On Self-adaptive Resource Allocation through Reinforcement Learning

Jacopo Panerati†, Filippo Sironi‡, Matteo Carminati‡, Martina Maggio§, Giovanni Beltrame†, Piotr J. Gmytrasiewicz¶, Donatella Sciuto‡ and Marco D. Santambrogio‡

†Polytechnique Montréal, ‡Politecnico Milano, §Lund University, ¶University of Illinois Chicago
Rationale

Methodology

(1) Reinforcement Learning (RL).

Objective

(2) Self-adaptive Computing.

Research Question

Is RL a suitable approach for self-adaptive computing?
## Rationale

### Methodology

1. Reinforcement Learning (RL).

### Objective


### Research Question

Is RL a suitable approach for self-adaptive computing?
Rationale

Methodology

(1) Reinforcement Learning (RL).

Objective

(2) Self-adaptive Computing.

Research Question

Is RL a suitable approach for self-adaptive computing?
Rationale

Methodology

(1) Reinforcement Learning (RL).

Objective

(2) Self-adaptive Computing.

Research Question

Is RL a suitable approach for self-adaptive computing?
Rationale

Methodology

(1) Reinforcement Learning (RL).

Objective

(2) Self-adaptive Computing.

Research Question

Is RL a suitable approach for self-adaptive computing?
Rationale

Methodology

(1) Reinforcement Learning (RL).

Objective

(2) Self-adaptive Computing.

Research Question

Is RL a suitable approach for self-adaptive computing?
A Typical Machine Learning Problem

Generic (Informal) Steps

• given a (labelled or unlabelled) training set $D \subseteq \mathbb{R}^d$
• pick, from hypotheses set $H$, a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ (or $C$)
• such that, given a new data-point $X \in \mathbb{R}^d$, $f(X)$ is the actual label of $X$
A Typical Machine Learning Problem

Generic (Informal) Steps

- given a (labelled or unlabelled) training set $\mathcal{D} \subseteq \mathbb{R}^d$
- pick, from hypotheses set $\mathcal{H}$, a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ (or $C$)
- such that, given a new data-point $X \in \mathbb{R}^d$, $f(X)$ is the actual label of $X$
A Typical Machine Learning Problem

Generic (Informal) Steps

- given a (labelled or unlabelled) training set $\mathcal{D} \subseteq \mathbb{R}^d$
- pick, from hypotheses set $\mathcal{H}$, a function $f : \mathbb{R}^d \to \mathbb{R}$ (or $\mathcal{C}$)
- such that, given a new data-point $X \in \mathbb{R}^d$, $f(X)$ is the actual label of $X$
A Typical Machine Learning Problem

Generic (Informal) Steps

• given a (labelled or unlabelled) training set $\mathcal{D} \subseteq \mathbb{R}^d$
• pick, from hypotheses set $\mathcal{H}$, a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ (or $C$)
• such that, given a new data-point $X \in \mathbb{R}^d$, $f(X)$ is the actual label of $X$
A Typical Machine Learning Problem

Generic (Informal) Steps

- given a (labelled or unlabelled) training set $\mathcal{D} \subseteq \mathbb{R}^d$
- pick, from hypotheses set $\mathcal{H}$, a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ (or $\mathbb{C}$)
- such that, given a new data-point $X \in \mathbb{R}^d$, $f(X)$ is the actual label of $X$
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed be fixed
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed to be fixed
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed be fixed
Example of Classification Problem

Hand-Writing

Recognition of hand-written digits is a typical classification problem. Data-points are matrices of pixels ($\in \mathbb{R}^d$) and the label set $C$ is \{0,1,2,\ldots,9\}.

012456789 012456789 012456789
Example of Classification Problem

Hand-Writing

Recognition of hand-written digits is a typical classification problem.
Data-points are matrices of pixels ($\in \mathbb{R}^d$) and the label set $C$ is \{0,1,2,..,9\}. 

