Self-Adaptive Computing for Many-Core Processors

Giovanni Beltrame
Polytechnique Montréal

MPSoC - Otsu, 15-19 July 2013

Table of contents

Objectives

- Present the challenges of many-systems
- Show the advantages of self-adaptivity
- Describe a framework for self-adaptivity based on Markov Decision Processes
- Provide proof of the effectiveness of the methodology
- List possible future research directions

Two Motivating Examples

Complex heterogeneous distributed systems

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

Mobile Personal Computing

- How to manage a constantly changing environment?
- e.g. best trade-off between perceived performance and power consumption

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

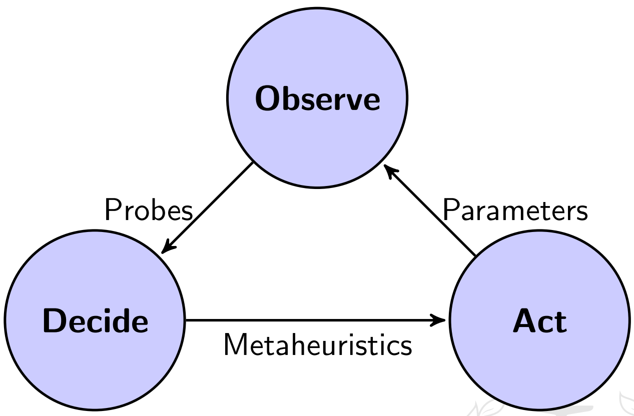
Self-Adaptive Systems

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



Autonomic Computing

Autonomic computing is about computing systems capable of managing themselves without intervention by human beings [KC03]

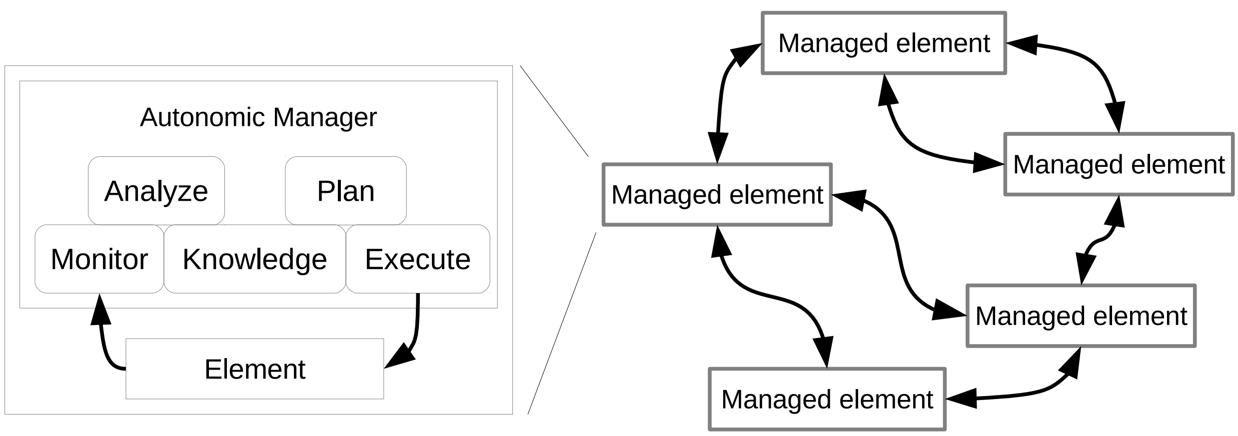
Self-managment through:

- Self-Configuration
- Self-Optimization
- Self-Healing
- Self-Protection

Distributed vs Centralized

Distributed approach: collections of autonomic elements

Related to multi-agent scenarios of artificial intelligence



Markov Decision Processes

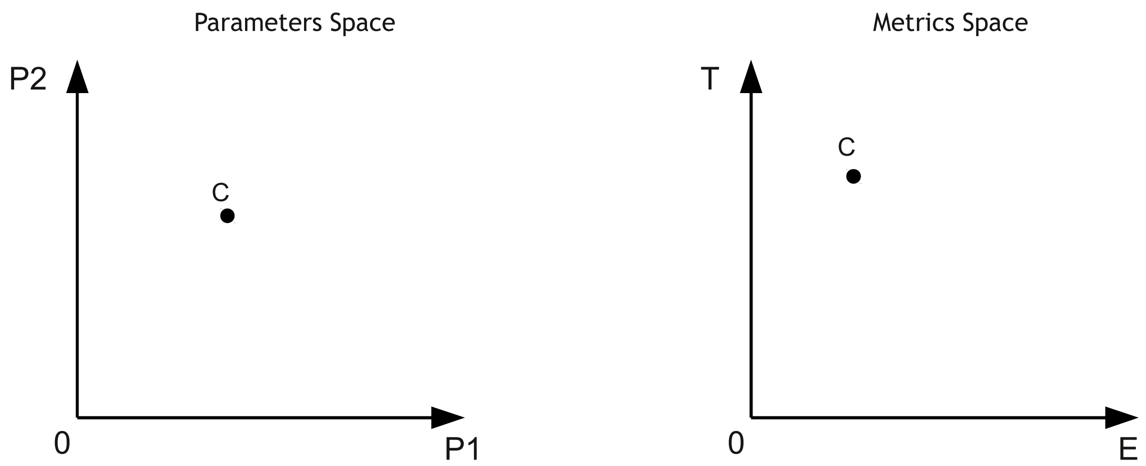
Design Space Exploration (DSE)

