

Massively Multiplayer Online Game Servers

Analysing the needs of an Erlang/OTP Virtual World Distributed Server

Erlang User Conference, Stockholm, Sweden
November 10, 2005



Erlang Training and Consulting Ltd
www.erlang-consulting.com

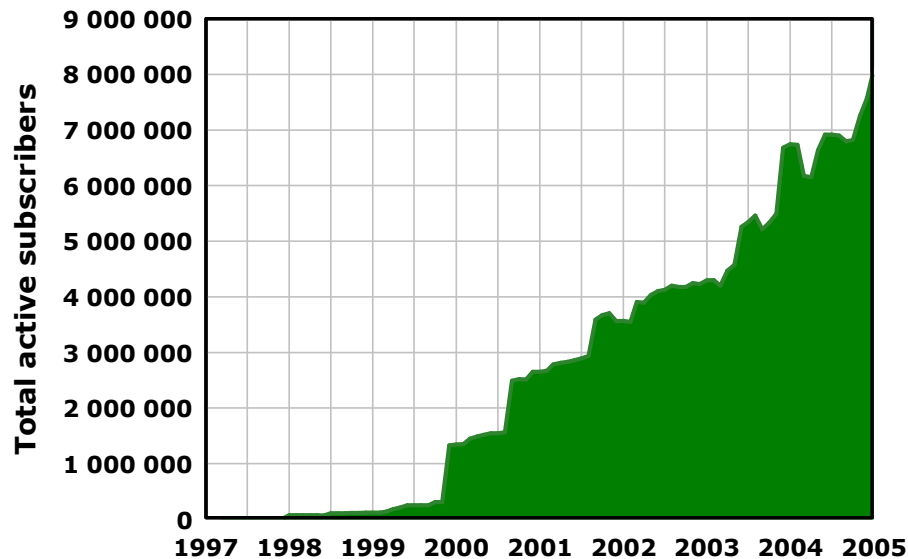
Michał Ślaski
michal@erlang-consulting.com

Contents

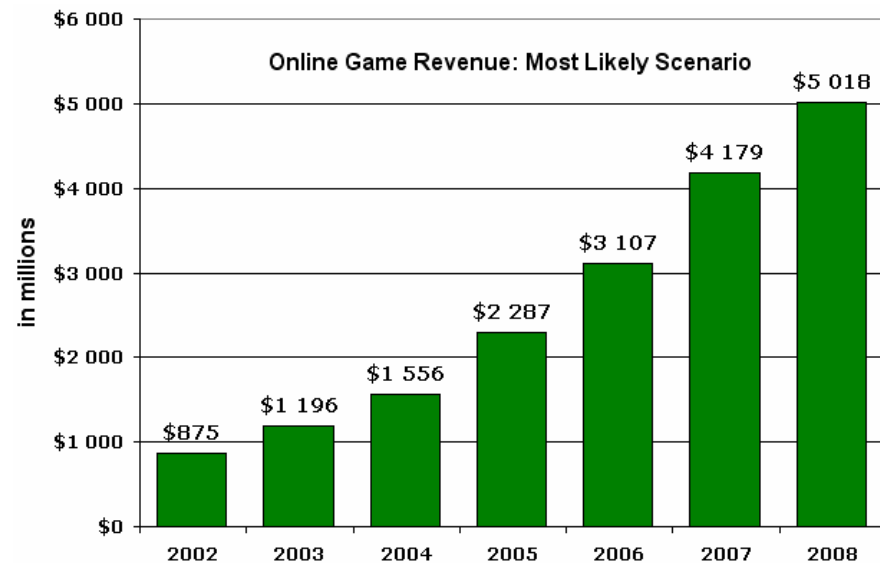
- **Massively Multiplayer Online Game Market**
- **Challenges**
- **Prototyping**
 - *Using Erlang/OTP*
 - *Client and Server Applications*
- **Benchmarking and Stress Testing**
- **Software and Hardware Architecture**
 - *Network and Server Nodes*
- **Synchronisation of the Game State**
- **Conclusions**

