A Grid-enabled Virtual Screening Gateway

Mohammad Mahdi Jaghoori, Allard J. van Altena, Boris Bleijlevens, and Silvia D. Olabarriaga
Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands
Emails: {m.jaghouri, a.j.vanaltena, b.bleijlevens, s.d.olabarriaga}@amc.uva.nl
The first two authors contributed equally to this paper.

***Abstract*—In computer-aided drug design, software tools are used to narrow down possible drug candidates, therefore reducing the amount of expensive in vitro research by a process called virtual screening. However, searching for drug candidates among a huge number of alternatives requires extensive computation. In this paper, we describe a science gateway for virtual screening that has been tailored to the specific needs of our local users. By reusing the generic architecture and code of a previously developed science gateway for another scientific discipline, it took us only two months from early requirements analysis to obtain a running gateway. The early empirical results show (1) considerable speed-ups, thanks to usage of grid infrastructure; and, (2) user satisfaction, thanks to the user-centred design of the web interface and automated data management.**

I. Introduction

In drug discovery research, a drug candidate is typically a small molecule (*ligand*) that can make a strong binding to the drug target. The target is typically a protein that is either a *receptor* responsible for transmitting signals to a cell, or an *enzyme* that enhances the rate of chemical reactions. A drug candidate can interfere with these biological processes incurred by the target molecule, if it can bind with high affinity at specific binding sites. Therefore, scientists usually search available libraries of ligands for putative drug candidates; this process is called *screening*. Biochemical screening, however, requires expensive lab equipment and takes considerable time. Alternatively, the initial phase of drug discovery nowadays involves virtual screening software tools that simulate binding of ligands and calculate binding affinities. AutoDock Vina [**autodockvina2010**] is one of the most advance software suites in this area.

In this paper we describe a Grid-enabled Screening Gateway (GSG) with AutoDock Vina developed in the interest of biochemical scientists at the AMC[1]. This gateway leverages the Dutch grid infrastructure resources in order to reduce months of computation time down to a few days of waiting time. This makes the analysis of very big ligand libraries feasible and enables biochemists to focus on the interpretation of the results rather than on performing the computations.

The motivations to create a new AutoDock Vina gateway, despite the existence of analogous ones (see Section II), are the following. First, to offer sufficient flexibility for manipulation of inputs and outputs. The users need full control both over the relevant execution parameters passed to AutoDock Vina, and over the part of the output results they want to download (see Section III and IV-B). Second, to provide automated data management, regarding both the various ligand libraries that are applicable in different studies and the provenance of the experiments that users run over time (see Section IV-D). And finally, to have the data processed by a portal and infrastructure for which a trust relationship has been established between the end-user and the service providers.

The contribution of this paper is twofold. On the one hand, we present a simple-to-use gateway for fast processing of AutoDock Vina analyses, tailor-made to the specific needs of biochemists at the AMC. On the other hand, we show that, based on the flexible gateway architecture initially reported in [**ShahandCCPE2014**], implementing any new gateway boils down to designing a workflow and a user interface (see Section IV). Such gateways benefit from all the provided services, including provenance, administration functionalities, automatic user notification, etc. Despite the small development time span of the GSG, the initial evaluation results discussed in Section V are promising.

II. Related Work

There are already a few systems offering high-throughput virtual screening [**MosGrid2012**, **UoW2013Azure**, **FlexScreen2011**, **dovis2008**]. Solutions like FlexScreen [**FlexScreen2011**] and DOVIS [**dovis2008**] aim at accelerating the throughput by using distributed computing resources, but lack a user friendly interface for biochemists. The most closely related to our work are the desktop grid portal for AutoDock Vina [**sztaki-autodock**, **UoW2013Azure**] and the MosGrid portal [**MosGrid2012**], as they present easy-to-use user interfaces. The former supports both AutoDock and AutoDock Vina, providing some functionalities that are a surplus to our users, but at the same time it also lacks some necessary functionality for our users. Namely, our users want to see more details in the output, i.e., be able to download *all* of the files that AutoDock Vina generates. The MosGrid portal incorporates many tools for molecular simulation using grid infrastructure. However, it is only available to users which are connected to a German institution, therefore this gateway was not available for our users. Deploying the MosGrid software on the Dutch grid is also not practical because of the different underlying technologies and policies. Furthermore, none of these gateways meet the extra-functional requirements of our users regarding trust and support for the local work process (see Section III). We therefore chose to base the implementation of

[1]The gateway is accessible on http://docking.ebioscience.amc.nl/.

