Cloud-TM: Harnessing the Cloud with Distributed Transactional Memories

Luís Rodrigues
INESC-ID/IST

LADIS 2009
(joint work with Paolo Romano, Nuno Carvalho, and João Cachopo)

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

Motivation

- Development of programming models and tools that simplify the design and implementation of applications for the cloud.

D-STMs

Advocate the use of Distributed Software Transactional Memory (D-STM) in the context of cloud computing.

- Limitations of other models.
- Limitations of D-STMs.
- Current work and FénixEDU.

Motivation

- Development of programming models and tools that simplify the design and implementation of applications for the cloud.

D-STMs

Advocate the use of Distributed Software Transactional Memory (D-STM) in the context of cloud computing.

- Limitations of other models.
- Limitations of D-STMs.
- Current work and FénixEDU.

Motivation

- Development of programming models and tools that simplify the design and implementation of applications for the cloud.

D-STMs

Advocate the use of Distributed Software Transactional Memory (D-STM) in the context of cloud computing.

- Limitations of other models.
- Limitations of D-STMs.
- Current work and FénixEDU.

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

MapReduce

- Program needs to be structured as a combination of *map* and *reduce* operations[DG08].
- Run-time automates:
  - Data partitioning.
  - Scheduling.
  - Failure recovery.
- However map-reduce programming model is unnatural for many applications.
  - Requires the use of a different programming paradigm.
  - Several extensions[ORS$^+$08].
  - Strong debate about the merits and drawbacks of the approach[Aba09, DSml].

PGAS

Partitioned Global Address Space

- Combines the DSM model with flavors from the message-passing (MPI-like) model[BCA+06].
  - Provides a global address space.
  - The programmer has explicitly control over data locality.
- Complex programming interface.
  - Targeted towards high-performance computing applications

D-STM

Distributed Software Transactional Memories

- Extends the TM abstraction across the boundaries of a single machine.
- Only recently the first systems have been implemented and reported[KAJ+08, BAC08, MMA06, CRCR09, AMS+07].
- Can D-STMs avoid the pitfalls of DSM systems?
    - Less synchronization points (only at commit time).
- Two-classes of systems:
    - Fully-replicated for small scale systems.
    - Partitioned address space for large-scale clusters.

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

Research Challenges

Automatic parallelization

- Extremely hard ... but
- Transactional support makes easier to implement strategies based on the speculative execution of portions of code[HWO98, SCZM00, LTC+06].

Research Challenges

Fault-tolerance

- Only started to be considered by recent D-STMs.

Research Challenges

Coping with Workload Heterogeneity

- STM performance (even in the centralized case) is heavily dependent on the workload.
- Different algorithms exist, optimized for different workloads (the same applies to the underlying communication protocols).
- Autonomic adaptation.

Research Challenges

Automatic Resource Provisioning

- Algorithms to automatically perform the provisioning of the computing resources are required.
- There is now a reasonable amount of work from the autonomic computing that can be used in the D-STM context.

Research Challenges
ACI vs ACID

- Many STM works do not consider durability of data.
- In a general purpose environment, persistence needs to be addressed.
- Furthermore it may need integrated with the mechanisms required to transfer data among nodes.

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

Some Preliminary Results

D^{2}STM

- Dependable Distributed STM[CRCR09] is a distributed fully replicated STM, that uses atomic broadcast to coordinate replicas. Bloom filters are used to control the size of messages.

Some Preliminary Results

Speculative Replication

- A technique that runs (potentially conflicting) transactions speculatively in different orders, to hide the inter-replica coordination latency.

Some Preliminary Results

Identifying and predicting the data access

- We are developing stochastic techniques for identifying and predicting the data access patterns of transactional applications[GCP09].
- Used for automatic partitioning and algorithm optimization.

Some Preliminary Results

Thread-level speculation techniques

- Aimed at achieving automatic, performance-effective parallelization of sequential programs [AC09].
- Current prototype is capable of automatically parallelizing Java code meant to be executed on a single multi-core machine (with very promising preliminary results).

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

The FénixEDU system

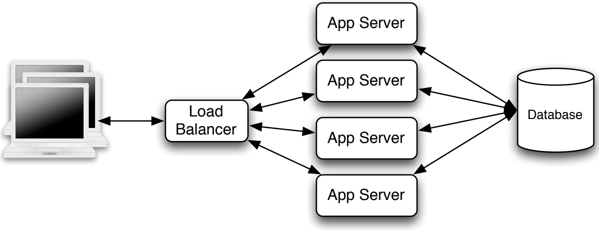
- Manages the on-line Campus activities
- Used in production by the Technical university of Lisbon
    - ... and being installed in other universities
- Processing between 1 000 000 and 4 500 000 transactions per day.

