knox¹, Sean Nemtzow¹, Sherief Reda², and Ayse K. Coskun¹

¹Boston University, Boston, MA - {yuan1z, prachis, cknox, snemtzw, acoskun}@bu.edu

²Brown University, Providence, RI - {sofiane_chetoui, sherief_reda}@brown.edu

Abstract—Over the last few decades, chip temperature has become one of the most important criteria for designing high-performance, cost-effective, and reliable integrated circuits (ICs). Increased power consumption and temperature not only degrade the performance of a chip, but also generate larger sub-threshold leakage power and cause reliability challenges. Therefore, thermal analysis is an essential procedure for designing any chip. Conventional thermal analysis relies on finite-element method (FEM) based multiphysics simulators (e.g., COMSOL and ANSYS). Such commercial simulators are computationally expensive and experience long solution times along with large memory requirements. These limitations make commercial simulators unsuitable for evaluating numerous design alternatives or runtime scenarios. Therefore, having fast and accurate thermal analysis is crucial for chip design and thermal optimization. In this paper, we discuss the key features of a SPICE-based PARallel Compact Thermal simulator (PACT) that achieves fast and accurate, standard-cell to architecture-level, steady-state and transient parallel thermal simulations.

Index Terms—Thermal simulation, SPICE, Compact thermal models, standard-cell level thermal simulation.

I. INTRODUCTION

Due to the increasing power density and hot spot temperatures in high-performance processors, thermal modeling and simulation have become essential procedures for designing any chip. Existing thermal simulation tools are limited by several major challenges as follows: (i) these simulators are not built to handle large and complex problems that are necessary to model and simulate standard-cell designs or emerging integration and cooling technologies; (ii) modeling and implementing emerging cooling methods is time-consuming and difficult with existing tools; (iii) existing compact and fast thermal simulators operate in architecture-level modeling space, without an easy way to gather information from standard-cell designs. This paper provides an overview a recently designed parallel compact thermal simulator, PACT [1]. PACT has been open-sourced at <https://github.com/peaclab>. PACT enables speedy and accurate standard-cell level to architecture-level chip thermal analysis, and includes the following key features:

- PACT utilizes the parallelism in modern computing systems to conduct parallel thermal simulations to speed up the process of solving problems with a large number of grid nodes (e.g., for standard-cell level problems or modeling the ultra-thin layers in a monolithic 3D stack).
- PACT offers support for various steady-state and transient solvers to speed up simulation time while maintaining the desired accuracy level (e.g., results show up to >186X

speedup compared to the commonly used tool, HotSpot [2], at a similar simulation accuracy).

- PACT can be easily extended to support emerging integration and cooling technologies by modifying the thermal netlist.
- PACT has been integrated with OpenROAD [3], an end-to-end silicon compiler. Users can evaluate the thermal behavior of full standard-cell designs from OpenROAD using PACT.
- PACT has an integrated transient thermal video generation tool, VisualPACT, that can help users better visualize the transient temperature results.

PACT aims to address the fragmentation of the thermal modeling tools space and provide a single tool that is able to conduct efficient thermal evaluation from standard-cell level to architectural level, for a variety of chip integration and cooling technologies.

II. OVERVIEW OF PACT

PACT is a SPICE-based standard-cell level to architecture level parallel compact thermal simulator. To explain how PACT works, we first go over the simulation flow of PACT and then discuss the core of PACT, which is a thermal netlist. Next, we illustrate the extensibility of PACT and the interface between PACT and OpenROAD. Following that, we elaborate on the available solvers of PACT. Finally, we discuss the compatibility of PACT and the transient thermal video generator, VisualPACT.

A. PACT Simulation Flow

An overview of the PACT simulation flow is shown in Figure 1. Users provide information of the chip stack and the cooling package to PACT. PACT then calculates the lateral and vertical thermal resistance, power consumption, and thermal capacitance for each grid and builds the thermal netlist. PACT allows users to select simulation types (e.g., steady-state and transient) as well as the solvers. Users can also run parallel thermal simulations on multiple cores and nodes via OpenMPI. PACT solves the final thermal netlist using the SPICE engine [4] and outputs the grid temperatures along with the simulation runtime and resource usage summary.