Massively Multiplayer Online Game Market

- Significant increase of active subscribers
- Very fast growing market



Source: MMOGCHART.COM



Source: IGDA, "2004 Persistent Worlds White Paper"

Challenges

- **Large quantities of data to process in real time**
- **Frequent synchronisation between clients**
 - *Acceptable delays of 150 - 500 ms*
- **Adaptation to the Client - Server connectivity**
 - *Server response times*
 - *Network latency*
- **Stress testing in the early development stages**
 - *Limited opportunities for load testing with real players*

Prototyping

Make mistakes on a small scale, not in a production project.

Mike Williams

It's not good enough to have ideas - you must also be able to implement them to know that they work

Mike Williams

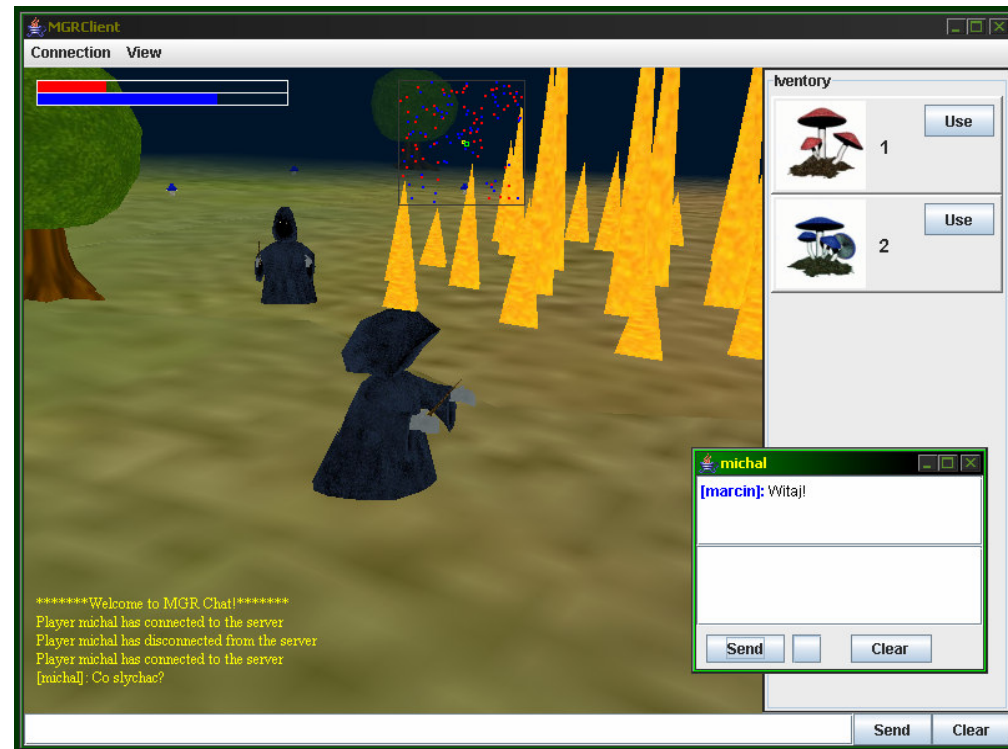
- **Experimenting with functional and technical aspects**
- **Determining the scalability of the architecture**
- **Testing new solutions**

Prototyping : Using Erlang/OTP

- **Open Source, excellent libraries**
 - *Mnesia*
 - *IDX-Tsunami*
- **OS and hardware independent**
- **Concurrency oriented programming**
 - *one process for every connected player*
- **Efficient internal communication**
- **Support for distributed applications**