Determination of the *optimal configuration(s)* of a system in relation to a set of objectives

Markov Decision Processes (MDPs)

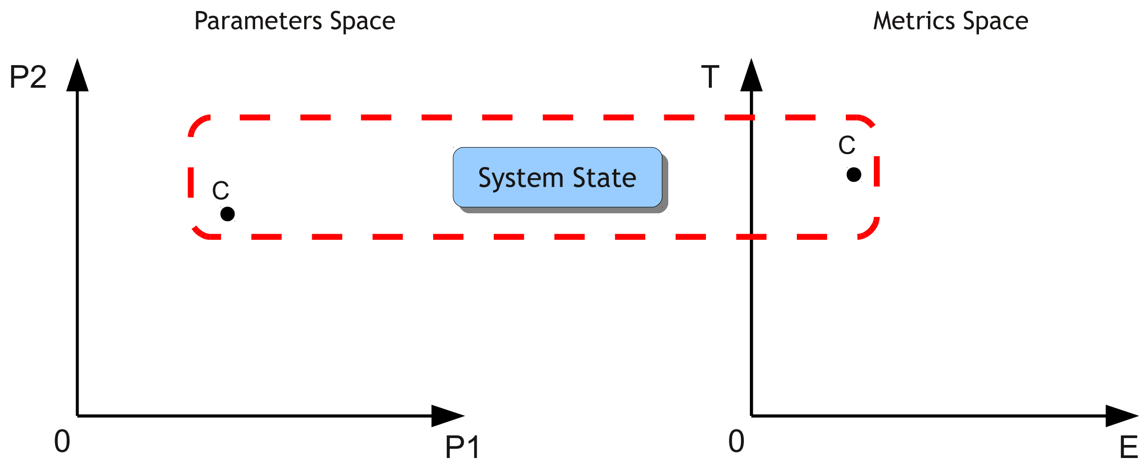
A mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker

DSE: Two Spaces



Which is the "best" parameter change?

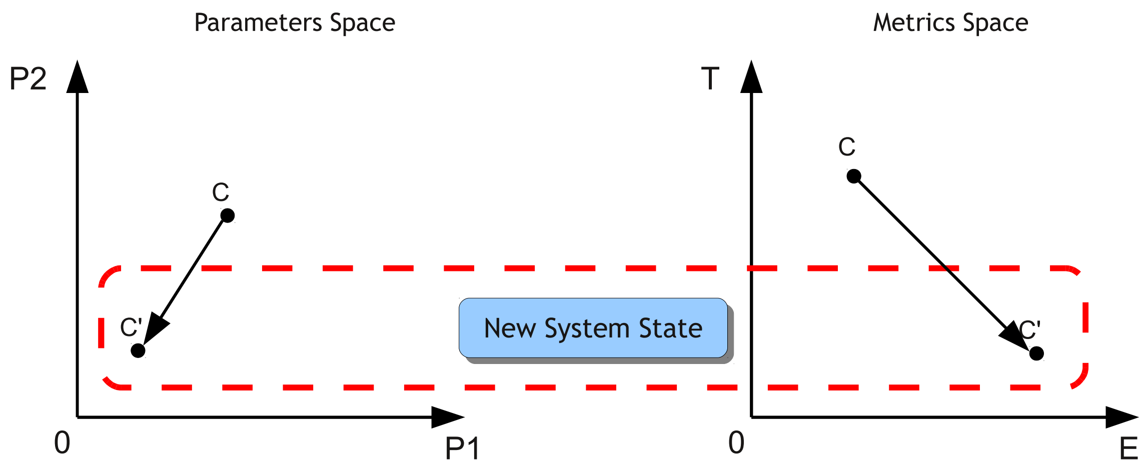
DSE: Two Spaces



Which is the "best" parameter change?

