Politecnico di Milano

Facoltà di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

. . . Titolo della tesi . . .
. . . al massimo su due righe . . .

Advisor: Elisabetta DI NITTO
Co-Advisor: Marco SCAVUZZO

Master thesis by:
Fabio ARCIDIACONO matr. 799001

Academic Year 2013-2014

dedica. . .

Ringraziamenti

Ringraziamenti vari, massimo una o due pagine.

Milano, 1 Aprile 2005

Fabio.

Estratto

abstract in italiano

Abstract

abstract in english

Table of Contents

Chapter 1

Introduction

Introduzione al lavoro. Inizia direttamente, senza nessuna sezione.

Argomenti trattati suddivisi sezione per sezione...

Per citare un articolo, ad esempio [1] o [1, 2] utilizzare il comando `cite`.

Per gestire i file di tipo `bib` esiste il programma `JabRef` disponibile sul sito `http://jabref.sourceforge.net/`.

Original Contributions

This work include the following original contributions:

- ...riassunto sintetico dei diversi contributi
- ...
- ...

Outline of the Thesis

This thesis is organized as follows:

- In Chapter 1 ...
- In Chapter **??** ...
- In Chapter **??** ...

- . . .

Finally, in Chapter 7, . . .

Chapter 2

State of the art

2.1 Introduction

Introduzione agli argomenti trattati nel capitolo, dalle 4 alle 10 righe.

2.2 ...

Argomenti trattati suddivisi sezione per sezione...

2.3 Figure

Per includere delle figure come la Figura 5.1 usare il comando `includegraphics`.

2.4 Algoritmi

Per includere degli algoritmi come l'Algoritmo 3 usare lo stile `algpseudocode` presente nel package `algorithmicx`.

2.5 Summary

Riassunto del capitolo

Algorithm 1 Un esempio di algoritmo.

```
Initialize $Q(\cdot, \cdot)$ arbitrarily
for all episodes do
    $t \leftarrow 0$
    Initialize $s_t$
    repeat
        $a_t \leftarrow \pi(s_t)$
        perform action $a_t$; observe $r_{t+1}$ and $s_{t+1}$
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t))$
        $t \leftarrow t + 1$
    until $s_t$ is terminal
end for
```

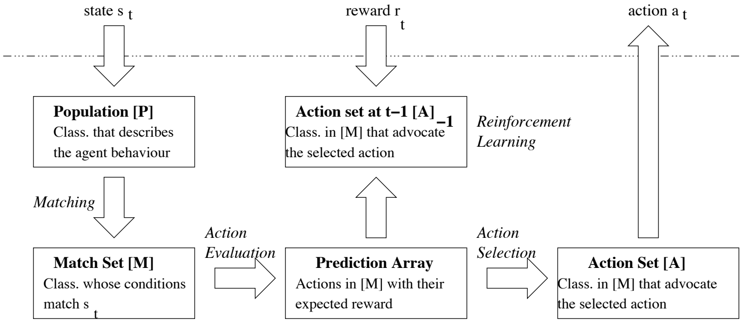
state s t
reward r t
action a t
Population [P]
Class. that describes the agent behaviour
Action set at t−1 [A]−1
Class. in [M] that advocate the selected action
Reinforcement Learning
Matching
Action Evaluation
Action Selection
Match Set [M]
Class. whose conditions match s t
Prediction Array
Actions in [M] with their expected reward
Action Set [A]
Class. in [M] that advocate the selected action

Figure 2.1: . . . titolo

Chapter 3

Problem setting

3.1 Introduction

Introduzione agli argomenti trattati nel capitolo, dalle 4 alle 10 righe.

3.2 ...

Argomenti trattati suddivisi sezione per sezione...

3.3 Figure

Per includere delle figure come la Figura 5.1 usare il comando `includegraphics`.

3.4 Algoritmi

Per includere degli algoritmi come l'Algoritmo 3 usare lo stile `algpseudocode` presente nel package `algorithmicx`.

