ParaDIME: Parallel Distributed Infrastructure for Minimization of Energy

Oscar Palomar, Gulay Yalcin, Santhosh Kumar Rethinagiri, Adrian Cristal, Osman S. Unsal and Gina Alioto
Barcelona Supercomputing Center
DMTM, Viena, 22-01-2014
Motivation

- Call on *Minimising Energy Consumption of Computing to the Limit*
- Combine techniques, cooperation and sharing of information between layers (application, runtime, programming model, hardware, device)
  - e.g. annotations for reduced precision and high performance vs low power sections
- New elementary devices operating at the limits of minimum energy consumption
ParaDIME Consortium
The ParaDIME stack

ParaDIME Infrastructure

Data Center

Computing Node/Stack
- Application/BM
- API
- Scala
  - Actor Sched
  - AKKA
- JVM
- OS
- Simulated HW
  - Accelerators
  - Cores
  - Interconnect
- Future Devices

Computing Node/Stack
- Application/BM
- API
- Scala
  - Actor Sched
  - AKKA
- JVM
- OS
- VM
- Hypervisor
- Hyper
- Real HW

Intra Data Center Scheduler

Multi Data Center Scheduler

ParaDIME – DMTM'14
OPERATION BELOW SAFE $V_{dd}$
Introduction

- Reducing the Vdd is a well-known technique for making trade-offs between performance and power
  \[ P = C \times V^2 \times f \]
- Limited by safeguard bands to ensure correctness
- Redundancy can be used to detect and correct errors but full replication incurs significant energy overhead

- Hardware transactional memory as its ability to keep a check-pointed state is very handy for error correction

- Replicated transactions for high error detection capability
  - Selectively replicate when high reliability is required to limit the overhead
Operation Below Safe Vdd

- Automatic HW lowering of Vdd
  - When it does not degrade performance (high cache miss rate)
- SW-guided
  - High-performance vs low-power annotation
  - More aggressive
- Errors detection and correction
  - Selective replication using transactional memory
  - Combine with approximate to reduce the overhead of replication
## Error Detection Mechanisms

<table>
<thead>
<tr>
<th>Method</th>
<th>Memory Overhead</th>
<th>Processing Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMR/TMR</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Assertions/Invariants</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Encoded Processing</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Symptoms</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Error probability

- Need to study error probability as a function of Vdd
- \( PTXf = 1 - (P_{\text{ALU}})^{\text{size}} \)
- \( E_{\text{recovery}} = E_{\text{TX}} \times PTXf \times (1/(1-PTXf)) \)
- FauITM on gem5-x86
- SPLASH
Influence of Transaction Size

![Graph showing the influence of transaction size on TX Fail Probability. The x-axis represents supply voltage, and the y-axis represents transaction fail rate. Different lines represent different transaction sizes: TX_1, TX_100, TX_1000, TX_10000, and TX_100000.]
Error probability

- Need to study error probability as a function of Vdd
- Develop error models for future devices used in important processor blocks as a function of Vdd
- Near and Far Future devices
  - FinFet
  - III-V devices
- Probability of not meeting timing constraints at a given Vdd for basic blocks
- Transform into probability of error
ParaDIME: Parallel Distributed Infrastructure for Minimization of Energy

Oscar Palomar, Gulay Yalcin, Santhosh Kumar Rethinagiri, Adrian Cristal, Osman S. Unsal and Gina Alioto
Barcelona Supercomputing Center
DMTM, Viena, 22-01-2014
IMEC’s Contribution

- **Emerging devices**

- **Voltage limit**

![Diagram showing voltage limits and emerging devices](image-url)
BSC’s Contribution – Hardware Level

- Energy-Efficient Message Passing
  - Message passing microarchitecture
  - Message passing accelerator
  - Fast task switch
  - Task passing

- Operation Below Safe Vdd
  - Automatic HW lowering of Vdd
  - SW-guided (low-power annotation)
  - Error detection and correction

Approximate Computing
- Reduced precision FP (annotation)
- Reduced error detection / correction

Heterogeneous Computing
- Architectural level
- Device level
TU-Dresden & AoTerra’s Contribution – (Runtime & Data Center Level)

- **Fine grain scheduling**
  - Schedule threads and processes
- **Course grain scheduling**
  - Peak load
- **Source aware scheduling**
  - Carbon footprint (Supply sources: renewable over non renewable)
- **Energy-proportional computing**
  - Switch-off machines when load is low
- **Energy-efficient data storage and replication**
University of Neuchatel’s Contribution (Programming model)

- Efficient message passing
  - Actor Model (Scala & Akka)
- Error detection/recovery
  - Transactional memory for failure recovery
- Annotations/Metadata
  - Energy Profiles
  - Low-level annotations (e.g. safe sections for lowering Vdd, approximate data types)
- Applications
  - Future energy aware benchmarks
  - Real-world application