B. Thermal Netlist and SPICE Circuit Components

Similar to other compact simulators, PACT also needs to calculate the thermal resistor, capacitor, and heat flow values. However, since PACT is a SPICE-based simulator, PACT can

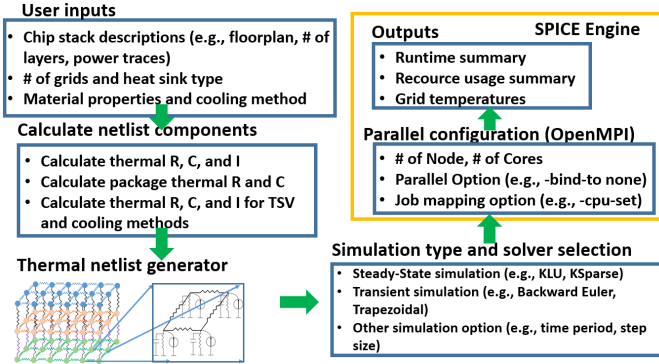


Fig. 1. PACT simulation flow.

Symbol	Component name	Equivalent terminology in PACT
	Resistor	Thermal Resistor
	Capacitor	Thermal Capacitor
	Current source	Heat flow (power)
	Voltage-controlled current source	Liquid convection in microchannel grid
	Voltage source	Assign initial temperature and ambient temperature
	PWL current source	Enable transient thermal simulation with step response or real power traces

Fig. 2. SPICE circuit component usage in PACT.

directly use the circuit components available in the SPICE library to construct the thermal netlist. To extend PACT to support emerging integration and cooling technologies, users need to add additional libraries or utility functions and modify the thermal netlist. Figure 2 shows the component symbol, component name in SPICE, and equivalent terminology in PACT. For steady-state simulation, PACT only uses resistors, voltage sources, and current sources to build the thermal netlist and conducts operating point analysis to solve the thermal netlist. For transient simulation, PACT also calculates the thermal capacitance of the corresponding grid node. For the thermal netlist with emerging cooling technologies, users need to add the circuit components from the SPICE library to model the unique cooling behavior of that cooling method. For transient thermal simulations with real power traces, PACT uses the piece-wise linear (PWL) function component and stores the power traces for each grid node in the corresponding PWL component to conduct transient analysis.

C. Extensibility of PACT

PACT offers standardized interfaces for easy integration of various compact models of emerging cooling techniques. These models are imported as python modules in PACT. All the cooling methods input parameters (e.g., liquid flow velocity, thermal resistivity, specific heat capacity, etc.) have to be specified as user inputs. Users have to create a python module (e.g., Liquid.py) to define the thermal parameters such

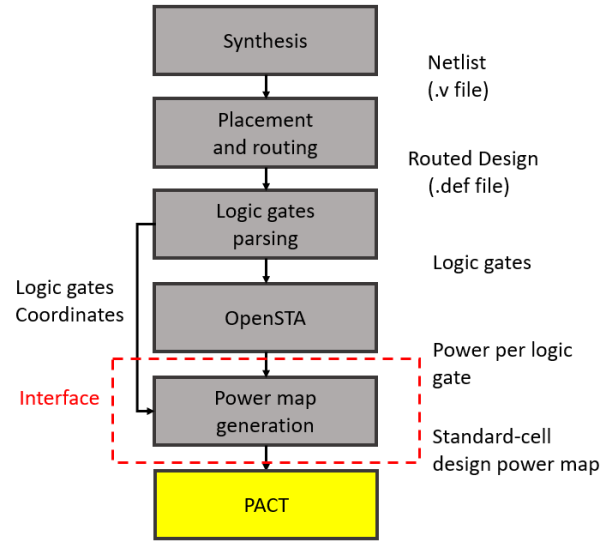


Fig. 3. Flow diagram of OpenROAD.

as thermal resistance. All these thermal parameters are then used to create the thermal netlist, where thermal resistance is modeled as electric resistors, specific heat is modeled as electric capacitors, other thermal parameters are modeled as specific electric components (see Figure 2). In addition, users also need to define the grid type (e.g., location of the virtual temperature node could be specified as center or bottom). PACT calls the appropriate cooling method library to obtain thermal parameters. Users may need to modify the thermal netlist generation process based on the cooling method. It is also possible for users to extend the SPICE library with a self-defined circuit component to support other emerging cooling technologies. Depending on the SPICE engine integrated with PACT, users can either modify the .lib file or create a new component written in Verilog-A [4].