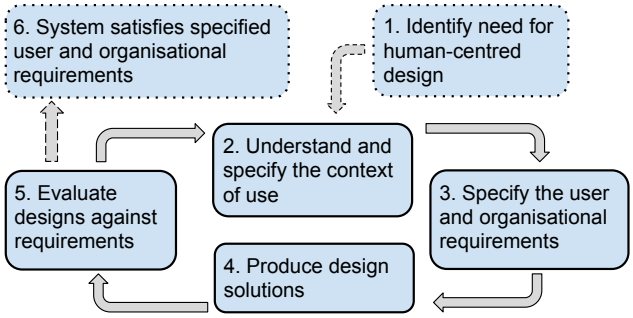

Fig. 1: Simplified process model described in the ISO 9241- 210:2010 for human-centered design of interactive systems [**ISO9241-210:2010**]. This iterative process involves incrementally refining the user-interface design until the user and organizational requirements are met.

the new GSG on the architecture and software of the existing NeuroScience gateway [**ShahandCCPE2014**], which has support for some of these functionalities. The implementation of GSG is aligned with the work process of the end-users, and provides the necessary trusted environment for this user group.

III. A User-Centered Design

One of the main goals of science gateways is to ease the use of available infrastructures for researchers so that they can better focus on the scientific interpretation of the outcome of their analysis. This puts the users (i.e., the researchers) and their needs at the centre of the interface design. Therefore, the ISO standard for human-centered design [**ISO9241-210:2010**] is a suitable method for our purpose (block 1 in Fig. 1).

This ISO standard describes a few important factors that are influential while using the system, referred to as the context of the system (block 2 in Fig. 1). The factors applicable to our situation are characteristics of the users, organizational tasks, and the physical environment: the system should be designed to maximally support the *biochemical* users in *performing virtual screening* task in the environment of *the AMC organization*. Giving context to the requirement analysis process helps in preventing identification of false requirements and wrong interpretation of requirements.

In order to identify the user and organizational requirements, (block 3, Fig. 1), we conducted a series of interviews with an expert biochemist at the AMC organization. This user supports the local screening studies at our organization and will be the primary user of the GSG. Table I summarizes the functional requirements in terms of the inputs and outputs. Additionally, the following points are identified as significant *extra-functional* requirements of the gateway:

- **Scalability**: the user should be able to process libraries that may contain hundreds of thousands of ligands in a timely fashion (i.e., should not take weeks or months to complete the experiment).
- **Provenance**: when the user wants to download the output, the input items (i.e., configuration file and receptor file) should be packaged with the output. The gateway should take the AutoDock Vina output files and generate an overview of the data.
- **Security**: the aggregated output data may not be publicly available because it has intellectual property value. The gateway and infrastructure should be trusted by the users.
- **Ease of use**: selection of ligand libraries or individual ligands should be possible. The gateways user interface for providing inputs should resemble the layout of AutoDock Vina. On the outputs, the user should be able to select any given percentage of the best binding ligands and download them.

TABLE I: Inputs and outputs as requested by the user

Inputs

Ligands	Ligands are possible drug candidate molecules provided by the ZINC database [**zinc**] and grouped into libraries.
AutoDock Vina Configuration	Size and centre of the area of interest, a 3D box (in X, Y, and Z parameters). The surface of the receptor that falls within the AoI is used by AutoDock Vina.
	Number of runs* performed per ligand with different random seeds for "dropping" ligands on the receptor surface.
	Exhaustiveness* defines a parameter to the built-in heuristics which determines how many variations are made.
	Energy Range*, calculated energies that do not fall within this range are discarded from the output to limit the number of output results.
Receptor**	Three-dimensional structure of the receptor the user is interested in. Optionally, the user may also provide a different arrangement of the atoms using a second receptor file.

* optional input; when unspecified, AutoDock Vina defaults will be used.
** For readability, we abuse the term receptor to refer to an enzyme, as well.

Outputs

Raw Output	Encompasses all of the files that AutoDock Vina generates per ligand.
Computed Binding Energies	A sorted list of binding energies tagged with their corresponding ligand identifier. The sorting puts best binding ligands on top.

After requirements analysis, we made a design in an online mock-up system that creates very neutral designs (block 4 in Fig. 1). This neutral design minimizes the visual impact of the layout and optimizes the placement of data elements on the page. The end-user provided some feedback on the mock-ups, which was then incorporated in the next iteration of the design cycle (block 5 in Fig. 1). For the GSG, we have performed three iterations so far to get to the current (mock-up) design; we will continue to use this standard to accommodate further needs of the users.

IV. Design and Implementation

Here we discuss the design and implementation of the GSG from different perspectives. First, we discuss the software architecture, including the data model and different components used in GSG. We continue by describing the design of the