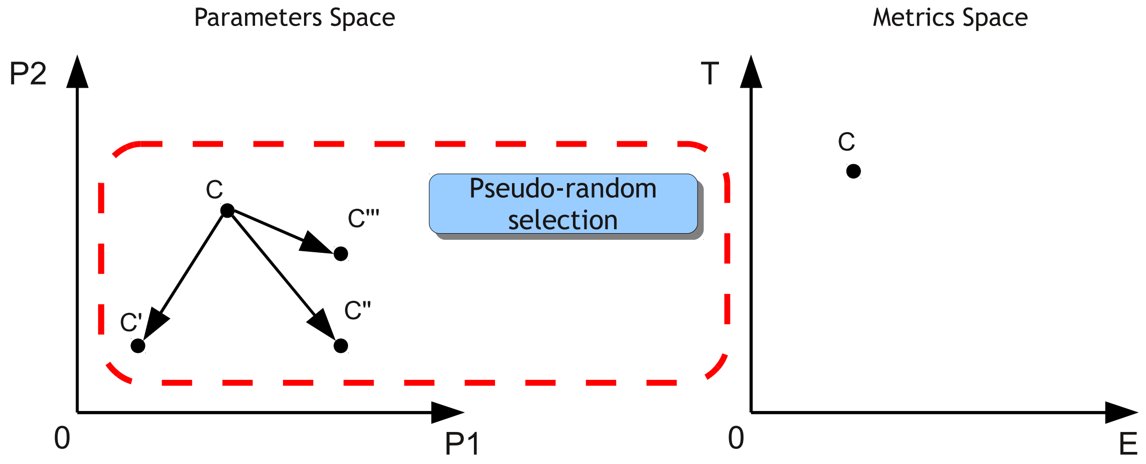
POLYTECHNIQUE MONTRÉAL Rationale Self-Adaptivity Proposed Approach Results Wrap-Up

DSE: Two Spaces



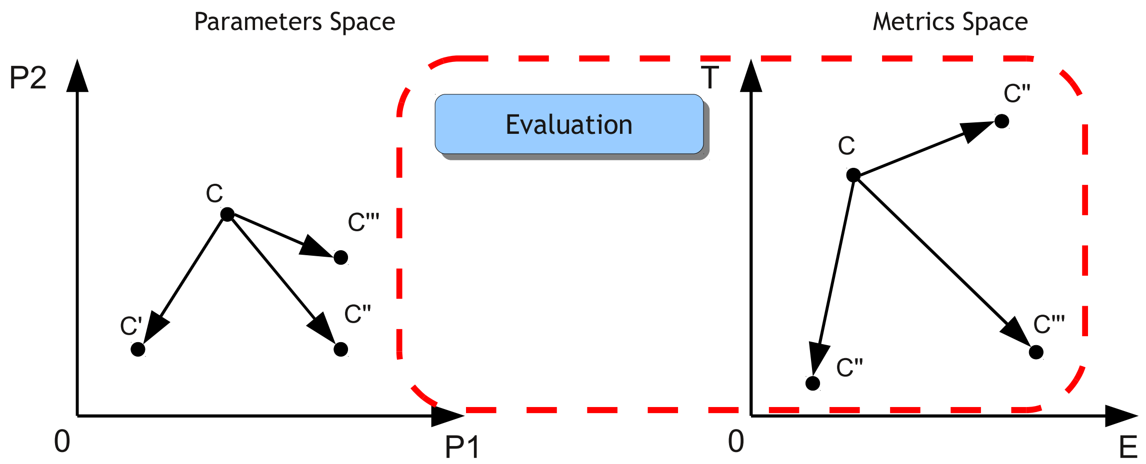
Which is the "best" parameter change?

The Pseudo-Random Approach



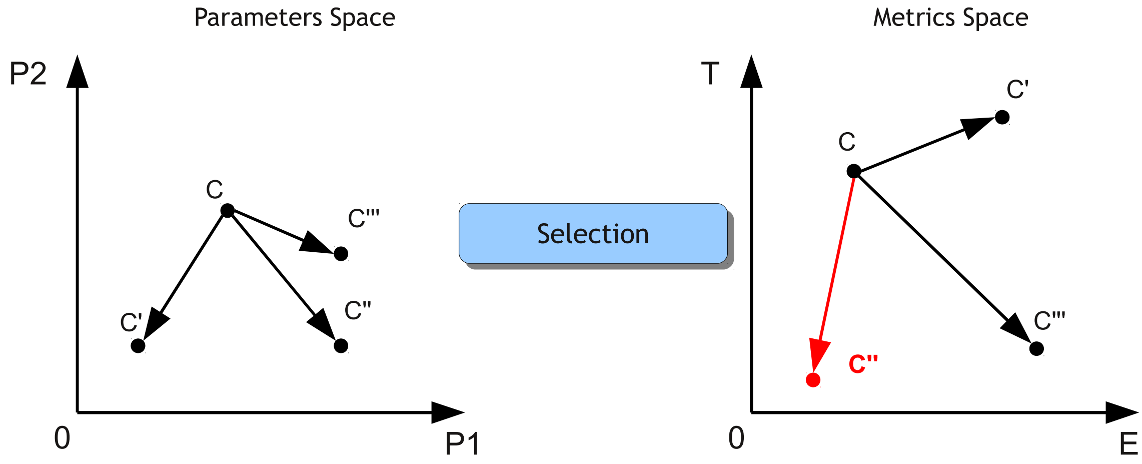
- Start from a set of configurations
- Evaluate metrics for all selected configurations (high cost)
- Choose the "best" and repeat the process
- Stop after a certain number of iterations