3.5 Summary

Riassunto del capitolo

Algorithm 2 Un esempio di algoritmo.

```
Initialize $Q(\cdot, \cdot)$ arbitrarily
for all episodes do
    $t \leftarrow 0$
    Initialize $s_t$
    repeat
        $a_t \leftarrow \pi(s_t)$
        perform action $a_t$; observe $r_{t+1}$ and $s_{t+1}$
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t))$
        $t \leftarrow t + 1$
    until $s_t$ is terminal
end for
```

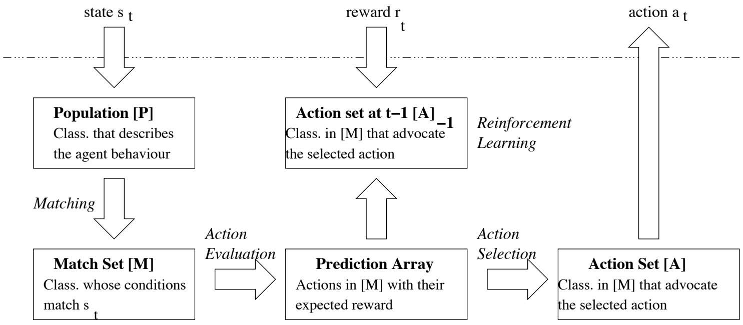

Figure 3.1: . . . titolo

Chapter 4

Kundera extension

4.1 Introduction

In this chapter will be presented the way in which Kundera is supposed to be extended, the problems occurred in the process and how the community helped in achieving the result. In section 4.3 are discussed the detail fo the extension for Google Datastore and in the section 4.4 the details for Azure Table.

4.2 Kundera's Client Extension Framework

Kundera as on open source project, thought that other developers could be interested in using it and extending its support to other datastore. So in the wiki is presented the Client Extension Framework which provides a short description on how Kunders clients should work and provides the interfaces and classes that should be developed in order to make the client work properly.

Looking at Kundera architecture, described in chapter 2, is clear the modularity on which Kundera has been developed. When dealing with classes JPA annotated, the Kundera core provides the necessary logic to fully support the JPA 2.1 specification and when it's time to interact with the underlying database (for persisting, updating or reading entities) it delegate the operation to the configured client in the persistence.xml file.

The steps to build a new Kundera client, basically these are the blocks to be developed:

- the Client, which is the gateway to CRUD operations on database, except for queries;
- the Client Factory, which is used by Kundera to instantiate the Client;
- the Query implementor, which is used by Kundera to run JPA queries by invoking appropriate methods in Entity Readers;
- the Entity Reader, which is used by Kundera to translate the queries into correct client method calls;
- optionally the Schema Manager, to support automatic schema generation.

4.2.1 Approaching the extension

It all seems quite simple but the problem is that the wiki is actually outdated. Two were the main problem in understaing what to do and how, firstly it turns out that the required interfaces are actualy a little different and also are the required methods secondary, and slightly more time consuming, is that no hints are given on the structure and informations carried by the methods arguments. The arguments carry data structures containing informations organized in the kundera metamodel which is the implementation of the JPA metamodel that contains all the information associated (throug annotations) to a class or a field.

Due to those problems and to shrink the developing time, the solution was to write on the Kundera google group page to ask the community for more updated infos about Kundera extension. Briefly an answer has come and I've started a conversation with one of the developers of Kundera who helped me giving the updated infos for the Kundera's Client Extension Framework and tell me to look forward to the other client implementation for some examples. Unfortunatley that was not enough since my question about the metamodel were ignored so the main problem was still unresolved.

At this point the most valid solution was to approach the extesion as a test driven development, so looking at the tests code of the other clients I've writed a set of unit tests one foreach feature (tests are analyzed in detail in chapter 6. With the tests failing and the code of Kundera core was then possible to reverse engineer the arguments thath were not documented and thus be able to develop the new extensions.