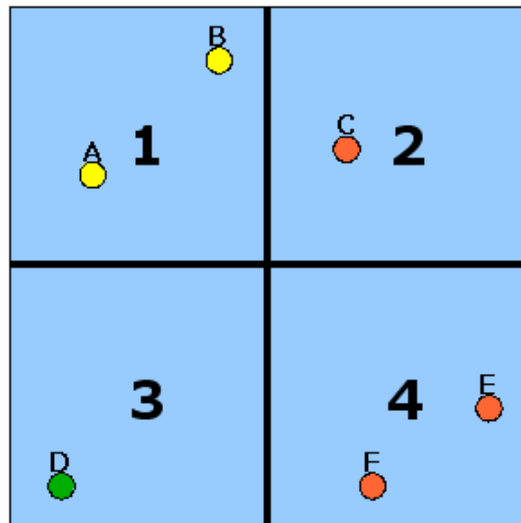
Prototyping : Client Application

- Java + Java3D
- Actions
 - *moving*
 - *collecting objects*
 - *magic*
 - *chatting*
- Player states kept on the server

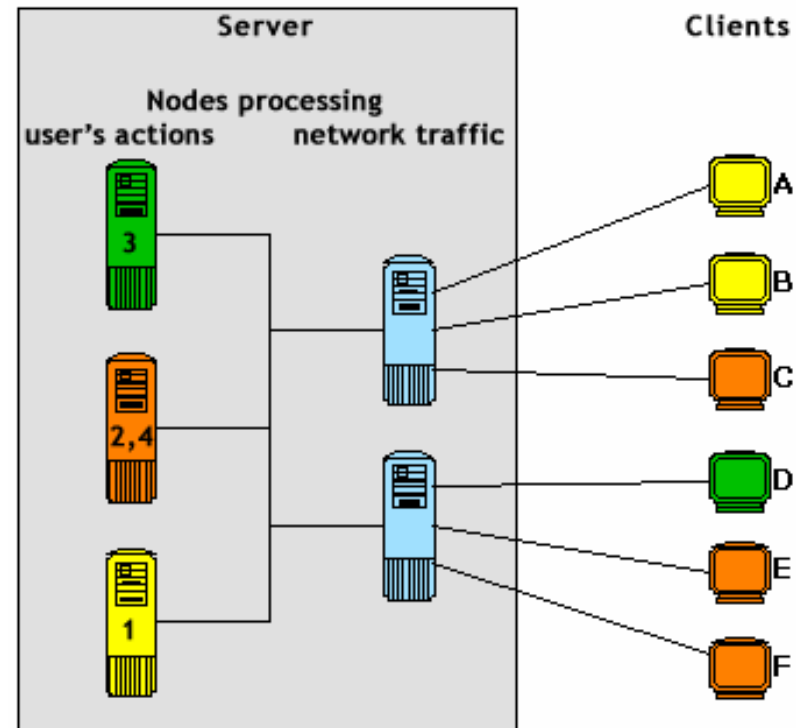


Prototyping : Server Application

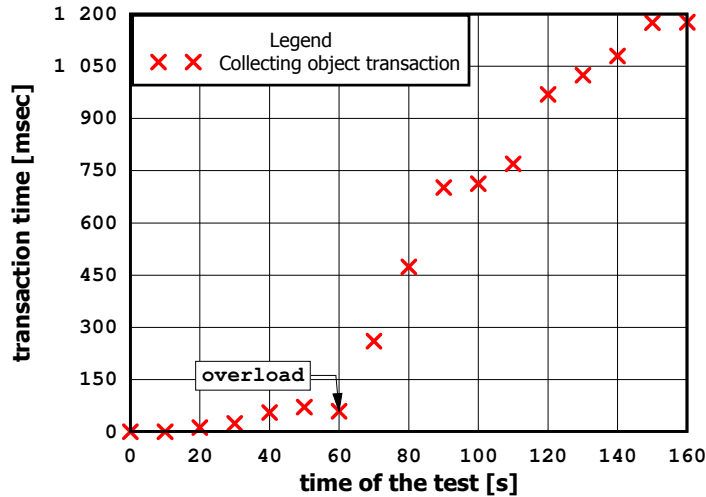
- The game terrain is divided into zones
- Players placed within the same zone are processed on the same node



