Distributed computation using O'SOAP

Contact: Paul Stodghill

The goal of this project is to build a general framework for doing distributed computation using O'SOAP [O'SOAP]. O'SOAP is a framework for web services processing and programming written in O'Caml. It provides both client and server tools, and is designed to enable legacy applications to be deployed as web services quickly and easily.

The conventional web services model involves the interaction between a single client and a single server. First, the client sends a message to a server. After performing some computation, the server sends a response back to the client. This basic model can be extended to handle asynchronous communication and other patterns, as is done in O'SOAP.

Sometimes a computation is too expensive to perform on a single server. One solution is to buy a larger server. Sometimes, however, the computation can be easily partitioned into small independent work units that can be distributed to a large number of compute nodes somewhere else on the network. This model of distributed computation is used by [SETI@Home] and [distributed.net].

This project involves building a distributed computation framework using O'SOAP. In the project, you will have the following three sets of computing nodes (for simplicity, you may use a single computer for all three sets, but it would be cooler if you actually used a large number of computers).

- Clients: A client node will submit a computation request to the server and poll the server until the results of the computation are available. Multiple clients can initiate concurrent computations.

- Server: The server will respond to client requests by partitioning the computation into a number of work units, which it will distribute to the slaves. It will also collect responses from the slaves and aggregate them into a single result for the client.

- Slaves: There will be a "large" number of slaves in this system. When idle, a slave will poll (What frequency should be used? Random? Exponential decay?) the server to see if a work unit is available. If work is available, the slave will perform the computation and then upload the results to the server.

Here are some issues that you will have to think about if you choose this project.

- What computation should you focus on? It will be one that is essentially "embarassingly" parallel. POVray is one possible example. Code breaking is another possibility.
• How should you handle slave node failures?

• How can you allow the client to specify what computation should be performed?

• Having 1000's of slaves polling a single server is not scalable. What other approaches might be used for very large systems?

References

[O'SOAP]
  http://www.asp.cornell.edu/osoap

[SETI@Home]
  http://setiathome.ssl.berkeley.edu/

[distributed.net]
  http://www.distributed.net/