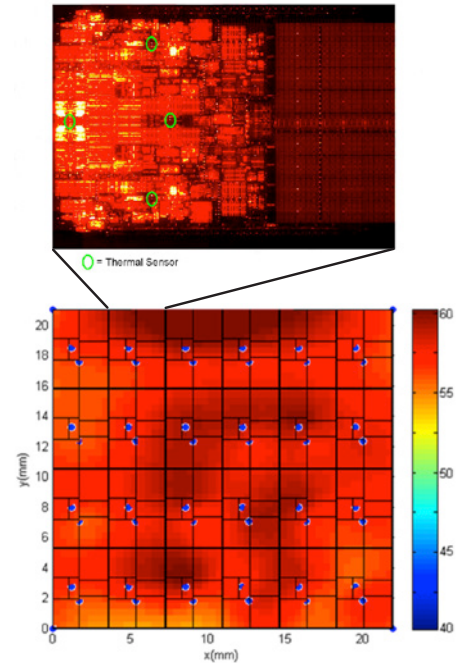


Thermal Control of Manycore and Multicore Processors

Today's high-end multicore and manycore CPUs are characterized by extreme power density and peak power consumption. The thermal dissipation systems for these processors are often designed with narrow, or even negative, margins for cost reasons. In addition, unexpected thermal emergencies may arise because of significant spatial and temporal variability of workloads, leading to nonuniform performance, power consumption, and temperature distribution. Hot-spot areas age faster since degradation effects are exponentially accelerated by high temperatures. This in turn can lead to chip damage or failure.

We are in an era of thermally limited computing. Hot-spot and thermal-runaway prevention based solely on worst-case thermal design is now unaffordable. Significant effort is thus being devoted to techniques that dynamically control the core power dissipation in a temperature-aware fashion, i.e., aiming to enforce a safe working temperature across the die surface. Today's multiprocessors include hardware support for dynamic power and thermal management, based on introspective monitors (i.e., per-core thermal/performance sensors and chipwide power gauges) and performance knobs. This infrastructure provides the sensors and the actuators for feedback control policies.

–European Research Council “Multitherman” Project

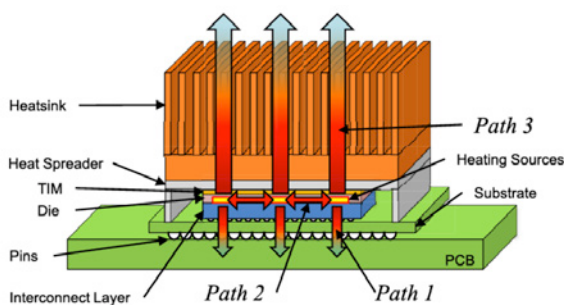


Top : Intel® Dual-Core Merom thermal map (Source: Proc. Int. Solid-State Circuits Conf., 2007); Bottom: Thermal map of the Intel single-chip cloud computer under full utilization (temperature scale in degrees Celsius)

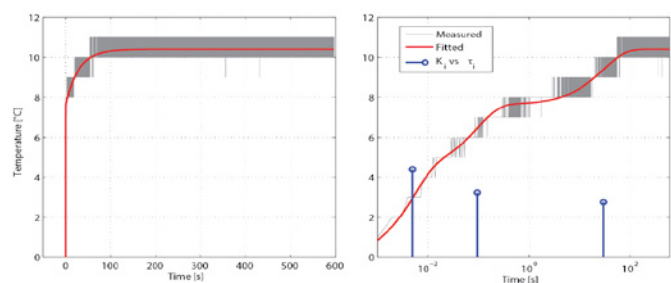
Thermal Control Challenges

Modern electronic devices have billions of transistors clocked at subnanosecond speed. Local on-die thermal transients have time constants of microseconds, whereas at the package and board level we see complex, nonlinear dynamics unfolding in seconds to minutes. A single chip can have hundreds of thermal domains that vary greatly in workload and intrinsic power density. Power consumption and heat generation are thus spatially and temporally heterogeneous, with nonlinear temperature dependency caused by leakage. In addition, the heat dissipation path is composed of different materials that lead to a multimodal time-domain response.

Accurate on-chip temperature sensors have high area cost and are affected by significant systematic and random noise. In addition, to keep post-manufacturing testing costs low, not all the sensors are accurately calibrated. Hence, manycore thermal management is a large-scale, hybrid, nonlinear multivariable control problem, affected by significant sensor, actuator, and process noise.