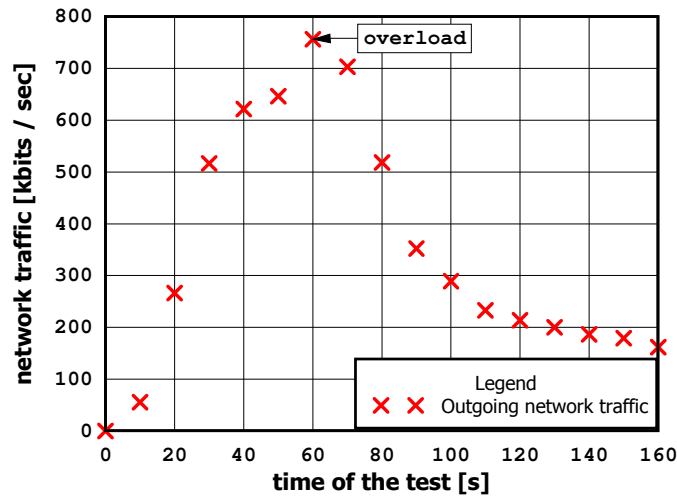
- Server distributed on a cluster of machines



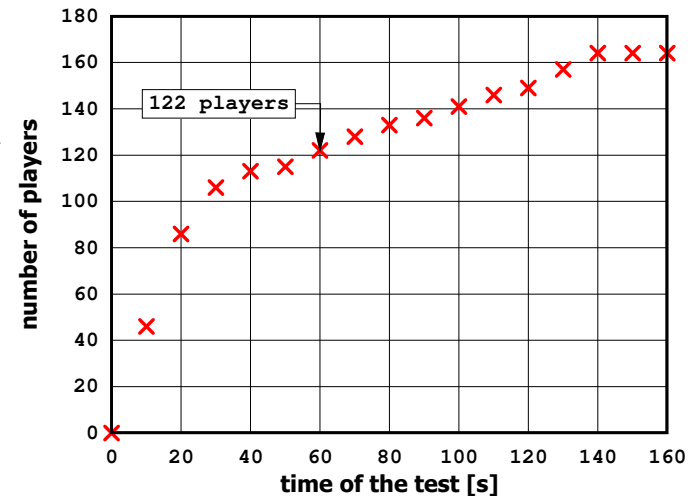
Benchmarking and Stress Testing



- Increase the number of players to overload the server
- Each player runs a transaction every two seconds
- Tests on a 600MHz Celeron processor with 128 MB of RAM

