Meeting agenda

Meeting	1 – Kick-off meeting
Date	Wednesday, September 29, 2010
Time	09:30
Venue	D20.19
Invited	Pierre Kuonen, Roche Jean-François, Valentin Clément, Jonathan Stoppani

Agenda

1. Definition of the goals, the tasks and the constraints that the project has to fulfill;
2. Explanation and overview of the development process of the project itself;
3. Clarification of some points about the current architecture and the specific needs of the Python implementation.

Questions

1. How are (protocol level) errors handled (non existent methods, erroneous requests,…)?
2. Is an asynch I/O implementation of the Python model feasible? Threaded programming is not too performant on Python because of its GIL (Global Interpreter Lock).
3. In which order should I "implement" the different components (Protocol/Serializers, Client, Server,…)?
4. Does a parallel object run in an isolated environment on the remote side (e.g. is it a process of itself)?
5. Are there some convenience methods for the callbacks mentioned for the async requests or the whole subsystem has to be coded "by hand" each time?
6. Theoretically it is possible to "translate" the java code to Python and to offer thus a similar implementation. If so, which parts of the Java implementation should/could be improved?
7. I read something about a remote logging service, can you tell me something more about it?
8. Are there already some test suites to test the conformance of eventual third party or alternative implementations?
9. How is the protocol structured? How many calls per TCP connection? If more then one, what happens when a client sends a second sync call while waiting for the reply to the first one? Is it a precondition that a client can only send one request at a time?
10. After some analysis with Wireshark, I found that the JobManager (:2711) is only contacted once and in the middle of the process, what are all connections which happen before? How can the port numbers be defined?
11. During the analysis I also noted that a simple object instantiation needs 3 TCP connections, but as soon as some object descriptors are added, this number grows "exponentially" (with two `ods`, more than 15 TCP connections). Is this normal, is there a detailed protocol description?
12. Could it be an idea to define/adapt a protocol and a serialization method for the POP model, instead of relying on a simple TCP stream with raw serialization?

Resources

1. http://bert-rpc.org/
2. http://incubator.apache.org/thrift/
3. http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/DistrObjects/DistrObjects.html
4. http://twistedmatrix.com/trac/
5. http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/NetServices/Introduction.html