The Pseudo-Random Approach



- Start from a set of configurations
- Evaluate metrics for all selected configurations (high cost)
- Choose the "best" and repeat the process
- Stop after a certain number of iterations

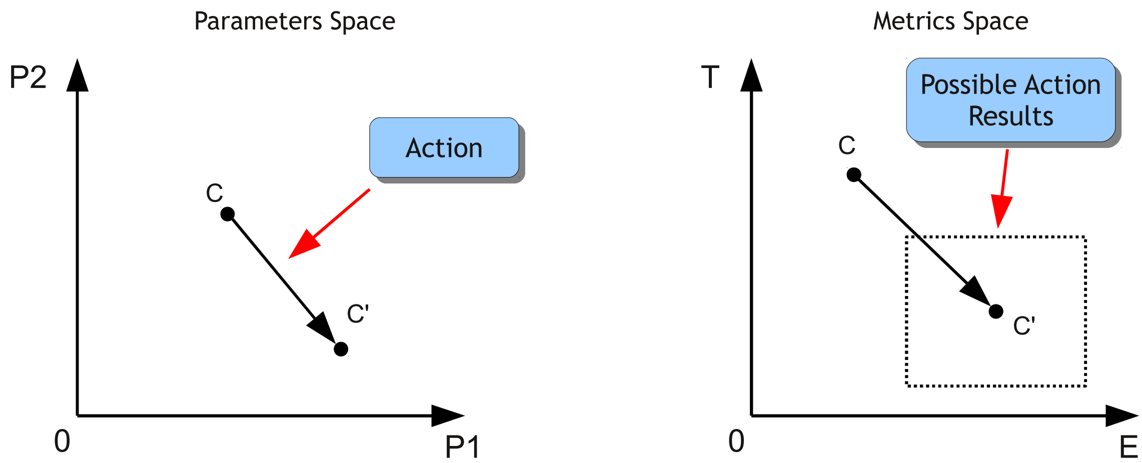
The Pseudo-Random Approach



- Start from a set of configurations
- Evaluate metrics for all selected configurations (high cost)
- Choose the “best” and repeat the process
- Stop after a certain number of iterations

The MDP Approach

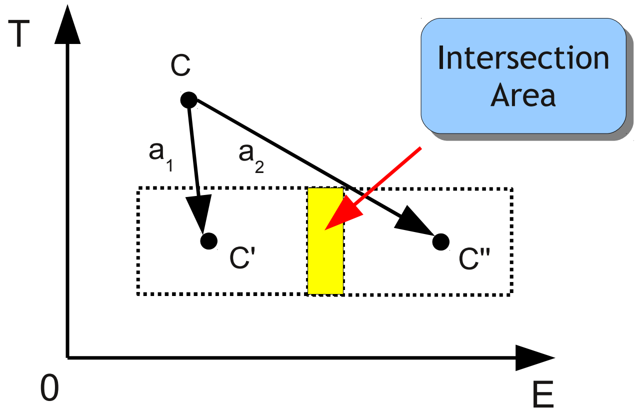
- Hypothesis: the effects of parameter changes are bounded



- How to learn the effects of parameter changes?

The MDP Approach (2)

- Hypothesis: two actions available: a_1, a_2



- Assign a probability density function to each bound
- Decision: potential improvement vs. probability of success

States and Value Functions

Optimization as a Markov Decision Process (MDP)

- A system configuration is considered a state s
- A value function associates each state to a value q
- Solving the MDP $\iff$ given an initial s, find the action sequence that brings to the "best" state s_f

Value Function Example

A variant of the energy-delay product, that can favor one of the metrics

$$q = T^{1-\alpha} E^{\alpha} \quad \text{with } \alpha \in [0, 1]$$

States and Value Functions

Optimization as a Markov Decision Process (MDP)

- A system configuration is considered a state s
- A value function associates each state to a value q
- Solving the MDP $\iff$ given an initial s, find the action sequence that brings to the "best" state s_f

Value Function Example

A variant of the energy-delay product, that can favor one of the metrics

$$q = T^{1-\alpha} E^{\alpha} \quad \text{with } \alpha \in [0,1]$$

POLYTECHNIQUE MONTRÉAL   Rationale Self-Adaptivity Proposed Approach Results Wrap-Up