The FénixEDU system

- Web application
- Object-oriented domain model
- Relational DBMS to store data
- Object/relational mapping tool to store objects in the database
- Runs on a STM implementation

The FénixEDU system

- Application server is distributed to improve system throughput
- Database is used to synchronize replicas



The FénixEDU system requirements

- It could clearly be run in the cloud.
- Programmers use the object-oriented model they are familiar with.
- Resource requirements are variable (with short periods of hig peak loads).
- It has consistency constraints.

Outline

1. Introduction
2. Programming models for the Cloud
3. Towards a Cloud-TM
4. Some Preliminary Results
5. FénixEDU
6. Conclusions
7. Bibliography

Conclusions

- D-STM have many good properties that make them a promising technology to support distributed applications (with consistency requirements) in the Cloud.
- To fulfill this goal, many challenges need to be faced.
- There is already evidence that these challenges can be addressed.

Daniel J. Abadi.
Data management in the cloud: Limitations and opportunities.
IEEE Data Eng. Bulletin, 32(1), March 2009.

Ivo Anjo and João Cachopo.
Jaspex: Speculative parallel execution of java applications.
In *INForum 2009: Proceedings of the 1st Simpósio de Informática*, Lisbon, Portugal, September 2009.

Marcos K. Aguilera, Arif Merchant, Mehul Shah, Alistair Veitch, and Christos Karamanolis.
Sinfonia: a new paradigm for building scalable distributed systems.
In *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 159–174, New York, NY, USA, 2007. ACM.

Robert L. Bocchino, Vikram S. Adve, and Bradford L. Chamberlain.
Software transactional memory for large scale clusters.
In *Proceedings of the Symposium on Principles and practice of parallel programming*, pages 247–258, New York, NY, USA, 2008. ACM.

Christopher Barton, CĆlin Casçaval, George Almási, Yili Zheng, Montse Farreras, Siddhartha Chatterje, and José Nelson Amaral.
Shared memory programming for large scale machines.
In *Proceedings of the PLDI '06*, pages 108–117, New York, NY, USA, 2006. ACM.

Maria Couceiro, Paolo Romano, Nuno Carvalho, and Luis Rodrigues.
D2stm: Dependable distributed software transactional memory.
Technical Report 30/2009, INESC-ID, May 2009.

Jeffrey Dean and Sanjay Ghemawat.
Mapreduce: simplified data processing on large clusters.
Comm. ACM, 51(1):107–113, 2008.

D. DeWitt and M. Stonebraker.
Mapreduce: A major step backwards,
`http://www.databasecolumn.com/2008/01/mapreduce-a-major-step-back.html`.

Stoyan Garbatov, João Cachopo, and João Pereira.

Data access pattern analysis based on bayesian updating.
In *INForum 2009: Proceedings of the 1st Simpósio de Informática*, Lisbon, Portugal, September 2009.

Lance Hammond, Mark Willey, and Kunle Olukotun.
Data speculation support for a chip multiprocessor.
SIGOPS Operating Systems Review, 32(5):58–69, 1998.

C. Kotselidis, M. Ansari, K. Jarvis, M. Lujan, C. Kirkham, and I. Watson.
Distm: A software transactional memory framework for clusters.
In *Parallel Processing, 2008. ICPP '08. 37th International Conference on*, pages 51–58, Sept. 2008.

Wei Liu, James Tuck, Luis Ceze, Wonsun Ahn, Karin Strauss, Jose Renau, and Josep Torrellas.
POSH: a TLS compiler that exploits program structure.
In *PPoPP '06: Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 158–167, New York, NY, USA, 2006. ACM.

Kaloian Manassiev, Madalin Mihailescu, and Cristiana Amza.
Exploiting distributed version concurrency in a transactional memory cluster.
In *Proceedings of the Symposium on Principles and practice of parallel programming*, pages 198–208, New York, NY, USA, 2006. ACM.

Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins.
Pig latin: a not-so-foreign language for data processing.
In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD int. conf. on Management of Data*, pages 1099–1110, New York, NY, USA, 2008. ACM.

J. Greggory Steffan, Christopher B. Colohan, Antonia Zhai, and Todd C. Mowry.
A scalable approach to thread-level speculation.
In *ISCA '00: Proceedings of the 27th annual international symposium on Computer architecture*, pages 1–12, New York, NY, USA, 2000. ACM.