Typical layered chip package section and main thermal path direction and model



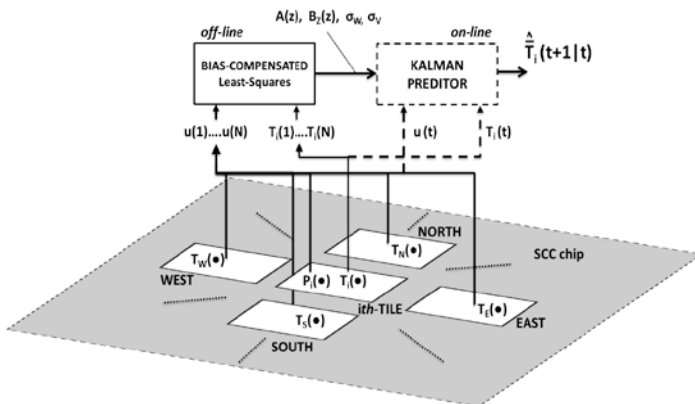
Linear (left) and logarithmic (right) transient analysis plots. The temperature is well approximated by a third-order exponential curve. The stem plot (right) shows the three time constants (4.9 ms, 94.6 ms, and 29.2 s).

Contributors: Andrea Bartolini, Andrea Tilli, and Luca Benini, Università di Bologna, Italy

Scalable, Optimal Thermal Control for Manycore Systems

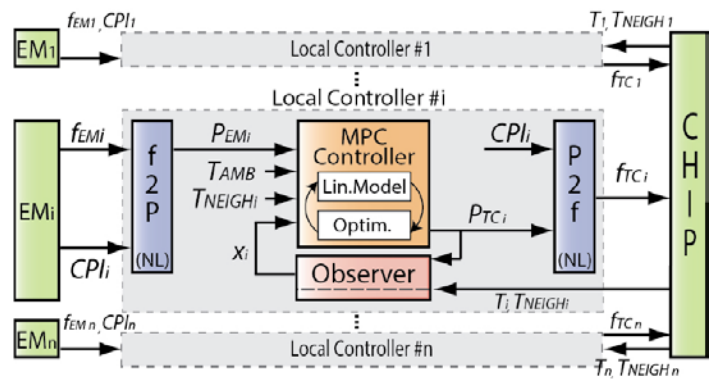
The previously highlighted challenges call for optimal control features:

- **Optimality:** The thermal controller acts (by voltage and clock scaling and shutdown) to reduce power consumption, but it must strive to minimize performance degradation while limiting temperatures to below a safe threshold across the entire silicon die. A model predictive control (MPC) approach can reduce performance degradation with respect to simple threshold-based control.
- **Predictability:** Regular workload phases can be exploited by a thermal model to predict future temperatures. This calls for system-level thermal models that relate different functional units and hardware macro block activity to the thermal map evolution.
- **Adaptability:** Fluctuations of process variations and ambient conditions (temperature, heat sink occlusion, etc.) may change the thermal behavior over the lifetime of a component. Model recalibration strategies and online system identification algorithms are required.
- **Robustness:** Thermal sensor readings are affected by significant output noise. System identification and controller design approaches are needed that are robust to measurement and process noise.
- **Scalability:** The trend toward massively parallel (100+) cores and hardware accelerators integrated on 3-D stacked dies calls for scalable control algorithms running in a few microseconds. Distributed control algorithms are needed that leverage the spatial localization of heat exchange and can exploit parallel hardware.
- **Modularity:** Thermal control not only happens at the hardware level, but it must interact with software layers such as the workload dispatcher and task scheduler.



Distributed and robust thermal model learning strategy based on ARX-plus-noise system identification. The model takes as input the core power consumption and neighbors' core temperatures. It estimates the noise variance and the model parameters suitable for the Kalman predictor. The complexity of each single model is constant as the number of cores increases. Inputs to the Kalman predictor are also the current core power consumption and neighbors' core temperatures.

Model validation: input power (dashed), measured temperature (black), and one-step-ahead predicted temperature (gray). The learned model can be used effectively at runtime to estimate the actual silicon temperature.



Distributed model predictive thermal controller. Each core executes its local control with temperature information from neighbor cores. Target frequency (f_{EM}) requests are generated by the energy manager (EM). Target power consumption (P_{EM}) is derived from f_{EM} and application properties (the nonlinear f2P function). The MPC exploits the thermal model to find the minimal power reduction that keeps the predicted temperature below a safe threshold. This value (P_{TC}) is then converted to a frequency setting (f_{TC}) through the P2f inverse function.

