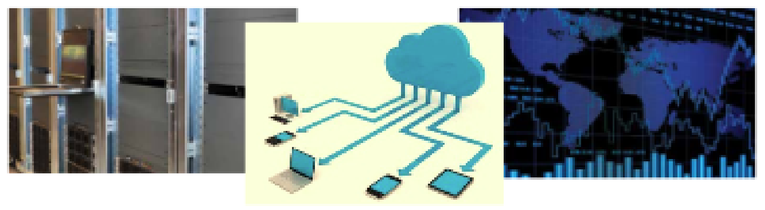
HTML5 WebSocket protocol and its application to distributed computing

Muller Gabriel
Supervisor: Dr. Mark Stillwell

School of Engineering
Cranfield University

Cranfield UNIVERSITY



Distributed Computing and e-Science group

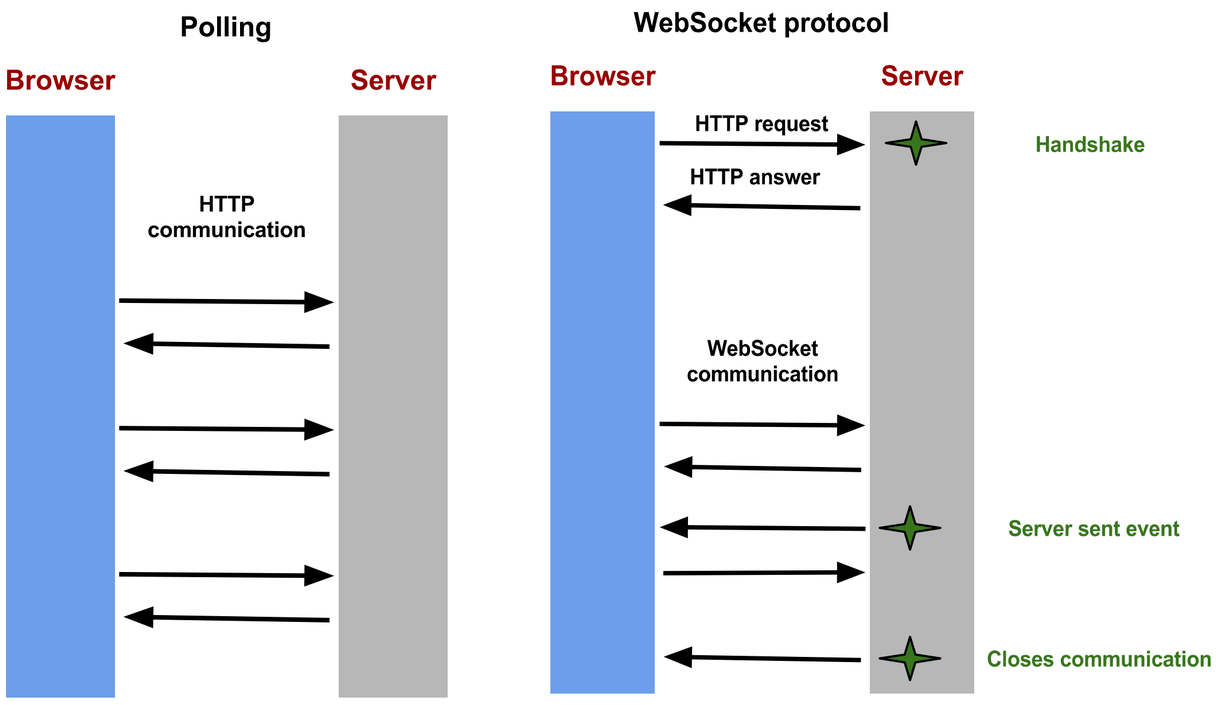
Motivation

HTML is the core language of the web. Daily new products are created to stay permanently connected to the web. WebSockets is the technology enabling this revolution.

As the web itself originally did, WebSockets enable entirely new kinds of applications.

Background

Until WebSockets were invented, client-server communication mainly used polling. WebSockets introduced two major upgrades to communications. Firstly they don't rely on HTTP, avoiding its overheads. Secondly it is a fully bidirectional protocol allowing server sent events.



Objectives

Node.js has been invented to create real-times websites with server sent capabilities. Coupled together, WebSockets and Node.js provide powerful real-time engines.