012456789 012456789 012456789
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed to be fixed
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed be fixed
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set $C$

Regression Algorithms
when labels are known to belong to $\mathbb{R}$

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality $K$ is assumed be fixed
Machine Learning Methodologies

Supervised Learning

Classification Algorithms
when labels are known to belong to a finite set \( C \)

Regression Algorithms
when labels are known to belong to \( \mathbb{R} \)

Unsupervised Learning

Clustering Algorithms
when labels are unknown but their cardinality \( K \) is assumed be fixed
**Example of Clustering Problem**

**Space Exploration**

Clustering algorithms can be used to identify patterns in remotely (e.g. in space) sensed data and improve the scientific return by sending to the ground station only statistically significant data [1].

\[\text{http://nssdc.gsfc.nasa.gov/}\]
Example of Clustering Problem

Space Exploration

Clustering algorithms can be used to identify patterns in remotely (e.g. in space) sensed data and improve the scientific return by sending to the ground station only statistically significant data [1].

1http://nssdc.gsfc.nasa.gov/
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

**Definition**

In behavioural psychology, reinforcement consists of the *strengthening* of a behaviour associated to a stimulus through its repetition.

**Pioneers**

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a **stimulus** through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Reinforcements in Behavioural Psychology

Definition

In behavioural psychology, reinforcement consists of the strengthening of a behaviour associated to a stimulus through its repetition.

Pioneers

B.F. Skinner (1904-1990), together with E. Thorndike (1874-1949), is considered to be one the fathers of current theories on reinforcement and conditioning [2].
Pavlov’s Dog

A precursor of Skinner theories

Ivan Pavlov (1849-1936) made conditioning famous with his experiment of drooling dogs.
Pavlov’s Dog

A precursor of Skinner theories

Ivan Pavlov (1849-1936) made conditioning famous with his experiment of drooling dogs.

![Image of a child with puppies](image-url)
reinforcement learning in computer science is something a bit different both from supervised/unsupervised learning and reinforcements in behavioural psychology.
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point → label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus → behaviour

Reinforcement Learning

state of the world → action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point → label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus → behaviour

Reinforcement Learning

state of the world → action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point → label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus → behaviour

Reinforcement Learning

state of the world → action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (I)

Supervised/Unsupervised Machine Learning

data-point $\rightarrow$ label (or a cluster)

Reinforcements in Behavioural Psychology

stimulus $\rightarrow$ behaviour

Reinforcement Learning

state of the world $\rightarrow$ action
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world → action → new state of the world → action → ..

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Reinforcement Learning

state of the world → action → new state of the world → action → ..

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world $\rightarrow$ action $\rightarrow$ new state of the world $\rightarrow$ action $\rightarrow$ . . .

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world $\rightarrow$ action $\rightarrow$ new state of the world $\rightarrow$ action $\rightarrow$ ..

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world $\rightarrow$ action $\rightarrow$ new state of the world $\rightarrow$ action $\rightarrow$ ..

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world → action → new state of the world → action → ..

Because the performance metric of RL (i.e., the collected reward $S$)

is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (II)

Reinforcement Learning

state of the world $\rightarrow$ action $\rightarrow$ new state of the world $\rightarrow$ action $\rightarrow$ ..

Because the performance metric of RL (i.e., the collected reward $S$) is computed over time, solving a RL problem allows to make

- planning
- complex, sequential decisions
- even counterintuitive decisions
Why Reinforcement Learning is Different (III)

If today was a sunny day

- a classification algorithm would label it as “go to the seaside”
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean *carpe diem* methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*

- Lucius Annaeus Seneca
Why Reinforcement Learning is Different (III)

If today was a sunny day

- a classification algorithm would label it as “go to the seaside"
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean carpe diem methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*

- Lucius Annaeus Seneca
Why Reinforcement Learning is Different (III)

If today was a sunny day

- a classification algorithm would label it as “go to the seaside”
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean carpe diem methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*

- Lucius Annaeus Seneca
Why Reinforcement Learning is Different (III)

If today was a sunny day

- a classification algorithm would label it as “go to the seaside”
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean *carpe diem* methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*

- Lucius Annaeus Seneca
If today was a sunny day

- a classification algorithm would label it as “go to the seaside”
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean *carpe diem* methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*
- Lucius Annaeus Seneca
Why Reinforcement Learning is Different (III)

If today was a sunny day

- a classification algorithm would label it as “go to the seaside”
- RL would tell you “you might as well study and enjoy the fact that you did not fail your exams later in the summer”

RL is not an epicurean *carpe diem* methodology, but a more farsighted and judicious approach.

