# MnemoSyne - Garbage Collection in Transactional Persistent Distributed Stores

## Goal

Improve programming productivity and reliability by providing fault-tolerant automatic memory management

## Partners

Inesc

## Duration

December 1999 - November 2003

## More information

http://www.gsd/projects/mnemosyne.htm
(PRAXIS/P/EEI/12111/1998)



# Motivation

- workers and organizations carrying out cooperative tasks interacting via shared data
  - e.g.: a virtual enterprise working with interactive CAD for engineering or financial purposes
- shared data accessed either concurrently or sequentially:
  - available at different locations
  - persists beyond completion of a particular application
  - remains consistent even in presence of faults
- these needs are being addressed by several European projects
  - e.g.: PerDiS long-term-research

*Distributed Systems Group*

## Environment: TPDS

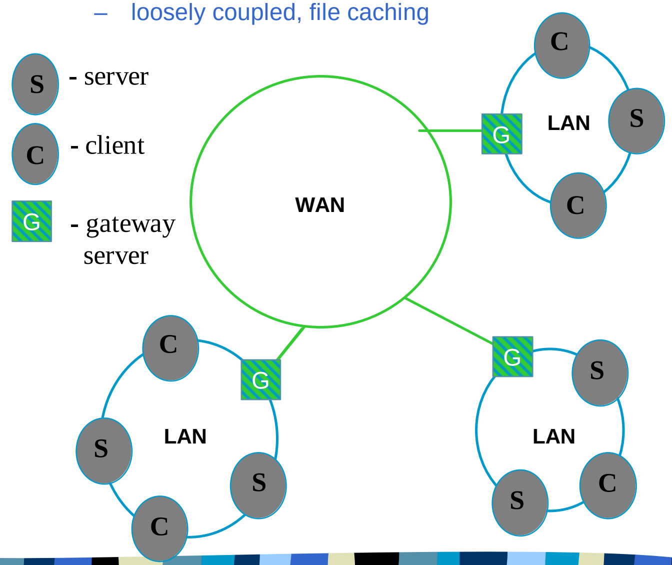
- wide-area transactional distributed persistent stores (TPDS):
  - shared address space spanning every participating site
  - API similar to a centralized system through the use of a distributed shared memory approach
  - automatic persistence (by reachability)
  - optimistic transactional support with versions, conflict detection, and reconciliation
  - support for disconnected work and check-in/check-out operations
  - cooperative caching
  - distributed checkpointing
  - **automatic memory management (i.e. garbage collection)**



*Distributed Systems Group*

## TPDS in a Wide Area Network

- LAN
  - strongly coupled, DSM based
- WAN
  - loosely coupled, file caching



*Distributed Systems Group*

## Problem: Maintain the Referential Integrity

- programmer must keep track of references:
  - stored on disk
  - sent to other sites
  - replicated in several objects
- manual memory management results:
  - dangling references
  - memory leaks
- in short lived applications a painful debug solves these problems, but
  - in a TPDS these errors are much more dramatic:
    - a dangling reference stored on disk will result in a fatal error much later
    - the application suffering the consequences is not responsible for the dangling reference
    - dangling references to remote objects
    - memory full of garbage preventing applications from working (due to leaks)



*Distributed Systems Group*

## Research Issues

- consistency interference:
  - the GC must not compete with transactions for consistent data (replicated on several sites)
- integration with fault-tolerant mechanisms:
  - the GC must be safe, live and complete in presence of multi-version transactions and checkpointing
- fault-tolerance:
  - the GC must be safe, live and complete in presence of site crashes and communication failures
- scalability:
  - the GC must be capable of handling a large amount of data without the need for lots of data structures
- performance:
  - GC overhead must be acceptable

*Distributed Systems Group*

## GC Current Solutions and Approach

- GC
  - aimed at simple systems based on function-shipping
  - no data replication
  - simple transactional support (either centralized or with no versioning)
  - no cooperative caching or distributed checkpointing
  - fault-tolerance is not fully handled:
    - site crashes
    - communication failures
  - not scalable given the large amount of data structures
- Approach
  - develop a general model of a TPDS and characterize the memory usage of applications
  - design, implement and measure a fault-tolerant GC algorithm:
  - continue previous work done in: Broadcast, PerDiS, COOP



*Distributed Systems Group*

## References


- Implementing Garbage Collection in the PerDiS System. Xavier Blondel, Paulo Ferreira and Marc Shapiro. Eight International Workshop on Persistent Object Systems: Design, Implementation and Use, Tiburon, California, USA, August 1998.
- Paulo Ferreira and Marc Shapiro. Modelling a Distributed Cached Store for Garbage Collection: the algorithm and its correctness proof. European Conference on Object-Oriented Programming (ECOOP'98), Brussels, Belgium, July 1998.
- Paulo Ferreira and Marc Shapiro. Larchant: Persistence by Reachability in Distributed Shared Memory through Garbage Collection. IEEE 16th International Conference in Distributed Computing Systems (ICDCS´96), Hong Kong, May 1996.
- Laurent Amsaleg, Paulo Ferreira, Michael Franklin and Marc Shapiro. Evaluating Garbage Collectors for Large Persistence Stores. Workshop on Object Database Behavior, Benchmarks, and Performance, at Object-Oriented Programming Systems and Languages (OOPSLA'95), Austin (TX-USA), October 1995.

# References (cont.)


- A Checkpoint Protocol for an Entry Consistent Shared Memory System. Nuno Neves, Miguel Castro and Paulo Guedes. Thirteenth ACM Symposium on Principles of Distributed Computing (PODC) Los Angeles, CA., August, 1994.
- Paulo Ferreira and Marc Shapiro. Garbage Collection and DSM Consistency. First ACM Conference on Operating Systems Design and Implementation (OSDI'94), Monterey, CA USA, Nov.1994.
- Orthogonal Persistence in a Heterogeneous Distributed Object-Oriented Environment. Pedro Sousa, André Zúquete, Nuno Neves, and José Alves Marques. The Computer Journal 37(6), 1994.
- Pedro Sousa, Manuel Sequeira, André Zúquete, Paulo Ferreira, Cristina Lopes, José Pereira, Paulo Guedes, and José Alves Marques. Distribution and Persistence in the IK Platform: Overview and Evaluation. Computing Systems, 6(4), Fall 1993.