States and Value Functions

Optimization as a Markov Decision Process (MDP)

- A system configuration is considered a state s
- A value function associates each state to a value q
- Solving the MDP $\iff$ given an initial s, find the action sequence that brings to the "best" state s_f

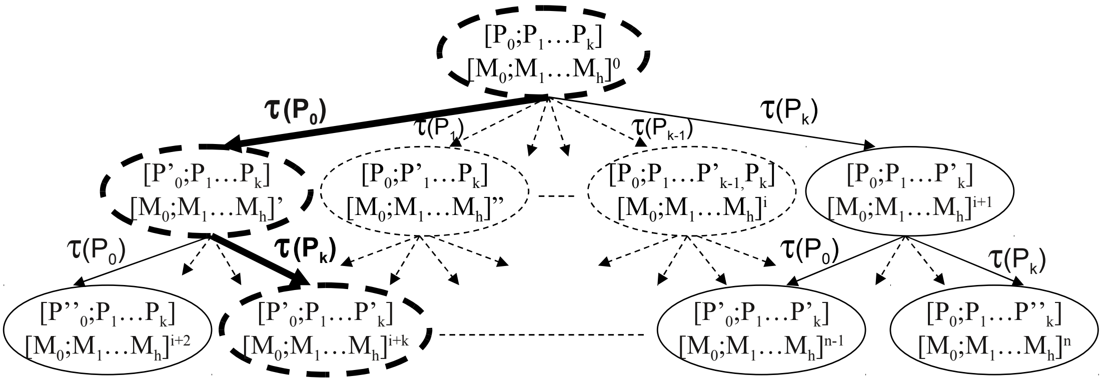
Value Function Example

A variant of the energy-delay product, that can favor one of the metrics

$$q = T^{1-\alpha} E^{\alpha} \quad \text{with } \alpha \in [0,1]$$

G. Beltrame – Self-Adaptive Computing   14/28 – mistlab.ca

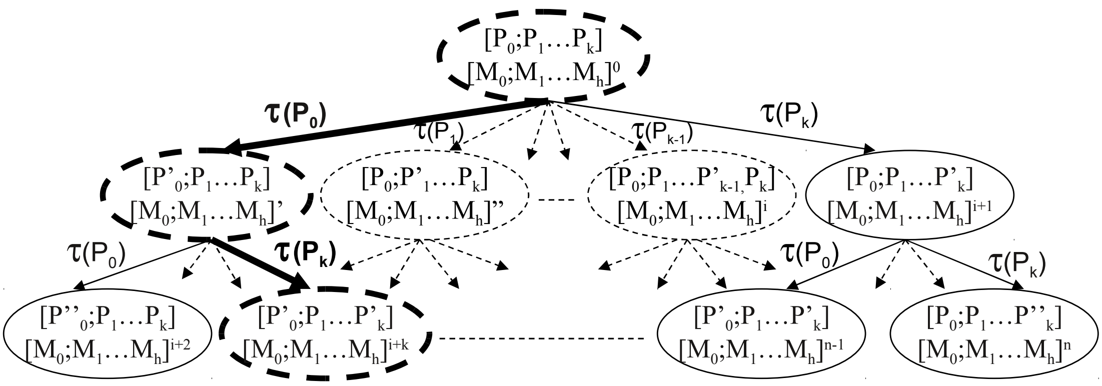
The Decision Graph



The best action sequence brings to the state with the "best" value

POLYTECHNIQUE MONTRÉAL Rationale Self-Adaptivity Proposed Approach Results Wrap-Up

