Current Projects at the MIST Laboratory

Giovanni Beltrame
Polytechnique Montréal
Table of contents

1 Introduction

2 Projects
   Predictable Computing
   Self-Adaptive Systems
   Thermal Modelling
   Other projects

3 Collaboration Venues

4 Wrap-Up
Outline

1 Introduction

Projects
- Predictable Computing
- Self-Adaptive Systems
- Thermal Modelling
- Other projects

3 Collaboration Venues

4 Wrap-Up
Objectives

• Introduce the lab director
• Introduce the lab and its activities
• Present current and prospective projects
• Identify collaboration and funding venues
Giovanni Beltrame: a short bio

Education

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

Industrial and Research Experience

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)
Giovanni Beltrame: a short bio

<table>
<thead>
<tr>
<th>Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>• MS, Electrical Engineering, UIC, USA (2001)</td>
</tr>
<tr>
<td>• MTech, CEFRIEL, Milano, Italy (2002)</td>
</tr>
<tr>
<td>• Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)</td>
</tr>
<tr>
<td>• PhD, Computer Engineering, Politecnico di Milano, Italy (2006)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Industrial and Research Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>• CEFRIEL, Italy (2001-2003)</td>
</tr>
<tr>
<td>• STMicroelectronics, Canada (2004)</td>
</tr>
<tr>
<td>• European Space Agency, The Netherlands (2006-2010)</td>
</tr>
<tr>
<td>• École Polytechnique de Montréal (2010-)</td>
</tr>
</tbody>
</table>
Giovanni Beltrame: a short bio

Education

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

Industrial and Research Experience

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)

G. Beltrame – Current Projects
Giovanni Beltrame: a short bio

**Education**

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

**Industrial and Research Experience**

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)
Giovanni Beltrame: a short bio

Education

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

Industrial and Research Experience

- CEFRIEL, Italy (2001-2003)
  - STMicroelectronics, Canada (2004)
  - École Polytechnique de Montréal (2010-)
Giovanni Beltrame: a short bio

Education

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

Industrial and Research Experience

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)
Giovanni Beltrame: a short bio

**Education**

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

**Industrial and Research Experience**

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)

mistlab.ca
Giovanni Beltrame: a short bio

Education

- MS, Electrical Engineering, UIC, USA (2001)
- MTech, CEFRIEL, Milano, Italy (2002)
- Laurea, Computer Engineering, Politecnico di Milano, Italy (2002)
- PhD, Computer Engineering, Politecnico di Milano, Italy (2006)

Industrial and Research Experience

- CEFRIEL, Italy (2001-2003)
- STMicroelectronics, Canada (2004)
- École Polytechnique de Montréal (2010-)
Our research in a nutshell
The MIST Lab - mistlab.ca

- Dedicated to embedded systems for aerospace

**Activities and tools**

- Modeling and simulation: RESP (resp-sim.googlecode.com) and TRAP (trap-gen.googlecode.com)
- Thermal analysis (ICTherm - ictherm.com)
- Radiation tolerance (InFault - mistlab.ca)
- Optimization (MOMDP - mistlab.ca)

- 2 postdocs (soon), 6 PhD students, 1 MSc student, 2-3 interns
- Lab funded by NSERC, FQRNT and CAE
Outline

1. Introduction

2. Projects
   - Predictable Computing
   - Self-Adaptive Systems
   - Thermal Modelling
   - Other projects

3. Collaboration Venues

4. Wrap-Up
**Predictable Computing: Rationale**

### High Performance Data Systems

- Multi- and many-core trend
- **Unpredictable timing behaviour**
  - Complex architecture, memory hierarchy, shared resources...
  - Being conservative doesn't cut it

### Novel approaches

- **Deterministic** architectures
- **Probabilistic** architectures
- Self-adaptive systems
Deterministic architectures

- **Base Idea:** modify computer architecture to guarantee determinism

### The MERASA Approach

- Multi-Core Execution of Hard Real-Time Applications Supporting Reliability
- Specialized pipeline, bus, cache, and memory controller

