

# An introduction to JavaSpaces

## Table of contents

- 1 Introduction..... 2
- 2 Overview..... 2
- 3 References..... 2

## 1. Introduction

As a model becomes more complex the elapsed time for a model calculation becomes much more time consuming and takes too long for a single personal computer (PC) to complete. To expand the computing power and shorten the elapsed time, distributed computation by a cluster of PCs can be used. This system is reasonable for ordinary laboratories, because they can make full use of their own PCs and increment computation power in a stepwise manner.

The calculation of models in simBio is suitable for distributed computation because each calculation is separated into an independent job and little communication is needed. Therefore linear effectiveness by adding PCs to the cluster result in a decrease of waiting time to obtain the result.

The distributed computation package named Jini is provided as open source. JavaSpaces, which is a service of Jini, provides a virtual shared memory environment on a computer network.

## 2. Overview

Therefore a calculation server using simBio (Worker) was developed, which reads a model XML of simulation request from the shared memory, and computes the model using simBio, then writes a calculated model XML to the shared memory. Workers are utilized from a client, which sets various parameters for a model, sends request XMLs, and collects the results. Workers on a network are started and stopped by a commander class, see [RemoteClient](#).

The following messages are transferred in the distributed computing process.

1. Client writes a model XML into JavaSpaces.
2. Worker reads a model XML from JavaSpaces.
3. Worker checks the version of Java classes, which was specified to be used in a model XML, by contacting an http class server.
4. Worker retrieves the latest version of Java classes from the http class server, if necessary.
5. Worker starts calculation of a model XML using simBio.
6. Worker writes a notification of starting the calculation into JavaSpaces.
7. Client reads a notification, and then writes a next request of a model XML at the end of the queue.
8. Worker writes a result XML with evaluators into JavaSpaces when the simBio calculation is over.
9. Client reads and gathers the result, and then writes a receipt.
10. Worker reads a receipt and discards a local copy of a result XML.

The figure below show a schematic diagram of distributed computing.

Client, and each server of Javaspaces, simBio, and http can be run on separate computers or on the same PC. The client and the http server are typically executed on the same PC.

## 3. References

1. Nobuaki Sarai [Acceleration of Sensitivity Analysis by Distributed Computation](#)

The published paper can also be read here [Acceleration of Sensitivity Analysis by Distributed Computation](#)