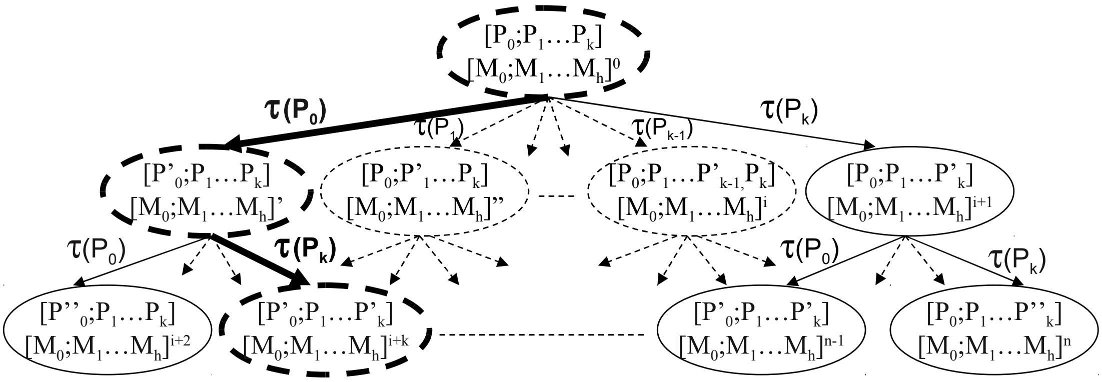
The Decision Graph



The best action sequence brings to the state with the "best" value

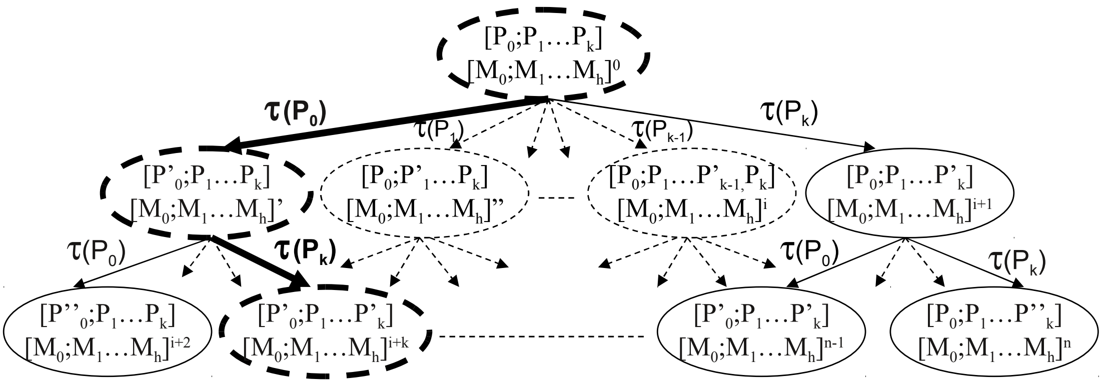
G. Beltrame – Self-Adaptive Computing 15/28 – mistlab.ca

The Decision Graph



POLYTECHNIQUE MONTRÉAL Rationale Self-Adaptivity Proposed Approach Results Wrap-Up

The Decision Graph



The best action sequence brings to the state with the "best" value

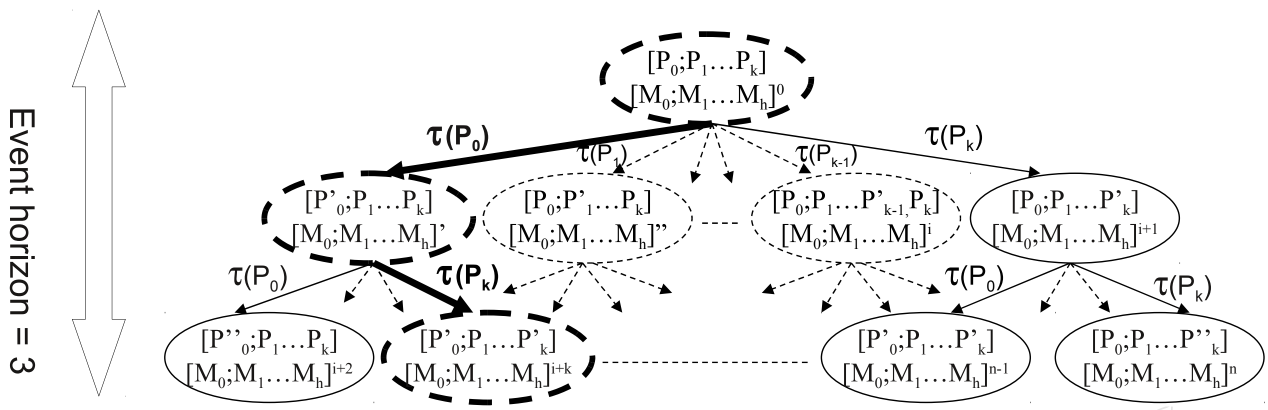
G. Beltrame – Self-Adaptive Computing 15/28 – mistlab.ca

Limiting the Graph Size

- Decision graph with many actions $\Longrightarrow$ state explosion
- Limit the depth of the tree with an event horizion I

Limiting the Graph Size

- Decision graph with many actions $\Longrightarrow$ state explosion
- Limit the depth of the tree with an event horizion I



Exploration-Exploitation Tradeoff

- Classic dilemma in learned decision making
- For unfamiliar outcomes: learning vs. exploiting knowledge
- Exploitation
  - Choose action expected to be best
  - May never discover something better
- Exploration
  - Choose action expected to be worse
  - Balanced by the long-term gain if it turns out better (Even for risk or ambiguity averse subjects)
  - nb: learning non trivial when outcomes noisy or changing