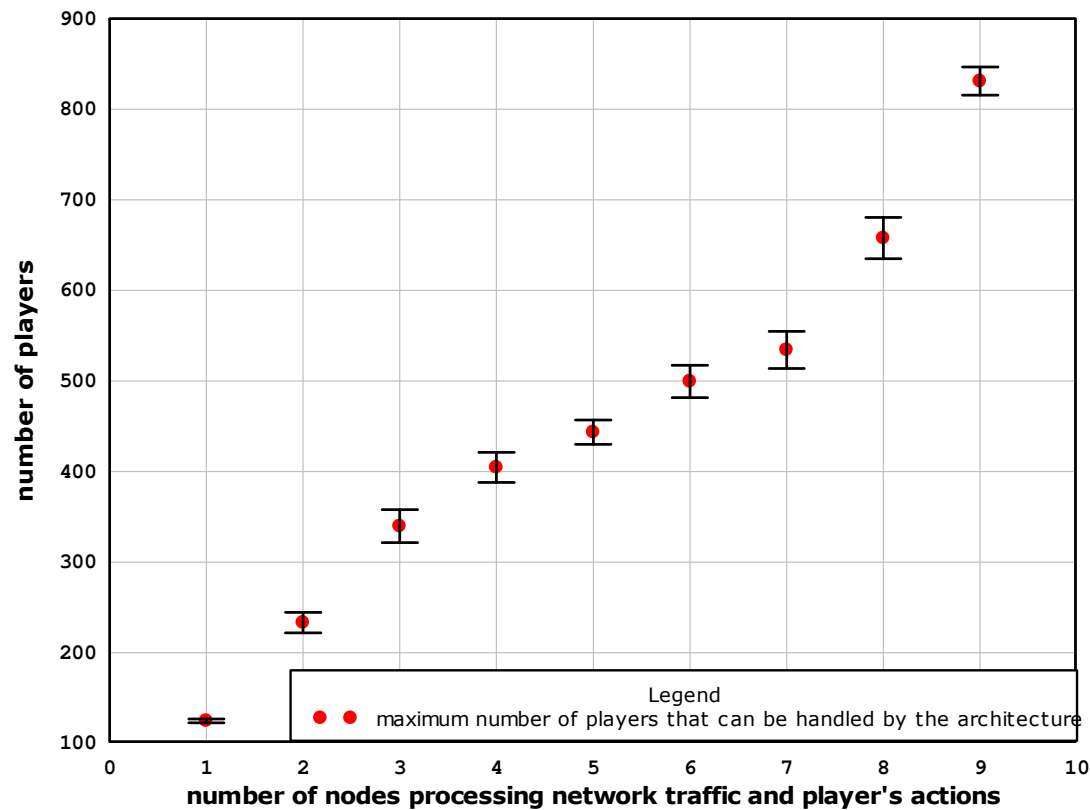


The server was overloaded in the 60th second of the test



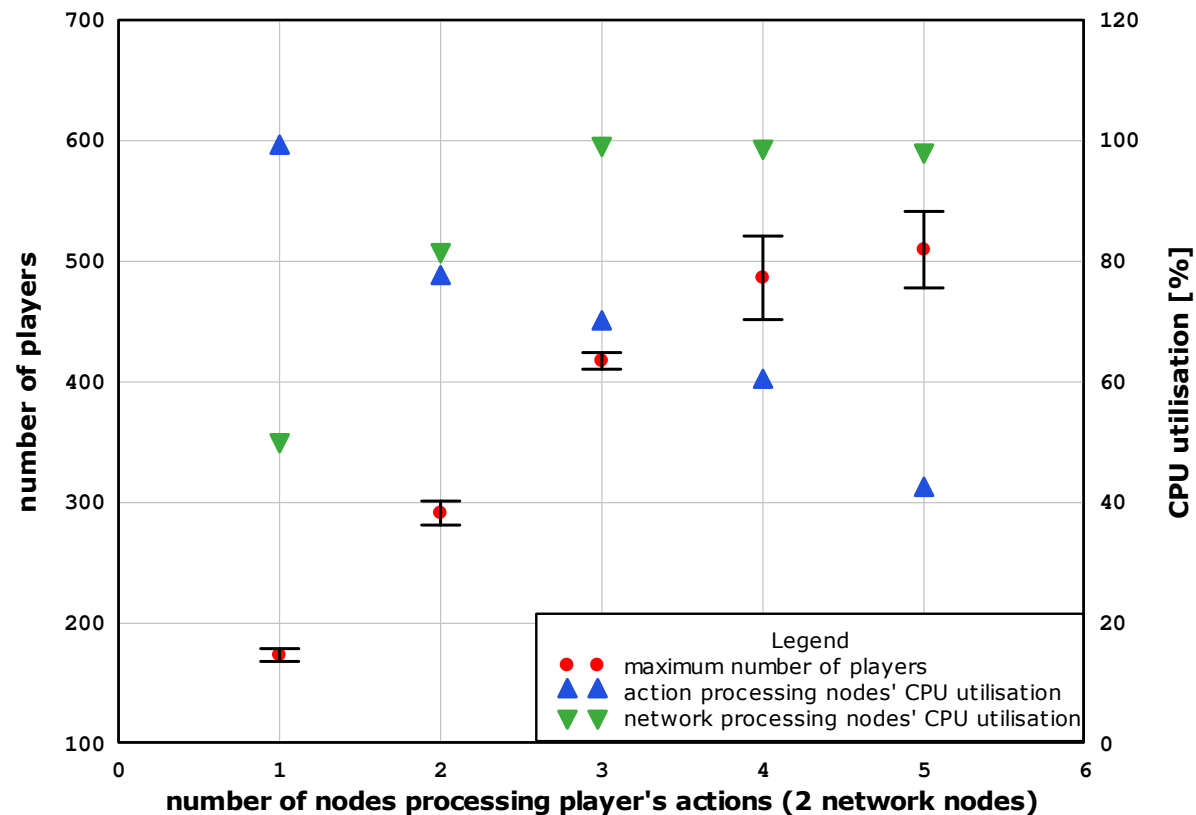
Software and Hardware Architecture

- Every machine is running network traffic and user actions



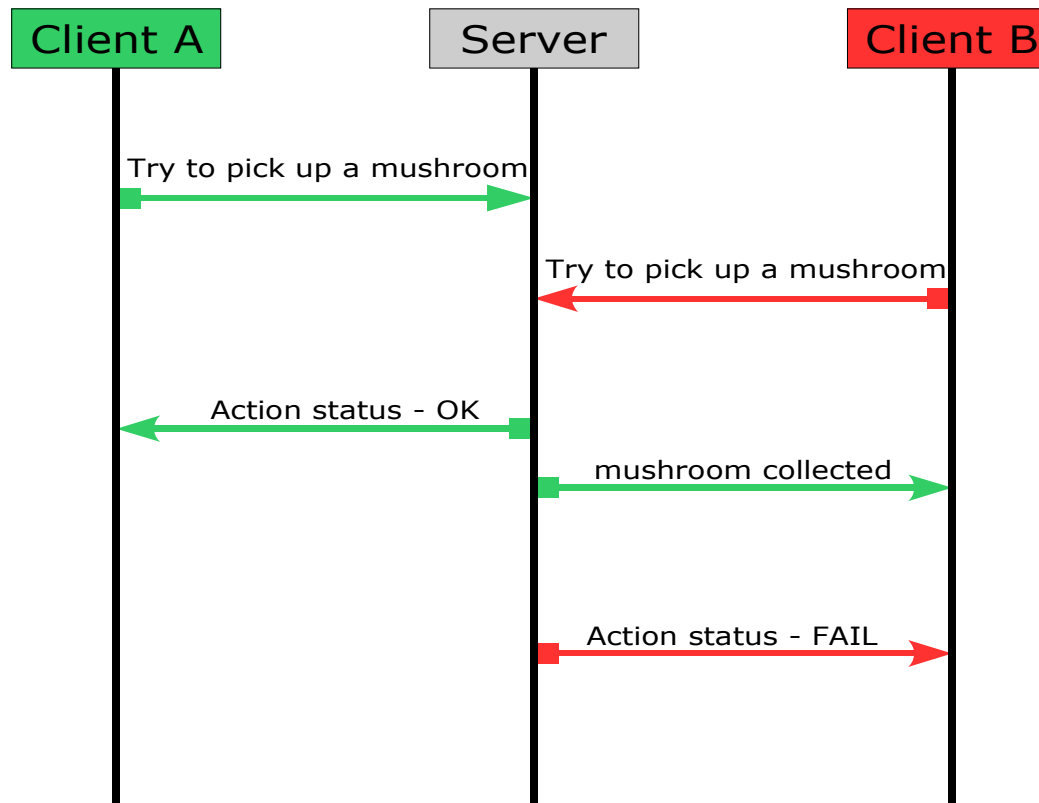
Software and Hardware Architecture

- Separate machines running network traffic & user actions



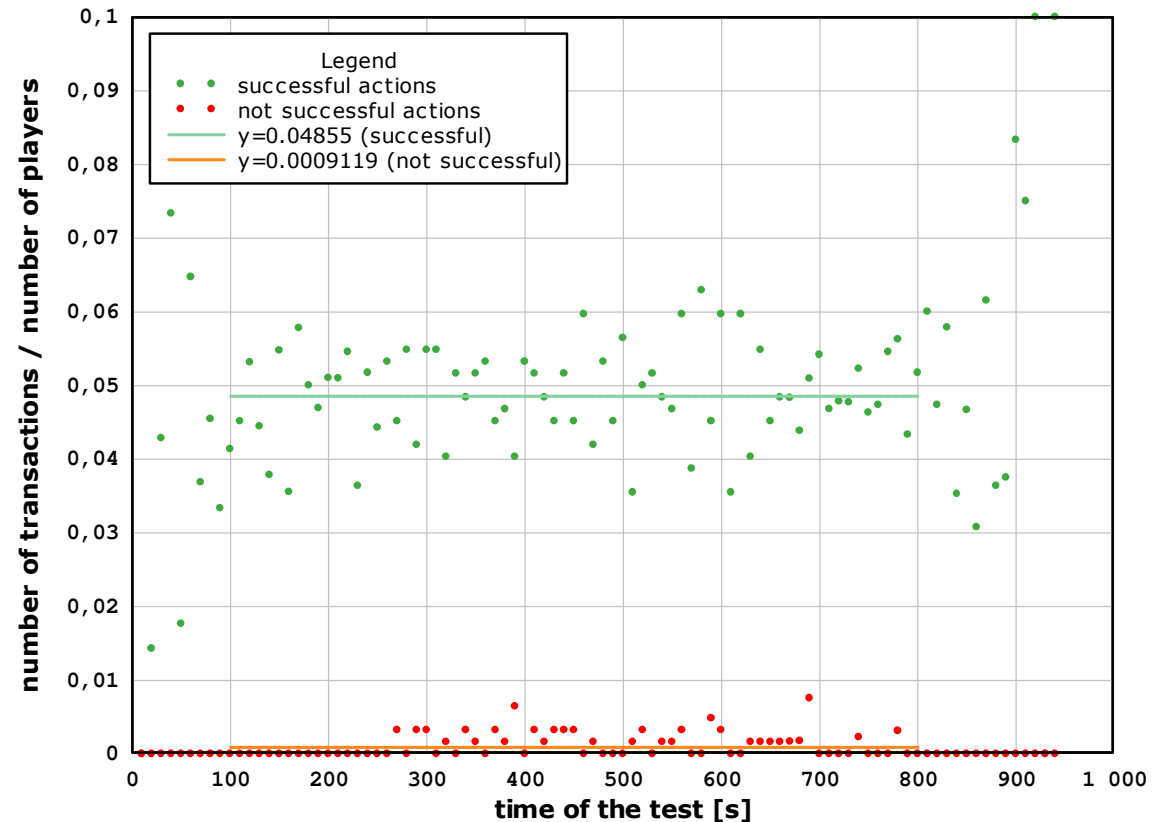
