TEMPERATURE AWARE SCHEDULING IN MULTIPROCESSOR SoCs

Ayse K. Coskun* Shervin Sharifi* Tajana Rosing* Keith Whisnant+ Kenny Gross*
* Computer Science and Engineering Department, University of California San Diego (UCSD)            + Sun Microsystems, San Diego

Funded by Sun Microsystems and UC MICRO Grant 06 - 198

MOTIVATION AND BACKGROUND

• Temperature Induced Problems:
  • Thermal Hot Spots
    • High leakage power
    • Slower devices
    • Degraded reliability
    • Increased interconnect resistivity
  • Thermal Cycles
    • Higher permanent failure rate
  • Spatial Gradients
    • Timing failures
    • Increased interconnect delay and IR drop

STATIC SCHEDULING

Optimal Scheduling and Allocation Using Integer Linear Programming (ILP)

Dynamic Scheduling

Low Overhead OS-Level Scheduling
• Negligible overhead in comparison to existing OS-level schedulers
• Adapts to changes in workload, power consumption and temperature

• Continuous System Telemetry®
  • Collects and analyzes time-series data using physical sensors and performance metrics
  • Advanced pattern recognition for reliability surveillance
  • Scheduler makes decisions based on the temperature measurements

CONTRIBUTIONS

Temperature-Aware Scheduling Techniques for MPSoCs:

Static:
• Optimal scheduling
• Baseline for dynamic scheduling
• Optimization for embedded systems with a priori known workload

Dynamic:
• Runtime adjustment to achieve best temporal and spatial profiles
• Negligible performance overhead

Dynamic Scheduling

Low Overhead Temperature Aware Scheduling

Adaptive-Random Policy
• Goal: Address several objectives but avoid high complexity
• Updates probabilities of sending workload based on temperature history
  • \( P_n \rightarrow \text{Evaluated at each job arrival} \)
  • \( W \) updated periodically (interval length: 1 sec, \( W = \beta / \text{Avthr} \))

Thermal Maps:
(a) Load Balancing
(b) Adaptive Random

Optimal Schedule with Minimized Hot Spots, Spatial Gradients and Thermal Cycles

Variables in the ILP

\( x_i: 1-0 \text{ variables; } x_i=1 \text{ iff task } T_i \text{ is assigned to } P_U \)

\( q_i: \text{Time spent above threshold temperature while executing task } T_i \)

\( n_{pr}: 1-0 \text{ variables; } n_{pr}=1 \text{ iff } p \text{ and } r \text{ are adjacent cores} \)

\( v_{ij}: \text{Overlap of tasks } T_i \text{ and } T_j \)

Core \( p \)

Core \( r \)

Task Graph
(Precedence, deadlines, thermal behavior)

System Properties:
• Floorplan
• Package Characteristics

ILP

Optimal Schedule

Minimizes and balances the hotspots
For a system of \( m \) cores, minimize:

\[
\text{max} \left\{ Q_j = \sum_{i=1}^{m} q_i, j \in \{1, \ldots, m\} \right\}
\]

Minimizes the spatial gradients

\[
\text{Minimize:} \sum_{p, r, T_i, T_j, v_{ij}} n_{pr} \left[ \sum_{k=1}^{n_{pr}} x_i s_k T_i + v_{ij} \right]
\]

Thermal hot spots (Without DPM)

Spatial Gradients (With DPM)

Thermal Cycles (With DPM)