*The point is, not how long you live, but how nobly you live.*

- Lucius Annaeus Seneca
moving on to self-adaptive computing..
### Typical Properties of Self-adaptive Computing

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-configuration</td>
<td>The system requires limited or no human intervention in order to set-up.</td>
</tr>
<tr>
<td>Self-optimization</td>
<td>The system is able to achieve user-defined goals autonomously, without human interaction.</td>
</tr>
<tr>
<td>Self-healing</td>
<td>The system can detect and recover from faults without human intervention.</td>
</tr>
</tbody>
</table>

Together with self-protection, these are the properties identified in [3] for autonomic system.
Typical Properties of Self-adaptive Computing

Self-configuration
The system requires limited or no human intervention in order to set-up.

Self-optimization
The system is able to achieve user-defined goals autonomously, without human interaction.

Self-healing
The system can detect and recover from faults without human intervention.

Together with self-protection, these are the properties identified in [3] for autonomic system.
Self-configuration Example

Multi-platform software

Software that is able to run on different hardware configurations seamlessly is a good example of self-configuration.
Self-configuration Example

Multi-platform software

Software that is able to run on different hardware configurations seamlessly is a good example of self-configuration.

![Diagram showing self-configuration process]

- **Inst. Tools** → **Detect Config.** → **Hardware** → **Run** → **Software** → **Install**
Typical Properties of Self-adaptive Computing

Self-configuration
The system requires limited or no human intervention in order to set-up.

Self-optimization
The system is able to achieve user-defined goals autonomously, without human interaction.

Self-healing
The system can detect and recover from faults without human intervention.

Together with self-protection, these are the properties identified in [3] for autonomic system.
Typical Properties of Self-adaptive Computing

Self-configuration
The system requires limited or no human intervention in order to set-up.

Self-optimization
The system is able to achieve user-defined goals autonomously, without human interaction.

Self-healing
The system can detect and recover from faults without human intervention.

Together with self-protection, these are the properties identified in [3] for autonomic system.
Self-optimization Example

Smart Video Players

Players that can adjust media encoding in order to maintain a certain Quality of Service (QoS) can be considered self-optimizing applications.
Self-optimization Example

Smart Video Players

Players that can adjust media encoding in order to maintain a certain Quality of Service (QoS) can be considered self-optimizing applications.
Typical Properties of Self-adaptive Computing

Self-configuration
The system requires limited or no human intervention in order to set-up.

Self-optimization
The system is able to achieve user-defined goals autonomously, without human interaction.

Self-healing
The system can detect and recover from faults without human intervention.

Together with self-protection, these are the properties identified in [3] for autonomic system.
Typical Properties of Self-adaptive Computing

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-configuration</td>
<td>The system requires limited or no human intervention in order to set-up.</td>
</tr>
<tr>
<td>Self-optimization</td>
<td>The system is able to achieve user-defined goals autonomously, without human interaction.</td>
</tr>
<tr>
<td>Self-healing</td>
<td>The system can detect and recover from faults without human intervention.</td>
</tr>
</tbody>
</table>

Together with self-protection, these are the properties identified in [3] for autonomic systems.
Self-healing Example

Reconfigurable Logic

FPGAs are a good playground for self-healing implementation. Part of the hardware resources can be used to verify the correct functioning of the rest of the logic and force reconfiguration when a fault is detected.
Self-healing Example

Reconfigurable Logic

FPGAs are a good playground for self-healing implementation. Part of the hardware resources can be used to verify the correct functioning of the rest of the logic and force reconfiguration when a fault is detected.
### Typical Properties of Self-adaptive Computing

<table>
<thead>
<tr>
<th><strong>Self-configuration</strong></th>
<th>The system requires limited or no human intervention in order to set-up.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Self-optimization</strong></td>
<td>The system is able to achieve user-defined goals autonomously, without human interaction.</td>
</tr>
<tr>
<td><strong>Self-healing</strong></td>
<td>The system can detect and recover from faults without human intervention.</td>
</tr>
</tbody>
</table>

Together with self-protection, these are the properties identified in [3] for autonomic system.
Typical Properties of Self-adaptive Computing

Self-configuration

The system requires limited or no human intervention in order to set-up.

Self-optimization

The system is able to achieve user-defined goals autonomously, without human interaction.

Self-healing

The system can detect and recover from faults without human intervention.

