FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Energy-efficient Persistence for Extensible Virtual Shared Memory on the Android Operating System

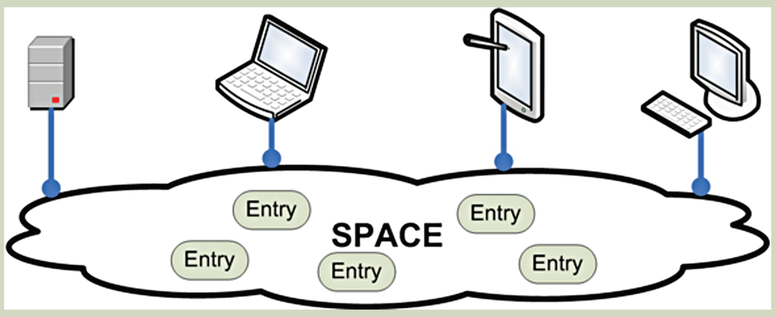
Masterstudium:
Software Engineering & Internet Computing

Jan Zarnikov

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzerbau
BetreuerIn: A.o. Univ. Prof. Dr. Dipl.-Ing. eva Kühn

INTRODUCTION

Space based computing solves problem of coordination of different processed of a distributed application by sharing data in a virtual medium called space.
eXtensible Virtual Shared Memory (XVSM) and its reference implementation MozartSpaces take the space based computing concept and extend it by providing a powerful coordination facilities for managing the contents of the space, transactions and remote communication. Clients can extend the functionality even further with aspects and custom coordination logic. MozartSpaces is available for Java 2 Standard Edition (J2SE) and the Android operating system.



High-level view of the Space-Based Computing Paradigm [Mor10]

PROBLEM STATEMENT

Until now MozartSpaces held all its data in memory only. Previous attempts to add persistent data storage by either implementing it as orthogonal functionality with aspects or by replacing the core of MozartSpaces with a database showed some drawbacks. A new approach has to be found. Following requirements were defined for the new persistence implementation

- The persistence must be **flexible and extensible**. It should be possible to use **different database engines** as backends.
- The persistence must be transparent to the clients.
- The persistence must be compatible with both the **J2SE** version and the **Android** version of MozartSpaces.
- The persistence should be **fast and scalable**.
- The persistence must be compatible with the **ACID transactions** in MozartSpaces
- When run on a mobile device the persistence should not consume too much **energy** in order to save battery.

PERSISTENCE IMPLEMENTATION

It was decided to solve the problem by creating a new persistence layer which is tightly integrated into the core of MozartSpaces. The persistence layer hides the database-specific details and can use different database engines:

- The Android version of MozartSpaces can use the lightweight **SQLite** database which is distributed as part of the Android operating system.
- **Berkeley DB** Java Edition can also be used to store data in both the J2SE and the Android version of MozartSpaces.

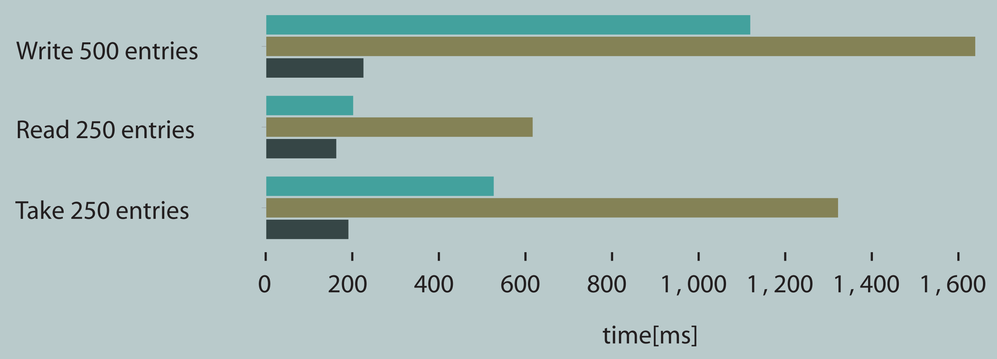
It is possible to add support for other databases without having to change the code of MozartSpaces itself. An in-memory mode is also available.

ACID transactions were implemented by:

- either **mapping** MozartSpaces transactions to transactions of the database (Berkeley DB)
- or by **buffering write operations** and then flushing them on transaction commit (SQLite)

PERFORMANCE

An extensive **benchmark suite** was developed to evaluate the performance of the persistence on different platforms (J2SE and Android) and in different settings. The chart below shows the performance of MozartSpaces with **SQLite**, **Berkeley DB** and the **in-memory mode** on the Android smartphone HTC Legend.

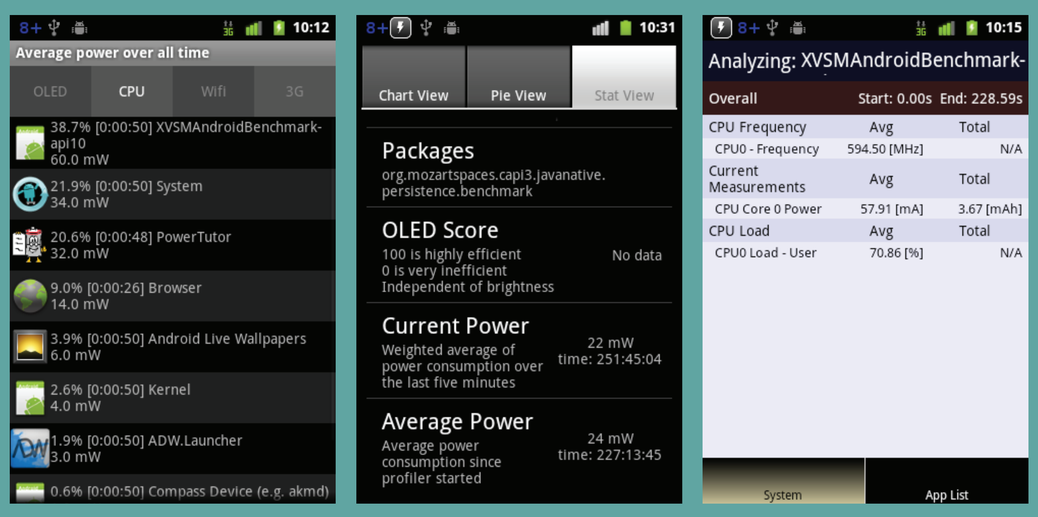


Using a database for data storage significantly **reduced the memory footprint** at runtime compared to previous versions of MozartSpaces. The persistence layer employs several optional and configurable **caches** to speed up the performance.

ENERGY EFFICENCY

Energy efficiency on mobile devices running Android OS was an important criteria. PowerTutor and Trepn Profiler were used to evaluate the power consumption at runtime.

Unfortunately on the tested device both profilers could only measure the consumption of the CPU and did not provide granularity fine enough for code optimizations.



Measuring energy-efficiency with PowerTutor (left and center) and Trepn Profiler (right)

CONCLUSION

The new extended version of MozartSpaces remains compliant with the XVSM specification and offers good performance: with persistence write operations take 5-20x longer and read operations 1.5-10x longer depending on the platform and configuration. Power consumption on Android OS caused by the persistence is small compared to other hardware components.

REFERENCES

- **Richard Mordinyi.**
Managing Complex and Dynamic Software Systems with Space-Based Computing.
PhD thesis, Vienna University of Technology, 2010.

Kontakt: jan.zarnikov@gmail.com