### The PTIDES Approach

- Programming Temporally Integrated Distributed Embedded Systems
- Code generation + special OS
Mixed-criticality workloads

- One system shares real-time and non real-time tasks
- Same principle as partitioning
- Non real-time tasks should not affect real-time ones
- Reserved computation and communication resources for real-time tasks

Advantages

- Better use of resources
- Quality of service for real-time tasks
Deterministic components

Deterministic pipelines

- Duplicated resources (i.e. two pipelines)
- One pipeline is reserved for real-time tasks

Deterministic bus

- Time-triggered architecture
- Reserved slots for real-time tasks

Deterministic memory hierarchy

- Cache way isolation and freezing for real-time tasks
- Memory controller prioritizing real-time tasks
Predictable Computing: Probabilistic Architectures

- Allow random behaviour in pipelines and memory hierarchies
- Probabilistic analysis to predict real-time system behaviour

![Diagram showing probability distribution over time for deterministic and probabilistic architectures.](image-url)
Avantages and challenges

Avantages

- Qualification and integration cost reduction
- Software composability
  - The Integrated Modular Avionics concept remains applicable
- Approach started with the PROARTIS project (Airbus, ESA...)

Challenges

- Completely random architectures
- Definition of the mathematical tools to calculate probabilities
### Avantages and challenges

#### Avantages

- Qualification and integration cost reduction
- Software composability
  - The Integrated Modular Avionics concept remains applicable
- Approach started with the PROARTIS project (Airbus, ESA...)

#### Challenges

- Completely random architectures
- Definition of the mathematical tools to calculate probabilities
Methodology

- Modified LEON3/GRLib components
- Probabilistically analysable cache: parametric random placement
  - Address rotation + random (re)placement
  - Avoid using expensive fully associative caches
- Thermal noise random number generation
  - Already used in cryptography
- Modelling and simulation using the lab tools
- Prototyping on FPGA

Raising the Technology Readiness Level

- Small satellite mission with the prototype planned
Self-Optimizing Systems

LEON3 RTEMS
LEON3 RTEMS
LEON3 RTEMS
LEON3 RTEMS

FPGA

Tâche periodique
Modèles
MDP
RTEMS Param. Sondes
Two Motivating Examples

Complex heterogeneous distributed systems

- How to control the complexity of administration?
- How to manage global goals with variable demand?

Critical Systems

- How to guarantee soft degradation?
- What about unplanned contingencies?
Open Issues in Many-Cores Systems

Challenges

New challenges for designers and developers:

- Thermal management
- Parallelism exploitation
- Resource sharing conflicts
- Reliability and soft degradation
- ...

Interacting and not mutually exclusive!

Machine learning and AI provide tools to manage complexity
Self-Adaptive Systems

Definition

A self-adaptive computer is capable of *adapting its behavior and resources* to automatically *accomplish a given goal*, in changing environmental conditions.

Challenges for many-core systems

- Probes & parameters
- Fast algorithms
- Learn complex behaviour
Experimental Setup

Experimental Goal

- Show that learning can efficiently allocate resources
  - number of cores, frequency step...
- Such that applications deliver **user-defined performance goals**

Experimental Platform

- Adaptation manager implemented in Linux (Intel i7 quad-core)
- Heart-rate monitors for the PARSEC benchmark suite as probes
- Core selection and frequency allocation as parameters
- Two learning algorithms:
  - Q-Learning and Adaptive Dynamic Programming
Some Results: Throughput

blackscholes PARSEC managed by core allocation
Thermal Modelling of Electronic Devices
Thermal Modelling of Electronic Devices