D. OpenROAD Interface

Figure 3 shows the flow diagram of using OpenROAD [3] to generate industrial inputs for PACT. We use OpenROAD to get the spatial power information at the standard-cell level. It is a top-level RTL-to-GDS flow, which is used to generate the post-routing design exchange format (DEF) files of the circuit. The standard-cell library files (lib and lef) and the DEF files are passed to OpenSTA [3], a gate-level static timing verifier, which returns the power of each instance in the circuit. The power traces, the spatial location of each instance, and the die dimensions are used to generate the power per single grid node in the circuit, based on the pre-defined number of grids. The grid power profile and the spatial location of the grid are passed to PACT as inputs.

E. PACT Solver

Unlike other compact thermal simulators, PACT supports various steady-state solvers (e.g., KLU, SuperLU, and AztecOO) and transient solvers (such as Trapezoidal, Backward Euler, and Gear) [4]. We list the information of available solvers in PACT in Table I. These solvers make PACT

TABLE I
INFORMATION ABOUT AVAILABLE SOLVERS IN PACT. N/A STANDS OF NOT AVAILABLE.

Solver [4]	Type	Mode	Simulation type
KLU	direct	sequential and parallel	steady-state
KSparse	direct	sequential and parallel	steady-state
SuperLU	direct	sequential and parallel	steady-state
AztecOO	iterative	parallel	steady-state
Belos	iterative	parallel	steady-state
Backward-Euler	N/A	sequential and parallel	transient
Trap	N/A	sequential and parallel	transient
Gear	N/A	sequential and parallel	transient

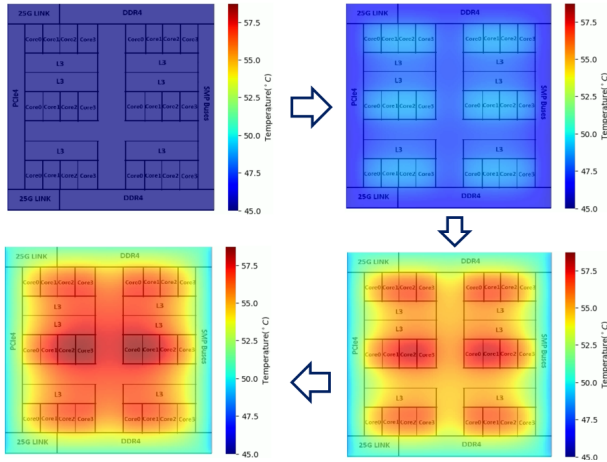


Fig. 4. The screenshots of an IBM Power9 processor thermal video generated using VisualPACT [6].

comprehensive and can be applied to solve the thermal netlists from various chip architecture designs with different simulation granularities. There are also accuracy and speed trade-offs between different solvers and different simulation modes (parallel or sequential) in PACT [4], [5]. The simulation mode, number of cores, problem size, and the solver type determine the overall accuracy and running time of the thermal simulation. Since the SPICE engine is designed from the ground up to be distributed-memory parallel, all these solvers can support parallel simulation via OpenMPI [4]. However, for existing compact thermal simulators such as HotSpot, 3D-ICE, and ThermalScope, the designers have not considered the standard-cell level simulation problem and how to utilize multicore and multiprocessor simulations through server clusters to tackle this problem. Therefore, PACT can be parallelized to achieve notable speedup when compared to existing compact thermal simulators.