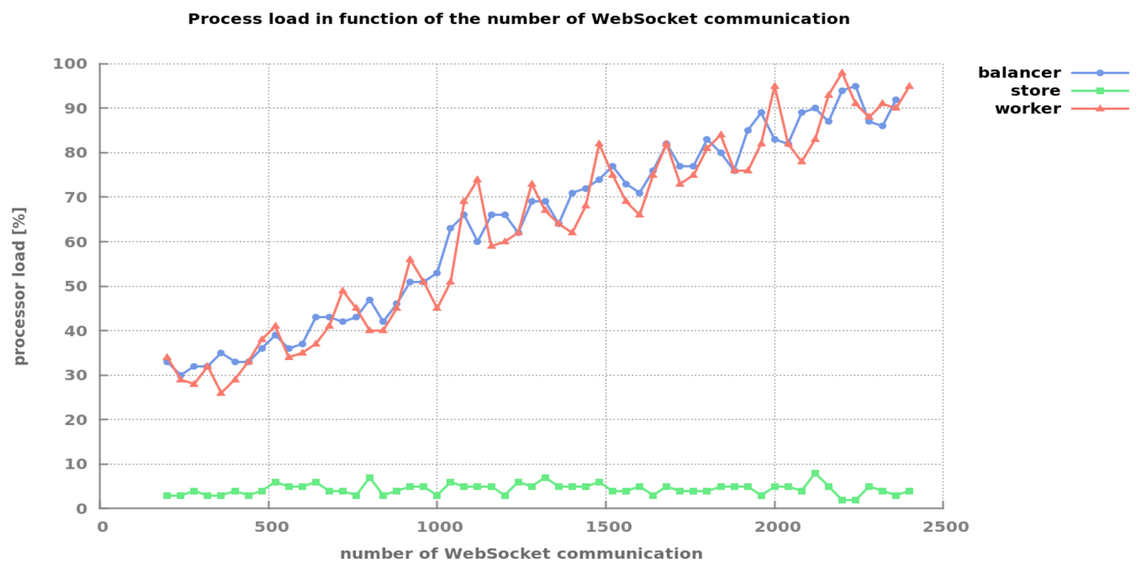
However WebSockets are still an active field of research. Until recently their was no library to easily build a WebSocket cluster making use of all cores available on a server.

SocketCluster is a new real-time engine aiming exactly at that.

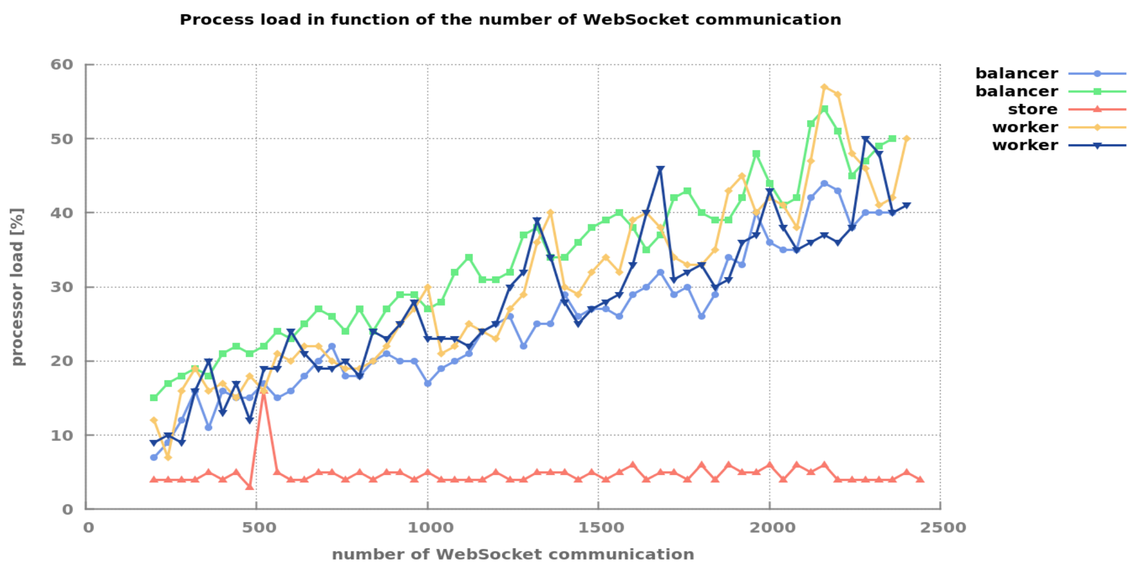
Experiments

SocketCluster is based on engine.io. However as opposed to engine.io it deploys itself as a cluster of processes in order to makes use of all cores on an instance.

There are three types of processes, load-balancer, stores and workers.



Process load in function of the number of WebSocket communication



In the first figure, worker and load-balancer cores have reached their maximum load.

This is not an issue for SocketCluster, new processes can be launched on additional cores. As can be inferred from the second figure, adding a worker and a load balancer drops the average load about to about fifty percent.

Conclusion

SocketCluster is a higly scalable WebSocket server theoritically enabling unlimited vertical scaling. Benchmarking proved an application needs roughly the same number of load-balancers then workers.

Hence, N being the server's cores number, SocketCluster is at least N/2 times more efficient then a basic engine.io implementation.

The Queen's Anniversary Prizes 2005 & 2007