Synchronisation of the Game State

- Players see inconsistent states in relation to the server



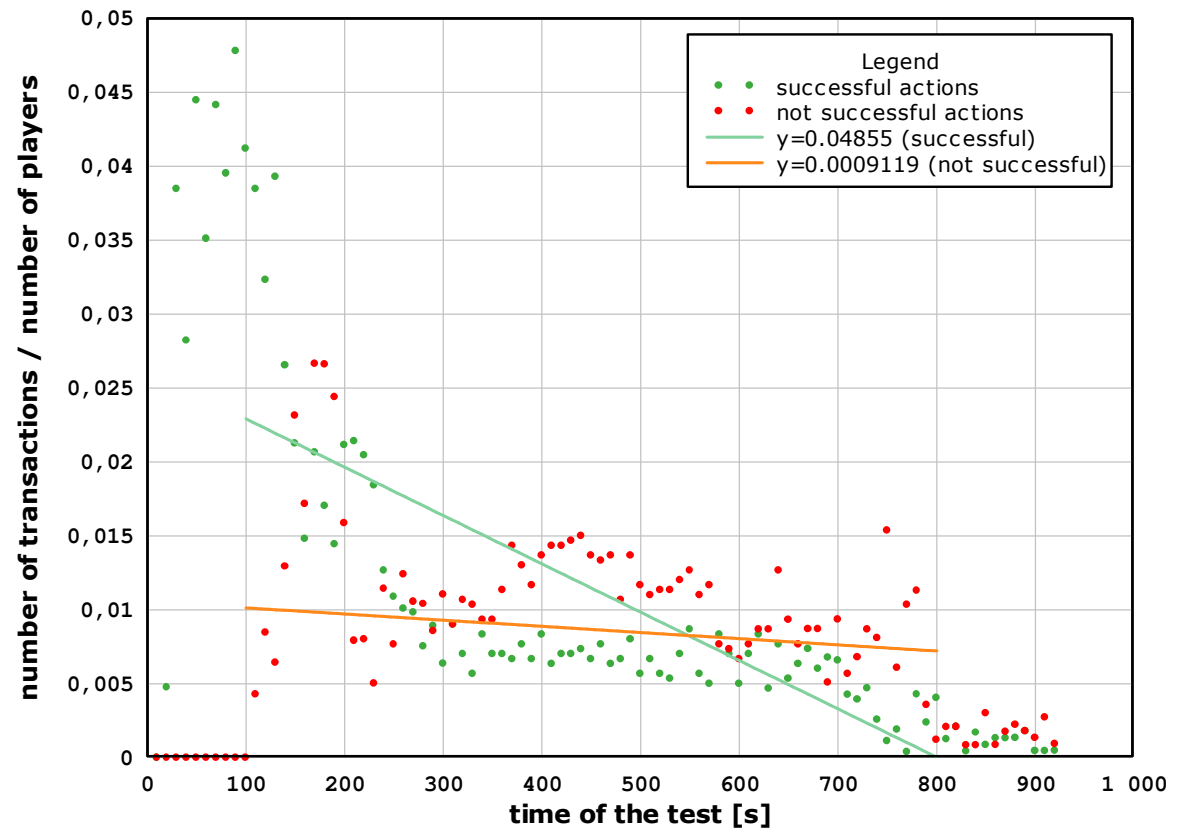
Synchronisation of the Game State

- When the Server is not overloaded almost all actions are successful
 - *Players see less inconsistencies in relation to the server*



Synchronisation of the Game State

- When the Server is overloaded more actions fail
 - *Players see inconsistent states in relation to the server*



Conclusions

- **Erlang/OTP is excellent for MMOG systems**
 - *Distribution*
 - *Libraries*
 - *Powerful open source tools*
- **Rapid development of the prototype allows**
 - *Evaluation of different algorithms*
 - *Evaluation of hardware configurations*
 - *All in a relatively short period of time*
- **Future work**
 - *Algorithms not dependent on the player's position*
 - *Refactor and fine-tune Mnesia and distribution algorithms*
 - *Bring to market in the real world*

Questions ?



Contents

- **Massively Multiplayer Online Game Market**
- **Challenges**
- **Prototyping**
 - *Using Erlang/OTP*
 - *Client and Server Applications*
- **Benchmarking and Stress Testing**
- **Software and Hardware Architecture**
 - *Network and Server Nodes*
- **Synchronisation of the Game State**
- **Conclusions**