F. Compatibility and VisualPACT

PACT can be integrated with architectural performance and power simulators (e.g., Sniper [7] and McPAT [8]). To let users better visualize the transient thermal behavior of target computing systems or cooling systems running realistic applications, PACT integrates a transient thermal video generation tool, VisualPACT. VisualPACT is based on OpenCV and can be used to generate .avi transient heatmap videos

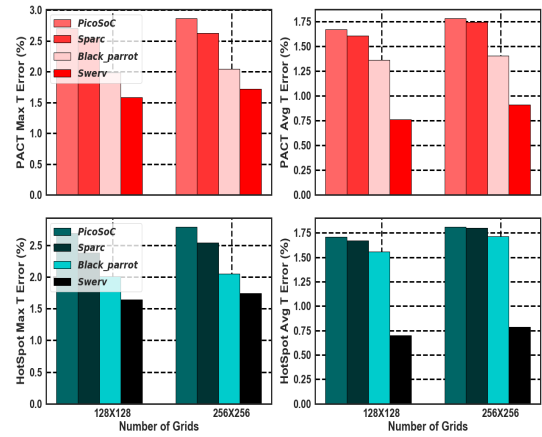


Fig. 5. Steady-state grid temperature validation results (utilization = 95%). Circuits with 95% utilization result in the highest maximum and average grid temperature error. The error is calculated with respect to COMSOL.

from transient PACT simulation temperature results. Users can specify the framerate, overlay floorplan image, temperature range, size of the heatmap, and dots per inch of the thermal video through the command line. After running a transient thermal simulation for a specific system, users can directly run VisualPACT to visualize the transient thermal behavior. We show the screenshots of an IBM Power9 processor [6] thermal video generated using VisualPACT in Figure 4.

III. VALIDATION AND SPEED ANALYSIS USING OPENROAD BENCHMARKS

To validate the accuracy of PACT, we compare the steady-state and transient simulation results to COMSOL and HotSpot using benchmark circuits from OpenROAD (PicoSoC, Sparc, Swerv, Black_parrot). Each benchmark circuit has three different utilization levels (85%, 90%, and 95%), utilization level affects the floorplan and power consumption of the circuit. The steady-state grid temperature validation results are shown in Figure 5. We observe that in comparison to COMSOL, PACT has maximum and average grid temperature errors of 2.77 % and 1.76%, respectively, which demonstrates the accuracy of PACT's steady-state simulation. The error is calculated with respect to COMSOL by dividing the grid temperature difference ($^{\circ}C$) by the maximum on-chip temperature reported by COMSOL. Figure 5 also shows the accuracy results for HotSpot with respect to COMSOL. As we see in the figure, when compared to COMSOL, PACT and HotSpot have similar maximum, average, and minimum errors.

Next, we compare the steady-state simulation time of HotSpot and PACT with various numbers of cores (8, 16, 56, and 112). We also include finer grid resolutions such as 512×512 and 1024×1024 . For parallel steady-state thermal simulations with multiple cores, we select KLU and AztecOO as PACT's solvers. The maximum steady-state simulation speedup of PACT with AztecOO compared to HotSpot is $1.83 \times$. We also run steady-state simulations using PACT with KLU. For parallel simulation using a serial solver like KLU, the thermal netlist is evaluated and assembled using

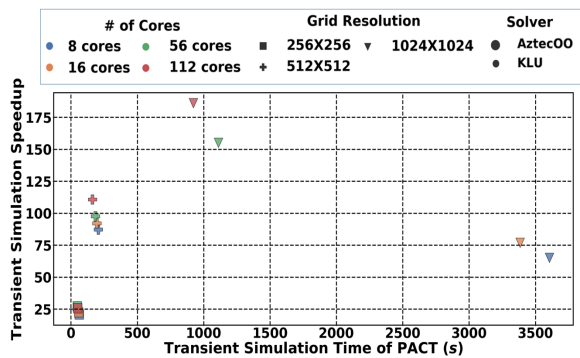


Fig. 6. Transient simulation times of PACT. The speedup of PACT against HotSpot is shown on the y-axis. The speedup is computed as the ratio of the simulation times of HotSpot and PACT.

multiple processors, but only one processor is used to solve the netlist [4]. However, AztecOO is a parallel iterative solver which uses multiple processors to evaluate, assemble, and solve the thermal netlist. PACT still achieves speedups with KLU compared to HotSpot, with a maximum speedup of $1.75\times$.

