Load Balancing & DFS Primitives for Efficient Multicore Applications

M. Grammatikakis, A. Papagrigiou, P. Petrakis, G. Kornaros, I. Christophorakis
TEI of Crete

This work is implemented through the Operational Program “Education and Lifelong Learning”, Action Archimedes III and is co-financed by the European Union (EU Social Fund) and Greek national funds (Natl Strat. Ref. Framework 2007-2013).
Reducing Power Consumption

- Power estimation performed at behavior-, RT-, gate- and circuit-level ensures that power specifications are not violated
- We address power- and thermal-aware computing at RT- and system-level
- Dynamic frequency scaling (DFS) & thermal management policies in the presence of CPU load balancing
Experimental Framework

- System setup with \( X \) CPUs & \( M \) memories where \( X=2, M=4 \)
- Communication via Hypercube NoC
Parallel Matrix Multiplication

- System setup with $X$ CPUs & $M$ memories where $X=3$, $M=3$
- Row-Major Data Allocation on $M$ Memories
Mapping IPs Onto 8-node Binary Hypercube NoC

<table>
<thead>
<tr>
<th>3-cube</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>X=2</td>
<td>M4</td>
<td>CPU₀</td>
<td>CPU₁</td>
<td>CPU₂</td>
<td>M₀</td>
<td>M₁</td>
<td>M₂</td>
<td>M₃</td>
</tr>
<tr>
<td>X=3</td>
<td>M₃</td>
<td>M₀</td>
<td>M₁</td>
<td>M₂</td>
<td>CPU₃</td>
<td>CPU₀</td>
<td>CPU₁</td>
<td>CPU₂</td>
</tr>
<tr>
<td>X=4</td>
<td>M₃</td>
<td>M₀</td>
<td>M₁</td>
<td>CPU₄</td>
<td>CPU₀</td>
<td>CPU₁</td>
<td>CPU₂</td>
<td>CPU₃</td>
</tr>
</tbody>
</table>

- The parallel array is stored onto the shared memory banks: $M₀, M₁, ..., M_{6-x-1}$ in row major ($X$ denotes number of CPUs)
Architecture, Application and Balancing Parameters

- Queue size at network router and memory controller: 8 packets
- Router clock period: $T_{\text{router}} = 2, 4, 8, \text{ or } 16\text{ns}$,
- CPU clock period: $T_{\text{cpu}} = 4\text{ns}$,
- Memory controller clock period: $T_{\text{memory}} = 8\text{ns}$.

- Array size (representing the matrix row and column size): $N = 192$ with corresponding number of slices: Slices $= 8$; the size of each slice is determined from $\text{Slice}_\text{Size} = \frac{N}{\text{Slices}}$.

- External load applied only on $\text{CPU}_0$; this extra load (selected as $12\text{ ns}$) is an extra delay applied to each instruction run by the application work thread on $\text{CPU}_0$, causing an imbalance in the cpu loads

- Load balancing implemented with remote read/write and monitoring primitives
DFS Policies at NoC Router

<table>
<thead>
<tr>
<th>RANGE</th>
<th>EXPECTED FINISH TIME – DEADLINE</th>
<th>DFS Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>&gt; DPM_RT_HIGH</td>
<td>All Routers: Scale Up</td>
</tr>
<tr>
<td>B</td>
<td>in [DPM_RT_LOW, DPM_RT_HIGH]</td>
<td>BS or PO Policy</td>
</tr>
<tr>
<td>C</td>
<td>in [-DPM_RT_LOW, DPM_RT_LOW]</td>
<td>All Routers: No Scaling</td>
</tr>
<tr>
<td>D</td>
<td>in [-DPM_RT_HIGH, -DPM_RT_LOW]</td>
<td>BS or PO Policy</td>
</tr>
<tr>
<td>E</td>
<td>&lt; -DPM_RT_HIGH</td>
<td>All Routers: Scale Down</td>
</tr>
</tbody>
</table>

- **BufferSize** or BS policy, is based on computing a *cumulative buffer size within a time window* and comparing the sum of the sizes of all router queues.
- **PacketOut** or PO policy adjusts the frequency level of a router based on the *cumulative packet rate within a time window*.
- Both proposed DFS schemes are extended to compare a predefined expected deadline (obtained via application profiling).
- Decisions must be coordinated at system level.
NoC Power vs. Simulation Time

- Load balancing primitive challenges DFS runtime decisions by demanding high adaptivity and flexibility
Load Balance & DFS Intrusion vs Execution Time

- LB improvement: 22% to 32% with 0.0003% to 0.0113% intrusion
- DFS intrusion is 0.003% to 0.009%
- DFS PO is 7% to 25% more intrusive than BS, but increases performance
Power and Thermal Management on multi-FPGA system

- Power- and thermal- aware system
- Real **42-processor prototype with self- and remote-controlled DFS features**
- A novel adaptation of Floyd-Steinberg dithering algorithm originally used for image processing, in order to reduce power hot-spots and smooth power spikes among the different neighboring cores.
• Six Xilinx ML405 FPGAs combined
• Xilinx Virtex OPB Bus interconnect
Multi-CPU island node

- Each island performs weighted DFS adjustments of its local tasks taking into account neighbors’ provided information
Power and Temperature Results

- Power results with various PPC & MicroBlaze soft-processors setups
Contribution to SystemC Language

HSoC (Heterogeneous System-on-Chip) is an, open source, SystemC-based, cycle-accurate virtual platform of heterogeneous shared memory-based multicore SoCs.

⇒ http://sourceforge.net/hsoc
Ongoing Work

- Integrating and evaluating our primitives with other communication-intensive workloads that exhibit dynamic load variation
- Examining power- and thermal-aware computing to Xilinx Virtex-7 FPGA VC707
References

• T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers", in Proc. Design Automation Conf., 2002.
• HSoC library, see http://hsoc.sourceforge.net