Exploration-Exploitation Tradeoff

- Tractable dynamic program in a restricted class of problems
- Solution requires balancing
  - Expected outcome values
  - Uncertainty (need for exploration)
  - Horizon/discounting (time to exploit)
- Optimal policy: Explore systematically
  - Choose best sum of value plus bonus
  - Bonus increases with uncertainty
- Intractable in general setting
  - Various heuristics used in practice

Proposed Framework

Many-Core Optimization as an MDP Problem with learning

Provide many-cores with the ability to learn how to improve their performance

A Near-Bayesian Approach

Similar to [KN09]

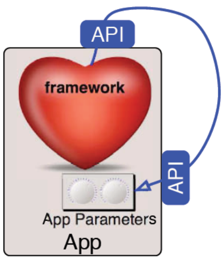
- Near-optimal w.r.t. to optimal Bayesian exploration
- Polynomial time complexity w.r.t the system parameters and the time horizon

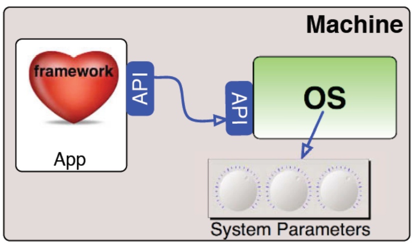
Performance Metrics (aka Probes)

State Representation

The system parameters to be monitored and controlled:

- Application throughput
- Deadlines met
- System temperature
- ...





[HES+10]

Effectors (aka Parameters)

Agent's Actions

The system parameters that need to be adjustet at run-time, e.g.:

- Scheduling policies
- Working frequency
- Degree of parallelism
- ...

