[unchanged]

## Additions to GINI and high level features

As mentioned above, the main design focus was to create an external component to GINI for ease of future integration. In the code, all the alterations have been centralized in the file */frontend/UI/Configurationi.py*. To make the changes active, simply download GINI, and overwrite the configuration file with the one provided with this project.

Within GINI, there are two ways to access the cloud functions: either from the initial setup wizard, or under the ‘server’ tab in the Config/Options section. In either of those places, selecting the ‘connect to cloud’ button will generate a popup window which gives access to the cloud DataBase.

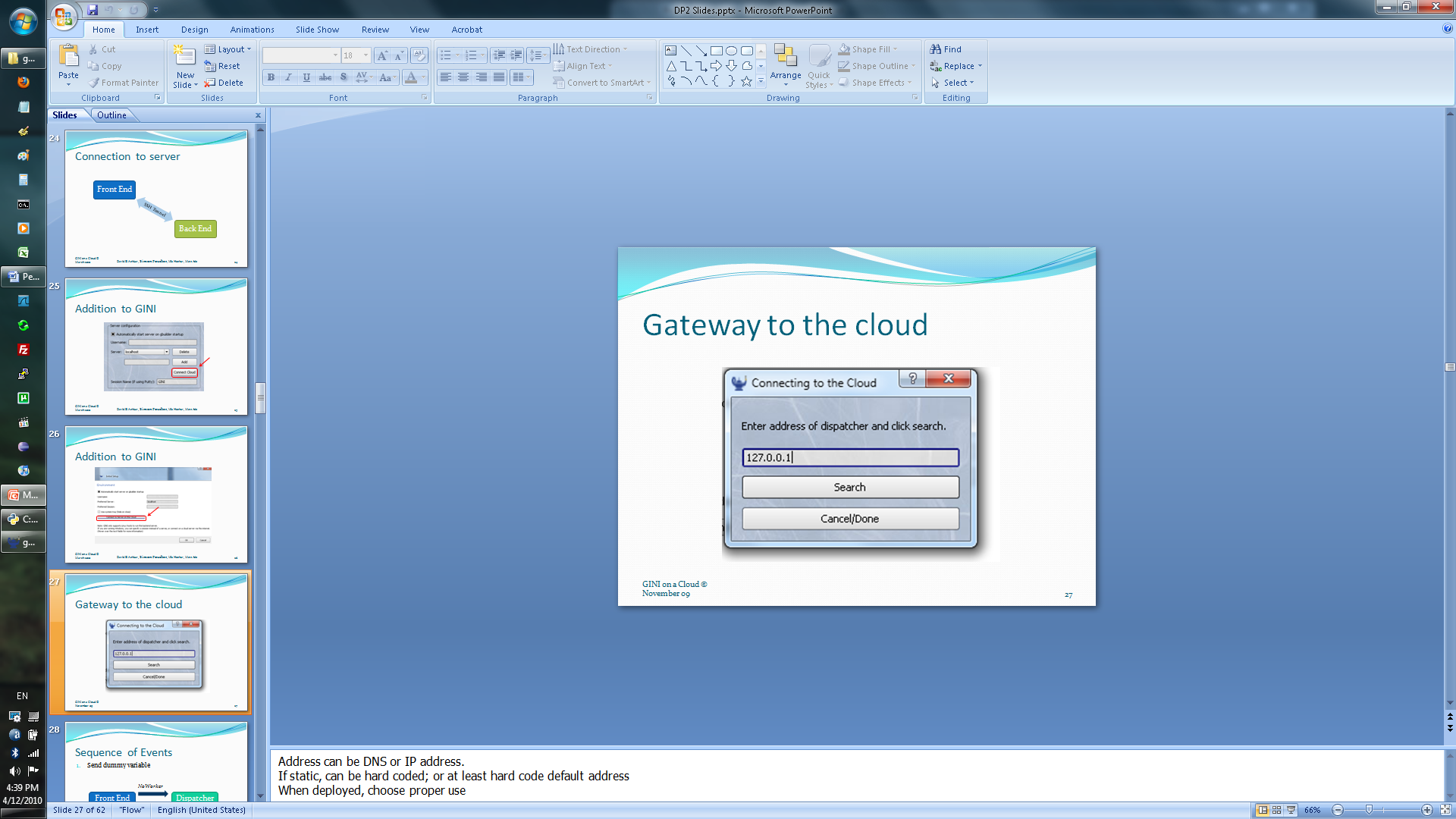
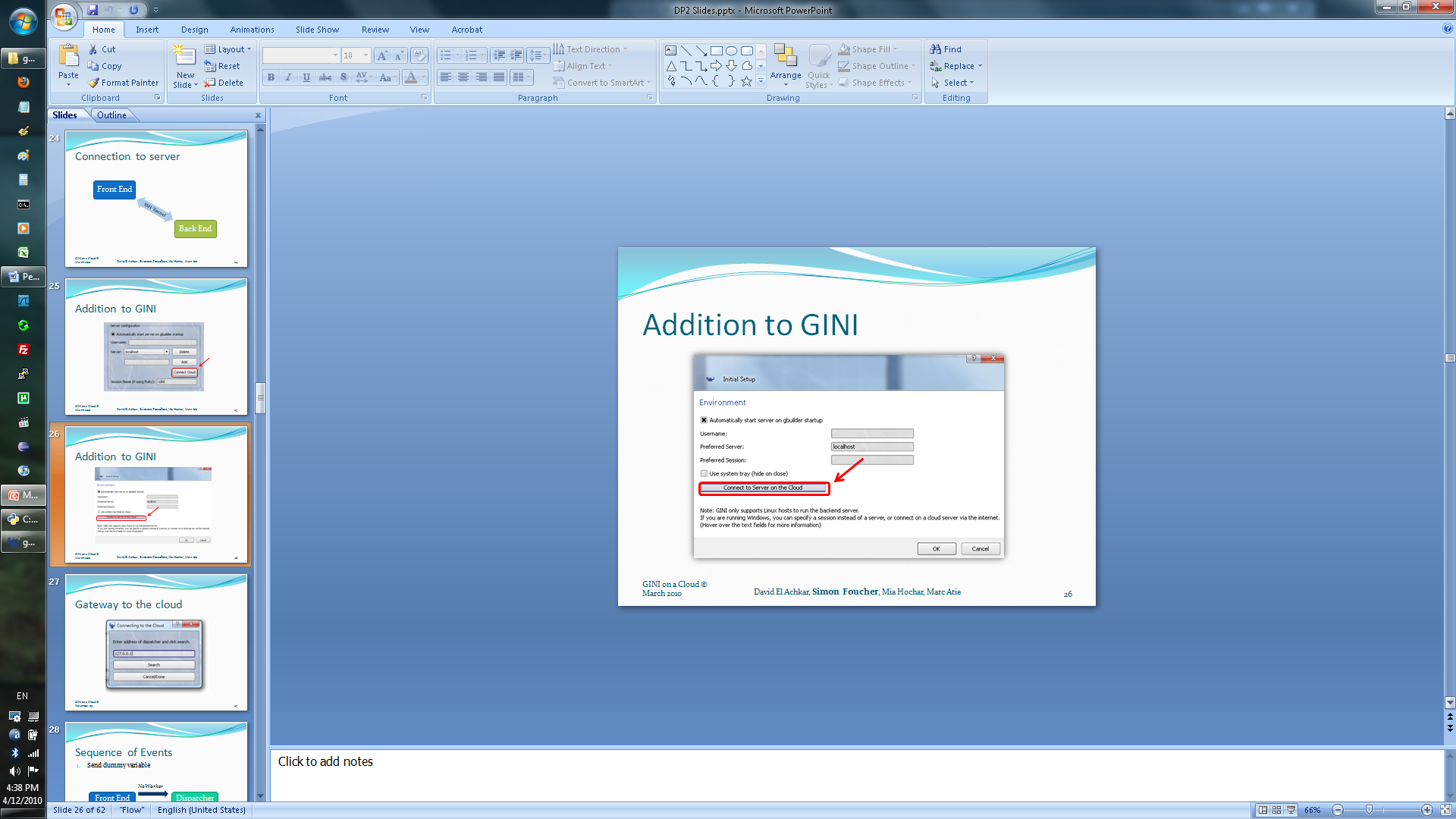


