Runtime Adaptation of Application Execution under Thermal and Power Constraints in Massively Parallel Processor Arrays

Éricles Sousa¹, Frank Hannig¹, Jürgen Teich¹, Qingqing Chen², and Ulf Schlichtmann²

¹Hardware/Software Co-Design, Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany
²Institute for Electronic Design Automation
Technische Universität München (TUM), Germany

Workshop on Software and Compilers for Embedded Systems (SCOPES)
St. Goar, Germany, June 1-3, 2015.
Motivation

• Here, we consider a resource-aware computing paradigm to exploit **runtime adaptation** without violating any **thermal** and/or **power** constraints in a programmable **MPPA**

[Source: Tilera Inc.]

[Source: http://www.phys.ncku.edu.tw/~htsu/humor/fry_egg.html]

[KIT's humanoid ARMAR]
A resource-aware computing paradigm

• Each application may use available computing resources in 3 phases:
  – Exploring and claiming them (invade)
  – Copying program code and data to the claimed resources as well as configuring them for parallel computing execution (infect)
  – Releasing occupied resources (retreat)

• Support for resource awareness at various levels
  – Application level
  – Runtime-system level
  – Architecture level

• Architecture consists of different compute tiles
  – General purpose CPU tiles
  – General purpose CPUs with reconfigurable fabrics
  – Programmable accelerators (MPPA)
• Motivation

• Tightly Coupled Processor Arrays (TCPAs)

• Proposed System

• Experimental Results

• Conclusion
Tightly Coupled Processor Arrays (TCPAs)

Class of massively parallel on-chip processor architectures

- Highly customizable at synthesis time
- Many parameters and configuration options
  - E.g., number of PEs, data path width

Flexibility at runtime

- Programmable, reconfigurable interconnect

Used as accelerators in MPSoC or tiled architectures for computationally intensive tasks from domains such as

- Digital signal processing, audio and video
- Linear algebra (matrix/vector computations)
- Cryptography, …
TCPAs cont’d

Processing elements (PEs)
• VLIW architecture
• Weakly programmable
  – Small instruction set and memory
  – Small register file
  – No direct main memory access

Memory structures
• Reconfigurable buffers at the borders of the array
• FIFOs at PEs’ inputs, and internal general purpose and feedback shift registers for cyclic data reuse
TCPAs cont’d

Interconnect
- Switched connections
- Single-cycle latency
- Switching possibilities can be defined at synthesis time, which enables the configuration of different interconnect topologies at runtime

Systolic Array
TCPAs cont’d

Interconnect
- Switched connections
- Single-cycle latency
- Switching possibilities can be defined at synthesis time, which enables the configuration of different interconnect topologies at runtime

2D Torus
TCPAs cont’d

Interconnect
• Switched connections
• Single-cycle latency
• Switching possibilities can be defined at synthesis time, which enables the configuration of different interconnect topologies at runtime

4D Hypercube
Designing a Customized TCPA
Challenges

• How to estimate temperature without thermal sensors?
• How to estimate power consumption in large processor arrays with potentially hundreds of PEs?
• How to satisfy application requirements considering thermal and power constraints?

Proposed Solution

• A runtime adaptation system that exploits quality/throughput tradeoffs while satisfying thermal and power constraints

• Our solution uses the invasive computing concepts. Thus, it is possible to achieve
  – Constant throughput by reducing the quality of image processing (e.g., accuracy of edge detection), or
  – Constant quality by reducing the throughput
Outline

• Motivation

• Tightly Coupled Processor Arrays (TCPAs)

• Proposed System

• Experimental Results

• Conclusion
Thermal Estimation

• Without thermal sensors, we estimate the initial temperature inside the chip by generating the floorplan of TCPA using Cadence Encounter and a 45 nm CMOS technology.

• The floorplan is the input for HotSpot and as output, we obtain the thermal conductance values between PEs and the average temperature value ($T_{initial}$) of the entire processor array.

ASIC implementation of a 4x4 TCPA using the NanGate 45 nm Open Cell Library

Estimated thermal dissipation when only 4 PEs (on top of TCPA) are executing an application.
Power Estimation

• Unlike gate-level simulation, we propose a macro-modeling for power estimation.