Experimental Setup

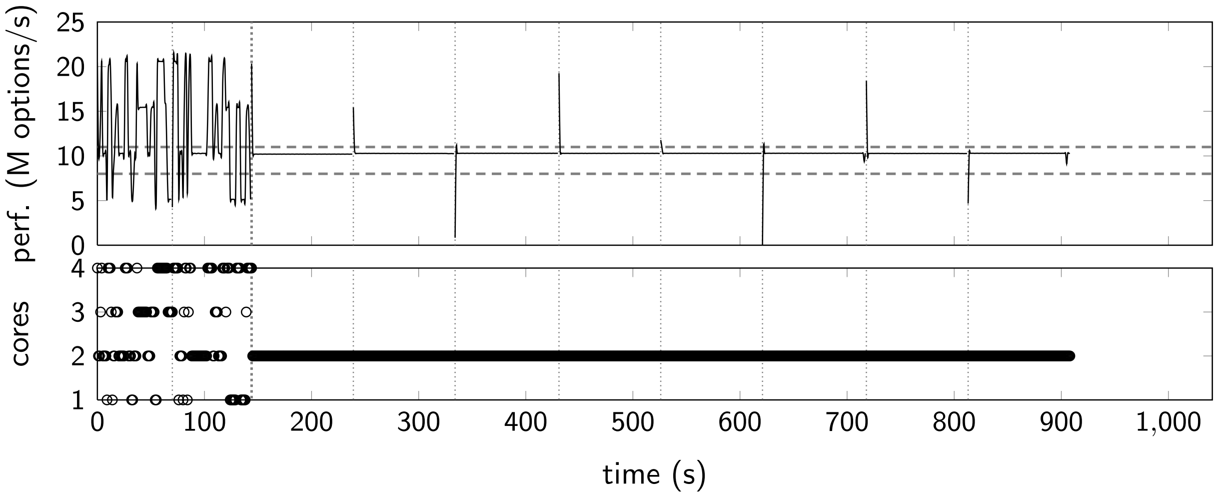
Experimental Goal

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

Experimental Platform

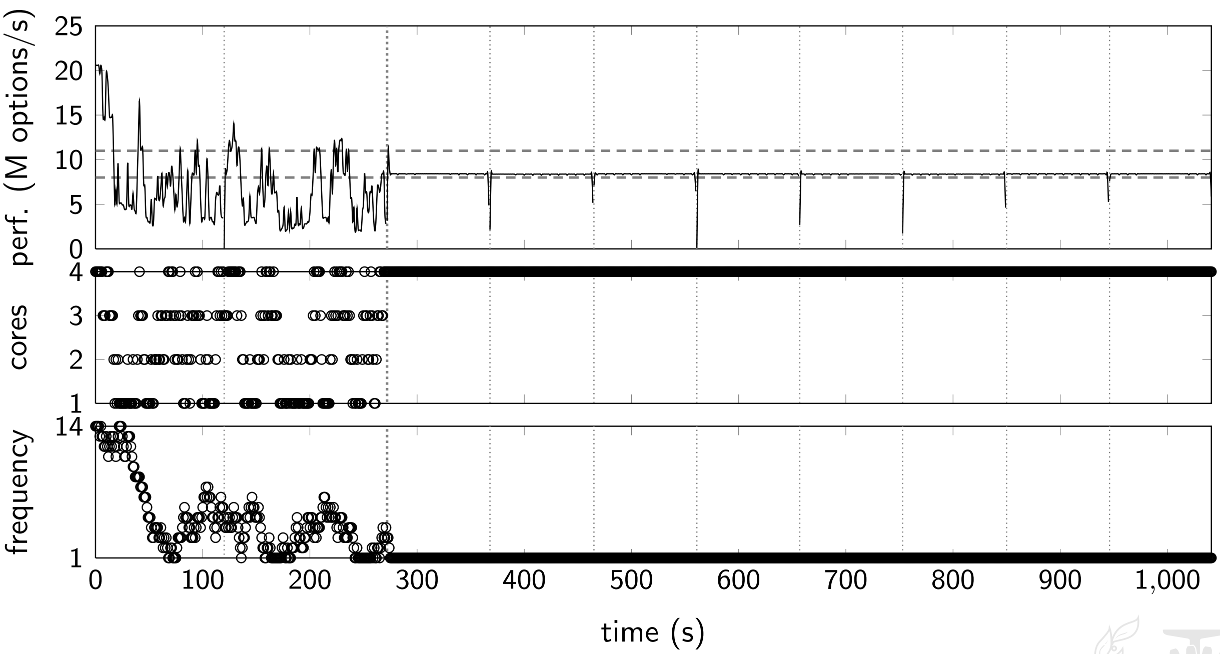
- Adaptation manager implemented in Linux (Intel i7 quad-core)
- Heart-rate monitors for the PARSEC benchmark suite as probes
- Core selection and frequency allocation as parameters
- Two learning algorithms:
  - Q-Learning and Adaptive Dynamic Programming

Some Results: Throughput [PSC+13]



blackscholes PARSEC managed by core allocation

Some Results: Throughput [PSC+13]



blackscholes PARSEC managed by core allocation and frequency scaling

Preliminary Results: Throughput [PSC+13]

Single-Agent Scenario: Results

Mean squared errors (MSE) w.r.t. desired throughputs

application	ADP (model-based) cores	ADP (model-based) cores & freq.	QL (model-free) cores	QL (model-free) cores & freq.
blackscholes	0.16	**0.11**	0.12	0.12
canneal	0.11	0.11	0.12	**0.10**
raytrace	0.17	0.17	**0.14**	0.19
swaptions	**0.10**	0.10	0.11	0.11

Some Results: Contention

Multi-Agent Scenario: Distributed Decision Making

- Measure contention over shared resources
- Developed an ad-hoc synchronization library usign heartbeats
- Expected rational outcomes:
  - Force interleaved execution of contending threads
  - Force parallel execution of non-contending threads
- Serialized execution preferred with fine-grain synchronization
- Parallel execution preferred in absence of synchronization

Some Results: Contention

Multi-Agent Scenario: Results

- Mix 1: high degree of synchronization
- Mix 2: includes both synchronizing and non-synchronizing threads
- Mix 3: has no synchronization

Execution Time	Workload mix 1	mix 2	mix 3
Unmanaged	151.25 ± 5.10	176.25 ± 2.90	216.00 ± 0.20
w/ Adapt. Manager	118.00 ± 0.70	142.50 ± 1.10	217.00 ± 0.20
Speed-Up	1.28×	1.24×	0.99×

Wrap-Up

- Provided a framework for self-optimizing autonomic systems
- Two learning algorithms to discover self-optimizing strategies
- Promising experimental results

Future Work

- More advanced strategies for the multi-agent approach
- Inclusion of time and real-time systems
- Experimentation on graceful degradation

The End

Questions?

http://mistlab.ca

References I

- Henry Hoffmann, Jonathan Eastep, Marco D. Santambrogio, Jason E. Miller, and Anant Agarwal, *Application heartbeats: a generic interface for specifying program performance and goals in autonomous computing environments*, Proceedings of the 7th international conference on Autonomic computing (New York, NY, USA), ICAC '10, ACM, 2010, pp. 79–88.
- J.O. Kephart and D.M. Chess, *The vision of autonomic computing*, Computer **36** (2003), no. 1, 41–50.
- J. Zico Kolter and Andrew Y. Ng, *Near-bayesian exploration in polynomial time*, Proceedings of the 26th Annual International Conference on Machine Learning (New York, NY, USA), ICML '09, ACM, 2009, pp. 513–520.
- J. Panerati, F. Sironi, M. Carminati, M. Maggio, G. Beltrame, P. J. Gmytrasiewicz, D. Sciuto, and M. D. Santambrogio, *On self-adaptive resource allocation through reinforcement learning*, Adaptive Hardware and Systems (AHS), 2013 NASA/ESA Conference on, 2013.