Gino as a server

At the moment Gino is run from cron. When it wakes up, it checks to see if an instance of itself is already running, and exits if it is, to not step on itself.

Gino also is fairly verbose in generating a log file (that blew the glast04 /pipeline/ partition last week and has been moved to u12/pipeline/). The log is rather hard to parse since Gino spits out processes that write to the log asynchronously. And huge.

If we want to check aliveness with the resource checker, the best option currently is to check the last touched date on the log file. The Gino process itself cannot respond to queries.

Matt has suggested we move towards a java server, initially wrapping the scheduler perl script.

He says the java wrapper can (out of the box, more or less, I think):
*handle log files, automatically breaking them up into nice-sized chunks
*provide network connections for querying, though this would only tell you the wrapper is running in that stage of evolution.

I imagine there are other features I am forgetting. It would be nice if Matt could elaborate, giving a fuller feature list, a pointer to further reading and perhaps a simple demonstration example that wraps a perl script issuing a print statement or two?