Figure 3. GINI Wizard Modifications

Figure 4: Access to the cloud component is provided from the initial setup Wizard (left), or from the server tab in the Config/Option dialogue (center). Pressing either of the buttons will generate a popup window with access to the cloud workforce.

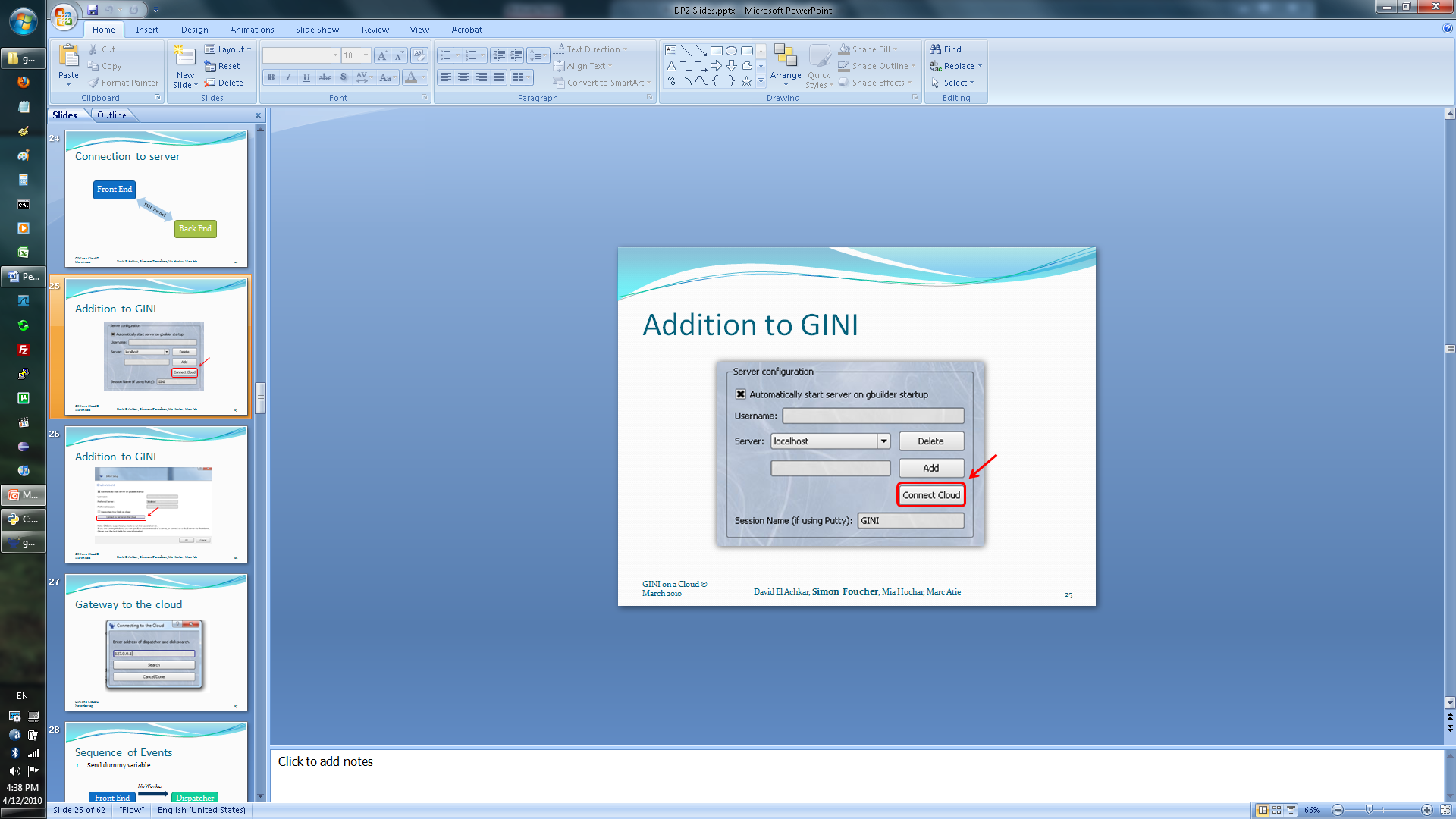


Figure 5. GINI Options Modifications

Once the FindCloudWorker object is built (the popup windows), the user has access to the cloud workers. The top text box gives the user the freedom of selecting where to connect to the Dispatcher. It was implemented as a text box during testing, and the default text value is embedded in the code and can easily be edited. In the deployment of the software, this textbox could easily be removed and replaced by a hidden static variable with the DNS name of the server hosting the Dispatcher. When the user presses the ‘search’ button, GINI sends a dummy variable to the Dispatcher and wait for a reply. In the test conducted, the reply came on average within 20ms, so the delay is negligible to the user.

If the variable returned is the same as the dummy variable sent, an error condition is recorded. Otherwise, the DNS name returned is added to the ‘server’ option field and selected as a default server. In the current implementation, the user still has the option to use this server or choose another one. In the final deployment, if the GINI backend is to run only on ‘hidden’ cloud workers, this whole operation could be migrated behind the scene and performed automatically whenever a client connects to a server. This would greatly reduce the cognitive load on the end user with regards to understanding how GINI actually works, and enable him/her to focus primarily on the simulation being performed. Before launching the server, the user will still have to enter his username to access the server. The testing has been performed using standard McGill IDs to successfully login to McGIll’s Linux machines.

Overall, the greatest advantage of this upgrade is that the user no longer needs to have any knowledge of the location and availability of Linux machines to run gServer. A user can simply fetch up-to-date information on the workforce available and the proceed with launching GINI’s backend.

## Dispatcher

[unchanged]