

Warsaw University of Technology

Faculty of Electronics and Information Technology

Dimitrios Georgousis

Gabriel Paic

Distributed Computing and Systems

Project Concept

Introduction

Topic: 13 write – update

Description: Implement distributed memory with read and write access, using write-update protocol.

We present a concept design for the above distributed system, which reflects our current vision of implementation. The aim of the project is educational.

Assumptions

1. Such a system will not need to have a variety of user types and, even if it did, we aim to focus on the ‘distributed’ properties of the system, thus we will be using only one type of user who can perform all operations on it.
2. The system will provide a memory interface for reading and writing. Also, it should provide an interface for memory requests (so that we can use a specific address supplied to us by the system, may also incorporate a memory freeing mechanism).
3. This memory will be the main memory of the different physical machines that make up this system. More specifically, we will have at least two nodes that just serve memory requests, a node that handles system monitoring requests (health check of nodes, fault tolerance etc.) and a node that essentially is the user process. It is only required that the memory request server nodes exist on different physical machines.
4. As mentioned previously, a system – monitor node.
5. All user processes have read and write access to any memory address.
6. Users will use a CLI interface to make requests. These requests are read, write, health check and a request that returns a usable (virtual) memory address.

Whether our system will expand to allow for memory allocations and deallocations or usage of hard drive memory (persistent storage) on top of main memory will depend on how robust our implementation will be and shall be determined during the development cycle.

Use Cases

User types: Such a system will not need to have a variety of user types and, even if it did, we aim to focus on the ‘distributed’ properties of the system, thus we will be using only one type of user who can perform all operations on it.

Operations performed by the user:

* Memory request: user requests some memory to use and receives an appropriate address where they can perform other operations.
* Memory reading: read a specific memory address.
* Memory writing: write to a specific memory address.
* System monitoring: user can see if the system is up and/or which nodes are currently operational etc. This is a health – check functionality.

System Architecture