• We synthesized different designs (e.g., varying number of functional units (FUs) in each PE, data path width, etc.)

• Then, we annotate the power consumption of each functional unit in a table.

• Each processing element of TCPA may have a different configuration, our model takes into account individual PEs of a two-dimensional array of size $X \times Y$. The number of FUs belonging to a PE is denoted by $N_{FUs}$.

• The values of $S$ are the average switching activities of FUs, which are obtained from a cycle-accurate simulation and vary according to each application.

$$P_{avg} = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} \sum_{n=0}^{N_{FUs}-1} P_{FU_n}(S(x,y,n))$$
The configuration manager holds configuration streams containing the assembly codes of PEs as well as the interconnect topology.

The configuration manager is mainly composed of three parts:
- Hardware/software interface
- Configuration loader
- Configuration memory
Runtime Adaptation

General assumptions

• Power budget can be dynamically updated in real-time by using a management of heterogeneous dark silicon processors, for example.

• The runtime thermal variation is calculated according to a well-known thermal equation, as follows:

\[ T = \text{threshold} + (\bar{T}_{\text{initial}} - \text{threshold}) \cdot e^{-g \cdot t} \]

- \( T \): Current temperature (°C) inside the accelerator
- \( \text{threshold} \): Minimum \( T_{\text{min}} \) or maximum \( T_{\text{max}} \) temperature bounds for an application
- \( \bar{T}_{\text{initial}} \): Initial temperature
- \( g \): Thermal conductance between Pes
- \( t \): Time
Runtime Adaptation

Software

Start

\[ (T < T_{\text{max}}) \text{ and } (P_{\text{avg}} < P_{\text{budget}}) \] (yes) or \( (T \geq T_{\text{max}}) \text{ or } (P_{\text{avg}} \geq P_{\text{budget}}) \) (no)

\textit{infect}

All PEs

\textit{invade}

All PEs

No runtime adaptation

Our runtime adaptation
Runtime Adaptation

Start

yes

(T < T_{max}) and
(P_{avg} < P_{budget})

no

infect

All PEs

(T \geq T_{max}) or
(P_{avg} \geq P_{budget})

yes

no

invade

All PEs

infect

Less PEs

infect

less PEs and
retreat

unsed PEs

(T \leq T_{min}) and
(P_{avg} \leq P_{budget})

yes

no

Software

Steady-state

\begin{align*}
&\text{Start} \\
&\quad \rightarrow \frac{yes}{no} \\
&\quad \rightarrow \boxed{(T < T_{max}) \text{ and } (P_{avg} < P_{budget})} \\
&\quad \rightarrow \boxed{\begin{align*}
&\text{invade} \\
&\quad \rightarrow \boxed{\text{invade} \quad \text{All PEs}} \\
&\quad \rightarrow \boxed{\text{infect} \quad \text{All PEs}} \\
&\quad \rightarrow \boxed{\begin{align*}
&\quad \rightarrow \boxed{\text{infect} \quad \text{less PEs and}} \\
&\quad \quad \rightarrow \boxed{\text{retreat} \quad \text{unsed PEs}} \\
&\quad \rightarrow \boxed{\begin{align*}
&\quad \rightarrow \boxed{(T \geq T_{max}) \text{ or } (P_{avg} \geq P_{budget})} \\
&\quad \rightarrow \boxed{(T \leq T_{min}) \text{ and } (P_{avg} \leq P_{budget})}
\end{align*}}
\end{align*}}
\end{align*}}

\text{- No runtime adaptation} \\
\text{- Our runtime adaptation}
Runtime Adaptation

Start

Yes

\((T < T_{\text{max}})\) and \((P_{\text{avg}} < P_{\text{budget}})\)

infect

All PEs

infect

Less PEs

Yes

\((T \geq T_{\text{max}})\) or \((P_{\text{avg}} \geq P_{\text{budget}})\)

invade

All PEs

invade

Less PEs

Yes

\((T \leq T_{\text{min}})\) and \((P_{\text{avg}} \leq P_{\text{budget}})\)

Yes

No

No

No

Yes

No

No

No

No

No

Software

\begin{itemize}
  \item \textbf{No runtime adaptation}
  \item \textbf{Our runtime adaptation}
\end{itemize}

