Rspec Design Document

Note: in the future, we are likely to move to the Rpsec Extension Format defined by ProtoGENI to ease our integration with them. Please check our API (https://seattle.cs.washington.edu/wiki/SeattleGeniAPI) for up-to-date information about which rspec format our production clearinghouse currently supports.

An rspec is a dictionary that is passed via XMLRPC to the SeattleGENI website. The only required field is an 'rspec_type' key that has a value (of type string). The format is meant to be extensible and will be extended from time to time.

The 'rspec_type' has a string value and defines the type of resource request. For instance, this is used to specify which sorts of resources are applicable to the request. The other fields are specific to that rspec type. Note that it is coincidental that all of our current rspec types have a number_of_nodes field. This may not be true for future types we define.

We chose this approach because it is simple to implement and expressive enough for our primary use cases. We examined approaches that were more expressive but found that they increased the complexity of our implementation without providing functionality needed by our users. We also were concerned about possible privacy implications of having a more expressive method for choosing nodes. As a result we chose to wait to add more expressive rspec types later. We do not perceive any limitations from our format because we can always extend the format to be more specific, expressive, etc. However, while our format does not have limitations, it is clear that at some point in the future more rspec types will need to be defined.

As of August 2009, there are currently three types of rspecs defined:

{'rspec_type':'lan', 'number_of_nodes':N}
Returns N nodes all with the same starting three octects of the IP address.

{'rspec_type':'wan', 'number_of_nodes':N}
Returns N nodes all with different starting three octects of the IP address.

{'rspec_type':'random', 'number_of_nodes':N}
Returns N nodes.

The latest list of valid rspec types can be found at: https://seattle.cs.washington.edu/wiki/SeattleGeniAPI

Passing in an rspec requests that nodes be assigned to the current user. In our SeattleGENI clearinghouse, we first examine the 'rspec_type' field and then call a specific function to handle the rest of the rspec. Resources are obtained by issuing database queries to a database which stores the node information and by performing a little processing on the results. The database queries and processing differs based upon the rspec type. We then contact the nodes to set up the user credentials so that the user actually obtains the resources.