For transient validations, we create a step response for each benchmark circuits and compare the grid temperature results against COMSOL and HotSpot. We run each transient thermal simulation with a step time of 3.33 ms and the total simulation time of 99.9 ms (total steps of 30). Compared to HotSpot, PACT has a maximum and average temperature difference of 0.05% and 0.01% across all the experiments, respectively. In comparison to COMSOL, PACT has a maximum and average difference of 3.28% and 1.1% , respectively. Since OpenSTA [3] lacks dynamic power traces, we utilize the steady-state power profiles from OpenROAD and randomly apply $\pm 15\%$ additional power values for each standard cell to create synthetic transient power traces. The results are shown in Figure 7. We can see that PACT temperature traces overlap with HotSpot temperature traces. The steady-state and transient validation results indicate HotSpot and PACT are at the same accuracy level.

We then compare the transient simulation time of HotSpot and PACT with cores = 8, 16, 56, and 112. For parallel transient thermal simulations with multiple cores, we select TRAP as the solver of PACT. Figure 6 demonstrates that PACT outperforms HotSpot in every test case. PACT can achieve a speedup of up to $186\times$ when compared to HotSpot.

IV. CONCLUSION

In this paper, we discuss the key features of a SPICE-based PARallel Compact Thermal simulator (PACT) that enables fast and accurate standard-cell level to architecture-level steady-state and transient thermal simulations. PACT can be easily extended to support emerging integration and cooling technologies and is also compatible with popular architecture-level performance and power simulators. When compared to COMSOL, PACT has a maximum temperature error of 2.77% for steady-state and 3.28% for transient simulation. Compared

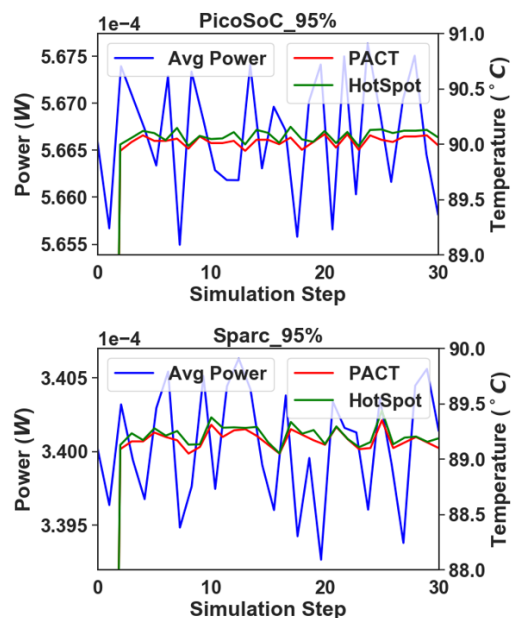


Fig. 7. Transient simulation results with synthetic power traces.

to HotSpot, PACT can achieve up to $1.83\times$ and $186\times$ speedup for steady-state and transient simulations, respectively.

ACKNOWLEDGMENT

This project has been partially funded by the NSF CRI (CI-NEW) grant #1730316/1730003/1730389.

REFERENCES

- [1] Z. Yuan, P. Shukla, S. Chetoui, S. Nemtsov, S. Reda, and A. K. Coskun, "PACT: An extensible parallel thermal simulator for emerging integration and cooling technologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.
- [2] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *IEEE Proc. of International Symposium on Computer Architecture (ISCA)*, 2003, pp. 2–13.
- [3] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem, G. Pradipta, S. Reda, M. Saligane, S. S. Sapatnekar, C. Sechen, M. Shalan, W. Swartz, L. Wang, Z. Wang, M. Woo, and B. Xu, "INVITED: Toward an open-source digital flow: First learnings from the OpenROAD project," in *Proceedings - Design Automation Conference*, 2019.
- [4] S. Hutchinson, E. Keiter, R. Hoekstra, H. Watts, A. Waters, T. Russo, R. Schells, S. Wix, and C. Bogdan, "The xyce™ parallel electronic simulator—an overview," in *Parallel Computing: Advances and Current Issues*. World Scientific, 2002, pp. 165–172.
- [5] N. J. Higham, *Accuracy and stability of numerical algorithms*. Siam, 2002, vol. 80.
- [6] S. K. Sadasivam, B. W. Thompto, R. Kalla, and W. J. Starke, "Ibm power9 processor architecture," *IEEE Micro*, vol. 37, no. 2, pp. 40–51, 2017.
- [7] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *ACM Proc. of International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, p. 52.
- [8] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *IEEE/ACM Proc. of 42nd Annual International Symposium on Microarchitecture (MICRO)*, 2009, pp. 469–480.