Temperature

Avg. Power

Time

infect

All PEs

infect

Less PEs and retreat

unsed PEs

SCOPES 2015 | Sousa et al. | Runtime Adaptation of Application Execution under Thermal and Power Constraints
Outline

• Motivation

• Tightly Coupled Processor Arrays (TCPAs)

• Proposed System

• Experimental Results

• Conclusion
Experimental Setup

FPGA design
- Xilinx Virtex-5 FPGA
- The architecture consists of a RISC processor and a TCPA, both connected through the ARM Advanced Microcontroller Bus Architecture (AMBA)
- The TCPA architecture is composed of a $5 \times 5$ array of VLIW processing elements running at 60 MHz

ASIC design
- Temperature and power consumption are estimated using the NanGate 45 nm technology
- Frequency: 550 MHz

Target application
- Edge Detection
- Input frame size: VGA ($640 \times 480$)

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Window Size</th>
<th>Number of PEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplace</td>
<td>5×5</td>
<td>25</td>
</tr>
<tr>
<td>Sobel</td>
<td>1×3</td>
<td>3</td>
</tr>
</tbody>
</table>
Experimental Setup

FPGA design
• Video stream Input/Output through DVI extension board
Experimental Results (FPGA Design)

Laplace $5 \times 5$

Sobel $1 \times 3$

![Graph showing execution time per frame versus number of PEs for 1x3 Sobel and 5x5 Laplace operations.](image)

- **Execution Time Per Frame (ms)**
  - 1x3 Sobel
  - 5x5 Laplace

- **Number of PEs**
  - 0 to 30

- **Level of Quality**
  - 0 to 3

Experimental Results (FPGA Design)
Experimental Results (FPGA Design)

Laplace 5 × 5

Sobel 1 × 3

Execution Time Per Frame (ms)

Number of PEs

Level of Quality

1x3 Sobel  ▲  5x5 Laplace
Experimental Results (FPGA Design)

Faster computation scenario providing 2 levels of quality and same throughput.
Experimental Results (ASIC Design)

ASIC Design

• Temperature variation inside the TCPA according to different number of PEs allocated for computation of $5 \times 5$ Laplace and $1 \times 3$ Sobel

• Temperature threshold
  - $T_{\text{max}} = 70^\circ C$
  - $T_{\text{min}} = 50^\circ C$

![Graph showing temperature variation over time for different numbers of PEs.](image)
Experimental Results (ASIC Design)

- Average power consumption of $5 \times 5$ Laplace and $1 \times 3$ Sobel using 25 and 3 PEs, respectively
  - $P_{avg} (5 \times 5 \text{ Laplace}) = 10.35 \text{ mW}$
  - $P_{avg} (1 \times 3 \text{ Sobel}) = 1.24 \text{ mW}$

- Reconfiguration overhead: $1.64 \mu s$
Outline

• Motivation

• Tightly Coupled Processor Arrays (TCPAs)

• Proposed System

• Experimental Results

• Conclusion
Conclusion

• We presented a strategy for runtime adaptation of application execution under thermal and power constraints in massively parallel processor arrays (MPPAs).

• Unlike [1], we exploit quality/throughput tradeoffs in image processing, keeping
  – (a) constant throughput by reducing the quality, or
  – (b) constant quality by reducing the throughput

• Running at 550 MHz, the reconfiguration overhead (1.64 μs) is more than 300 times faster than the execution time per frame (0.56 ms).

• The maximum power consumption of 25 PEs is 10.35 mW

Further Information

**Runtime Adaptation of Application Execution under Thermal and Power Constraints in Massively Parallel Processor Arrays**

**Contact**
Éricles Sousa  
Hardware/Software Co-Design, Department of CS  
University of Erlangen-Nuremberg  
Cauerstr. 11, 91058 Erlangen, Germany  
Email: sousa@cs.fau.de  
Phone: + 49 9131 85-67312

**Acknowledgements**
- This work is supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre “Invasive Computing” (SFB/TR 89)
- Research Training Group 1773 "Heterogeneous Image Systems"
- Brazilian National Council for Scientific and Technological Development (CNPq)

[www.invasive-computing.org](http://www.invasive-computing.org)