Together with self-protection, these are the properties identified in [3] for autonomic system.
Research Question

Is RL a suitable approach for self-adaptive computing?
Case Study

Testing Environment

- Desktop workstation
- Multi-core Intel i7 Processor
- Linux-based operating system

Objective of our Experiments

Case Study

Testing Environment

- Desktop workstation
- Multi-core Intel i7 Processor
- Linux-based operating system

Objective of our Experiments

Case Study

Testing Environment

- Desktop workstation
- Multi-core Intel i7 Processor
- Linux-based operating system

Objective of our Experiments

Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$

- A reward function $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

$^2$ `sched_setaffinity` system call
$^3$ `cpufrequtils` package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled\(^2\) and (2) CPU frequency\(^3\)

- A reward function $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

---

\(^2\) sched_setaffinity system call
\(^3\) cpufrequtils package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$

- A reward function $R(s): S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

$^2$ sched_setaffinity system call
$^3$ cpufrequtils package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$

- A reward function $R(s) : S \to \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

$^2$ `sched_setaffinity` system call
$^3$ `cpufrequtils` package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]
- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$
- A reward function $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

$^2$ sched_setaffinity system call
$^3$ cpufrequtils package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$

- A reward function $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

---

$^2$ `sched_setaffinity` system call

$^3$ `cpufrequtils` package
Tests Set-Up

Reinforcement Learning Framework

- A finite set of states $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- A finite set of actions $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^{2}$ and (2) CPU frequency $^{3}$

- A reward function $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

$^{2}$ sched_setaffinity system call
$^{3}$ cpufrequtils package
**Tests Set-Up**

### Reinforcement Learning Framework

- **A finite set of states** $S$ → heart rate of the PARSEC benchmark application measured through Heart Rate Monitor (HRM) APIs [5]

- **A finite set of actions** $A$ → (1) number of cores on which the PARSEC benchmark application is scheduled $^2$ and (2) CPU frequency $^3$

- **A reward function** $R(s) : S \rightarrow \mathbb{R}$ → whether a user-defined target (in heartbeats/s) is met or not

---

$^2$ `sched_setaffinity` system call

$^3$ `cpufrequtils` package
Self-configuration

`blackscholes` managed exploiting ADP and core allocation.
Self-configuration

`blackscholes` managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

`blackscholes` managed exploiting ADP and core allocation.
Self-configuration

`blackscholes` managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-configuration

blackscholes managed exploiting ADP and core allocation.
Self-optimization

canneal managed exploiting ADP and core allocation.
Self-optimization

canneal managed exploiting ADP and core allocation.
Self-healing

canneal managed exploiting ADP, core allocation, and frequency scaling.
Self-healing

J. Panerati et al. – On Self-adaptive Resource Allocation through Reinforcement Learning
Self-healing

**canneal** managed exploiting ADP, core allocation, and frequency scaling.
Self-healing

canneal managed exploiting ADP, core allocation, and frequency scaling.
Conclusions

- Reinforcement learning and its relation with other machine learning methodologies and behavioural psychology
- Properties of self-adaptive computing
- How to exploit reinforcement learning for self-adaptive computing
- Experimental results showing reinforcement learning enabling self-adaptive computing properties
Conclusions

- Reinforcement learning and its relation with other machine learning methodologies and behavioural psychology
- Properties of self-adaptive computing
- How to exploit reinforcement learning for self-adaptive computing
- Experimental results showing reinforcement learning enabling self-adaptive computing properties
Conclusions

- Reinforcement learning and its relation with other machine learning methodologies and behavioural psychology
- Properties of self-adaptive computing
- How to exploit reinforcement learning for self-adaptive computing
- Experimental results showing reinforcement learning enabling self-adaptive computing properties
Conclusions

- Reinforcement learning and its relation with other machine learning methodologies and behavioural psychology
- Properties of self-adaptive computing
- How to exploit reinforcement learning for self-adaptive computing
- Experimental results showing reinforcement learning enabling self-adaptive computing properties
Conclusions

- Reinforcement learning and its relation with other machine learning methodologies and behavioural psychology
- Properties of self-adaptive computing
- How to exploit reinforcement learning for self-adaptive computing
- Experimental results showing reinforcement learning enabling self-adaptive computing properties
Q&A

4http://www.dilbert.com/