Chip structure

Power dissipation

Thermal simulation

- Time varying

Fast and accurate simulation needed

Transient response

Temperature

55°C

40°C

25°C
Thermal Modelling of Electronic Devices

Chip structure

Power dissipation

Fast and accurate simulation needed

Temperature

- Time varying

Transient response
Thermal Modelling of Electronic Devices

Chip structure

Power dissipation

Thermal simulation

- Time varying

Fast and accurate simulation needed

Temperature

55°C
40°C
25°C

Transient response
Results:

Simulation time: 95s

- **Heatsink**
- No liquid cooling
- **Liquid cooling**

iPhone (7.4M nodes)
Hardware-aware Real-Time Code Generation

The diagram illustrates the process of hardware-aware real-time code generation. It starts with DSLs & Compilers, which generate the Meta-model compiler. The Meta-model compiler then maps and deploys the code to an Execution Platform. The Monitoring System ensures the code's performance and efficiency. The process involves collaboration between experts in Hardware and Software (HW/SW Expert) and Domain-Specific Languages (DSL Expert). The diagram also includes a Network and EADSL for realistic execution and monitoring.
System-level Cosimulation

- SystemC Kernel
- SystemC IP
- Python Wrapper
- ReSP Core
- Start, stop, pause...
- Simulation and timing control
- Wrapper Generation
- SystemC IPs
- ReSP IPs
- Simulation Controller
- Analysis Tools
- ReSP Tools
- ISS-1
- ISS-2
- ISS-n
- Interconnection Module
- Function Call Emulator
- Memory
- UART
- Simulated Environment
- Host Environment
- Registration

G. Beltrame – Current Projects
Verification of Memory Protection Constructs

Voter 1

T1_OR -> T1_AND_0 -> T1_FF_0
T1_OR -> T1_AND_2 -> T1_FF_1
T1_OR -> T1_AND_1 -> T1_FF_2

Voter 2

T2_OR

T2_AND_2
T2_And_1
T2_FF_0
T2_FF_1

T3_NOT
T3_NOR
T3_AND_0
Additional Projects

- Smart scrubbing of FPGAs for Improved Reliability
- Real-time task mapping
- Device lifetime optimization via slack allocation
Outline

1. Introduction

2. Projects
   - Predictable Computing
   - Self-Adaptive Systems
   - Thermal Modelling
   - Other projects

3. Collaboration Venues

4. Wrap-Up
Working together

- Provides access to the lab **infrastructure and know-how**
- Most background IP is open-source
- Support PhD and MSc students and benefit from their research
- Can be used to discover **new talent** for the company
Funding Opportunities 1/2

Establishing a collaboration: NSERC Engage Grants

• NSERC pays $25K for a 6-month project
• IP belongs to the industrial partner
• No cash commitment needed, ~ 100% success rate, no deadline
• Ideal to “test” a potential partner

NSERC Collaborative Grants

• NSERC adds x2 the cash amount provided by industry
• IP agreement between university and industrial partner
• 1 to 3-year project, ~ 90% success rate, no deadline
• 3 months turnaround (unless requesting more than $200K/year)
Funding Opportunities 2/2

**NSERC Strategic Grants**

- $150K/year completely funded by NSERC, 3-year projects
- IP agreement and in-kind (no cash) commitment needed
- New tech with applications in the following 10 years, ~ 25% success rate
- One call per year, deadline April 1st

**CRIAQ Projects**

- Combination of provincial and federal funding
- Up to a factor 5x of cash commitment
- Mandatory participation of two companies and two universities
- Standard CRIAQ IP agreement
- No deadline, ideal for more complex projects
Outline

1. Introduction

2. Projects
   - Predictable Computing
   - Self-Adaptive Systems
   - Thermal Modelling
   - Other projects

3. Collaboration Venues

4. Wrap-Up
Wrap-Up

- Computer Architecture for many-core real-time systems
  - Deterministic and probabilistic approaches
- Self-adaptive and self healing systems
- Software analysis
- Radiation and fault tolerance

Several opportunities for funded collaboration

- NSERC Engage to kickstart
- More venues for follow-up projects
The End

Questions?
http://mistlab.ca