4.3 GAE Datastore client

4.3.1 Datastore structure

Here talk also about strong vs eventual consistency

4.3.2 Implementing JPA relationships

4.3.3 Other supported JPA features

4.3.4 Query support

4.4 Azure Table client

4.4.1 Azure Table structure

Here talk also about strong vs eventual consistency

4.4.2 Implementing JPA relationships

4.4.3 Other supported JPA features

4.4.4 Query support

4.5 Summary

In this chpater has been introduced in details how Kundera extension should been developed, the problem encountered during the development, how tey've been addressed and the detail of the implementation of the two extensions including what the feature currently supported. In the next chapter will be explained how has been possible to integrate Kundera into CPIM as part of the NoSQL service.

Chapter 5

CPIM extension

5.1 Introduction

Introduzione agli argomenti trattati nel capitolo, dalle 4 alle 10 righe.

5.2 ...

Argomenti trattati suddivisi sezione per sezione...

5.3 Figure

Per includere delle figure come la Figura 5.1 usare il comando `includegraphics`.

5.4 Algoritmi

Per includere degli algoritmi come l'Algoritmo 3 usare lo stile `algpseudocode` presente nel package `algorithmicx`.

5.5 Summary

Riassunto del capitolo

Algorithm 3 Un esempio di algoritmo.

```
Initialize $Q(\cdot, \cdot)$ arbitrarily
for all episodes do
    $t \leftarrow 0$
    Initialize $s_t$
    repeat
        $a_t \leftarrow \pi(s_t)$
        perform action $a_t$; observe $r_{t+1}$ and $s_{t+1}$
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t))$
        $t \leftarrow t + 1$
    until $s_t$ is terminal
end for
```

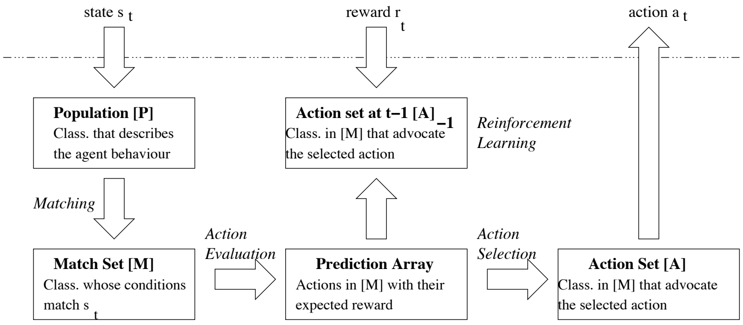
state s t
reward r t
action a t
Population [P]
Class. that describes the agent behaviour
Action set at t−1 [A]−1
Class. in [M] that advocate the selected action
Reinforcement Learning
Matching
Action Evaluation
Action Selection
Match Set [M]
Class. whose conditions match s t
Prediction Array
Actions in [M] with their expected reward
Action Set [A]
Class. in [M] that advocate the selected action

Figure 5.1: . . . titolo

Chapter 6

Evaluation

6.1 Introduction

Introduzione agli argomenti trattati nel capitolo, dalle 4 alle 10 righe.

6.2 Test correctness of CRUD operations

JUnit tests

6.3 Performance tests

Task about YCSB and Kundera-benchmarks

6.4 Summary

Riassunto del capitolo

Chapter 7

Conclusions and future Works

Conclusioni del lavoro e sviluppi futuri. Massimo una o due pagine.

Appendices

Appendix A

Configuring Kundera extensions

A.1 Introduction

Introduzione agli argomenti trattati nell'appendice, dalle 4 alle 10 righe.

A.2 ...

Argomenti trattati suddivisi sezione per sezione. Alla fine del capitolo non includere alcun sommario.

Appendix B

Run YCSB tests

B.1 Introduction

Introduzione agli argomenti trattati nell'appendice, dalle 4 alle 10 righe.

B.2 ...

Argomenti trattati suddivisi sezione per sezione. Alla fine del capitolo non includere alcun sommario.

Bibliography

[1] D. H. Ackley. *A connectionist machine for genetic hillclimbing.* Kluwer, Boston, 1987.

[2] Lee Altenberg. The evolution of evolvability in genetic programming. In Kinnear Jr. Kenneth E., editor, *Advances in Genetic Programming*, chapter 3, pages 47–74. MIT Press, 1994.