Runtime Application Mapping Using Software Agents

Mohammad Abdullah Al Faruque, Thomas Ebi, Jörg Henkel

Chair for Embedded Systems (CES)

Karlsruhe Institute of Technology
Overview

- Motivation
- Related Studies
- Our ADAM Approach
  - Distributed application mapping
  - Agent-based mapping algorithm
  - Cluster selection algorithm
  - Mapping inside selected cluster
- Results
- Summary
Motivation

- Projection: hundred or thousand core chip
- Multiple applications can be executed
- Hardware platform limits number of parallel executing applications
- User-behavior is unpredictable
Motivation

- Projection: hundred or thousand core chip
- Multiple applications can be executed
- Hardware platform limits number of parallel executing applications
- User-behavior is unpredictable

Exemplary timeline

User starts applications randomly

- t1
- t2
- t1
- t2
- t1
- t2
- t3
- t1
- t2
- t3
- t1
- t2
- t4
- t3
- t1
- t2
- t4
- t3
- t4

audio
video
navigation
Motivation

- Projection: hundred or thousand core chip
- Multiple applications can be executed
- Hardware platform limits number of parallel executing applications
- User-behavior is unpredictable
Motivation

User-behavior changes!

Application **mapping** is required at **runtime**

- Projection: **hundred** or **thousand core chip**
- **Number of parallel executing applications**
- **User-behavior** is **un-predictable**

Al Faruque, Ebi, Henkel

MAP2MPSoC @ 2010

http://ces.itec.kit.edu/
On-line detection of **hardware faults**!

Require **runtime application mapping**
Motivation

On-line detection of **hardware faults**!

Require **runtime application mapping**

- **Hardware faults** due to several reasons i.e. thermal hotspot
- **Reliable** system from un-reliable components
  [Shekhar borkar: DAC’07]
- Some tasks may be re-mapped
Related Studies

- **Design-time application mapping**
  - Only possible for *application-specific* design
  - ...

- **Runtime (centralized) application mapping**
  - L. T. Smit et al. EPI ’04: small task graphs, not scalable
  - Centralized mapping approach using single **Centralized Manager (CM)**
    - E. Carvalho et al. ’07: different heuristics, not scalable
    - ...

- **Task migration**
  - V. Nollet et al. DATE’05: reconfigurable hardware, task migration
  - S. Bertozzi et al. DATE’06: issues related to task migration
  - ...

Overview

- Motivation
- Related Studies
- Our ADAM Approach
  - Distributed application mapping
  - Agent-based mapping algorithm
  - Cluster selection algorithm
  - Mapping inside selected cluster
- Results
- Summary
Runtime Application Mapping
Hierarchical Approach

A **global agent** selects cluster for an incoming application; **multiple instances** possible

- Cluster-based collection of **monitoring data**
- Data containing information needed for the mapping e.g. PE resource usage or link bandwidth utilization
- Report the summarized state to the upper-level
- No communication bottleneck
- Mapping solution computation is distributed

An agent is a computational entity; small system-close task run on regular PEs: (1) responsible for resource management (2) may need memory (3) portable/migratable (4) recoverable (5) instances created/destroyed with a cluster

---

Legend:
- global agent
- cluster border
- PE
- virtual cluster is a subset of total number of tiles without fixed boundaries, can be modified at run time
- cluster agent is responsible for task mapping operations within a cluster (any PE of any PE type)
- global agent selects cluster for an incoming application; multiple instances possible

---

Al Faruque, Ebi, Henkel
MAP2MPSoC @ 2010
http://ces.itec.kit.edu/
Mapping request received

Cluster agent ↔ global agent negotiation

Search and select based on summarized data about application

Suitable cluster found: map the application to this cluster

Cluster-level application mapping

Suitable cluster not found: try migration

Legend: ● Start state ○ End state

.state hides a sub state chart (complexity)

Mapping request received

Cluster agent ↔ global agent negotiation

Search and select based on summarized data about application

Suitable cluster found: map the application to this cluster

Cluster-level application mapping

Suitable cluster not found: try migration
ADAM – Overview

- Migrate tasks until a suitable candidate cluster
- On success, tasks were mapped
- On fail, try another clusters
- Find a candidate cluster to start re-clustering
- Start re-clustering
- Finish
Overview

- Motivation
- Related Studies
- Our ADAM Approach
  - Distributed application mapping
  - Agent-based mapping algorithm
  - Cluster selection algorithm
  - Mapping inside selected cluster
- Results
- Summary
Computational Effort

Mapping Computational Effort (Fixed cluster size)
Comparison to Centralized Scheme (E. Carvalho et al.'07)

- centralised NN
- centralised MAC
- centralised PL
- ADAM (Our Scheme)

- Result comparison on a system with 2048 tiles
- \( \frac{98304}{13863} \approx 7.09 \approx 7 \text{ times} \) lower computational effort compared to Nearest Neighbor (NN)

NoC Size [Tiles]

Execution Time [Cycles]
Mapping Traffic

Traffic produced to collect the current MPSoC state

64x64 NoC: 2551/238.9 ≈ 10.7 times lower traffic in ADAM compared to a centralized schemes
An agent-based distributed application mapping is proposed

Heterogeneous MPSoC architecture is considered

Provides 7 times lower computational effort compared to Nearest Neighbor (NN) heuristics

10.7 times lower traffic produced by this mapping functionality compared to a centralized scheme

A deviation of mere 12.7% compared to exhaustive mapping algorithm
Publications on This Topic

- **T. Ebi, M. A. Al Faruque, J. Henkel**: "TAPE: Thermal-Aware Agent-Based Power Economy for Multi/Many-Core Architectures", ICCAD'09, (Received the IEEE/ACM WILLIAM J. McCALLA BEST PAPER AWARD)

- **M. A. Al Faruque, R. Krist, J. Henkel**: "ADAM: Run-time Agent-based Distributed Application Mapping for on-chip Communication", DAC'08
Thank you for attention!