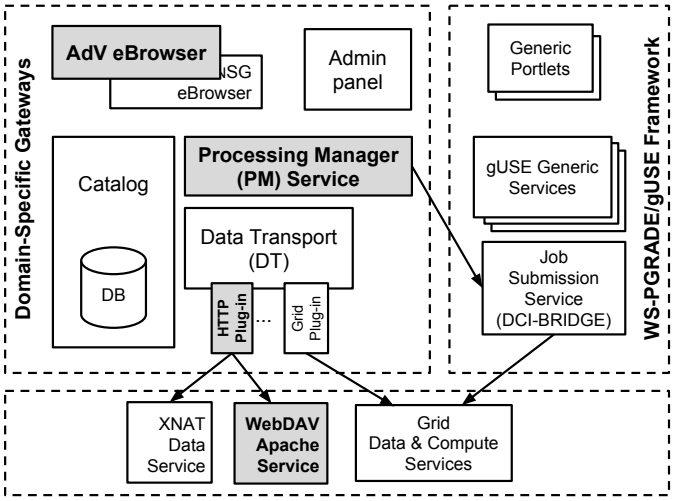

Fig. 2: GSG reuses the architecture of the NSG [**ShahandCCPE2014**]. The grey boxes represent the new components (i.e., AutoDock Vina eBrowser and the data resources for it) and components with considerable changes (i.e., Processing Manager Service). Arrows are used to show cross-boundary communications.

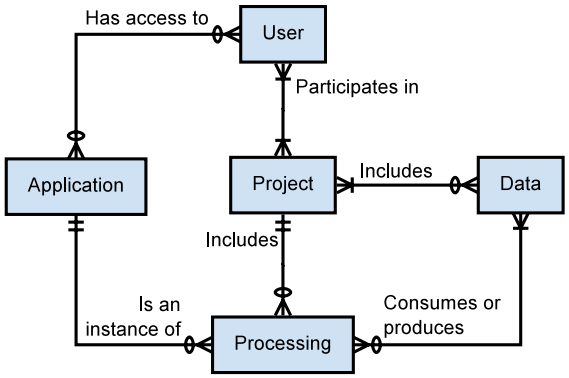

Fig. 3: Simplified generic data model which has been adapted from [**ShahandCCPE2014**]. GSG creates `Projects` dynamically to store the `Data` entered/selected by the users online. An instance of the `Processing` entity is an execution of the `Application` (in this case AutoDock Vina) on the set of specified `Data` in the context of the given `Project`.

user interface based on the requirements analysis. Then, we explain how AutoDock Vina application is wrapped into a workflow, so that it can be used in GSG. Finally, we cover further implementation details regarding provenance and data management.

A. Software Architecture

The architecture of GSG is shown in Fig. 2. It reuses the NSG components [**ShahandCCPE2014**], and is therefore also based on gUSE [**wspgrade12**]. gUSE is a generic workflow management system that, among other functionalities, supports split-and-merge workflows by automatic submission of multiple parallel jobs to the grid infrastructure and collection of their outputs. Furthermore, gUSE offers the possibility to automatically or programmatically resume workflows when they fail in the middle. Based on this functionality, NSG and therefore GSG enable the system administrator to manually resume workflows whenever human intervention is needed to fix the cause of the failure. Section IV-C shows how AutoDock Vina is wrapped into a gUSE workflow.

Components from the previous gateway that have been reused in the GSG are: Catalogue, Processing Manager (PM) and Data Transport (DT). For our implementation, adaptations have been made to the Catalogue and PM components. A new user interface has been designed to specifically serve the needs of the GSG end-users; nevertheless, a part of the user-interface related to the administration of the gateway could be reused.

Catalogue: The gateway operates based on the data model presented in Fig. 3. This generic data model makes it possible to store basic provenance information about the performed processing actions. The Catalogue provides functionality to store and query this provenance information. Furthermore, the Catalogue keeps track of the users, their data and the applications. Note that although the system is designed specifically for AutoDock Vina, the generic data model allows multiple applications to be used. This gives the GSG the potential of being extended with other biochemical applications in the future.

Processing Manager (PM): This component uses the information in the Catalogue to submit applications, and stores the monitoring information back to the Catalogue. Note that the PM has evolved since the NSG was released, the most important change being the addition of a web-service interface. This means that the user interface is technologically decoupled from PM and its internal tasks. For example, now it is also possible to use the PM from within legacy systems.

Data Transport (DT): This component is used by the PM to move the input and output data as needed between different resources. In GSG, the HTTP plugin of DT has been upgraded to additionally deal with the WebDAV protocol; that is, to perform create, read, update, and delete (CRUD) actions on a server through the HTTP protocol. Therefore, in principle any WebDAV-enabled storage service can be used, provided that it is trusted by the users.

B. User Interface

Three HTML pages were created from the user-centered design process: `New Job`, `In Process`, and `Outcomes`. The completed designs for `New Job` and `In Process` can be seen in Fig. 4. The `Outcomes` page uses the same layout as `In Process`, with some additional data elements.

Input design: The `New Job` page is created to contain all the required input items, i.e., receptor file, ligands, and AutoDock Vina configuration items. The fields are ordered to match the AutoDock Vina software layout. Ligands can be selected either individually or by selecting a whole library (Fig. 4). Since the number of available ligands is very big, search functionality has been added to easily track down specific ligands. Inputs from the HTML form on the `New Job` page are processed in the back-end and put in the right format that can be processed by the gUSE workflow wrapping AutoDock Vina.

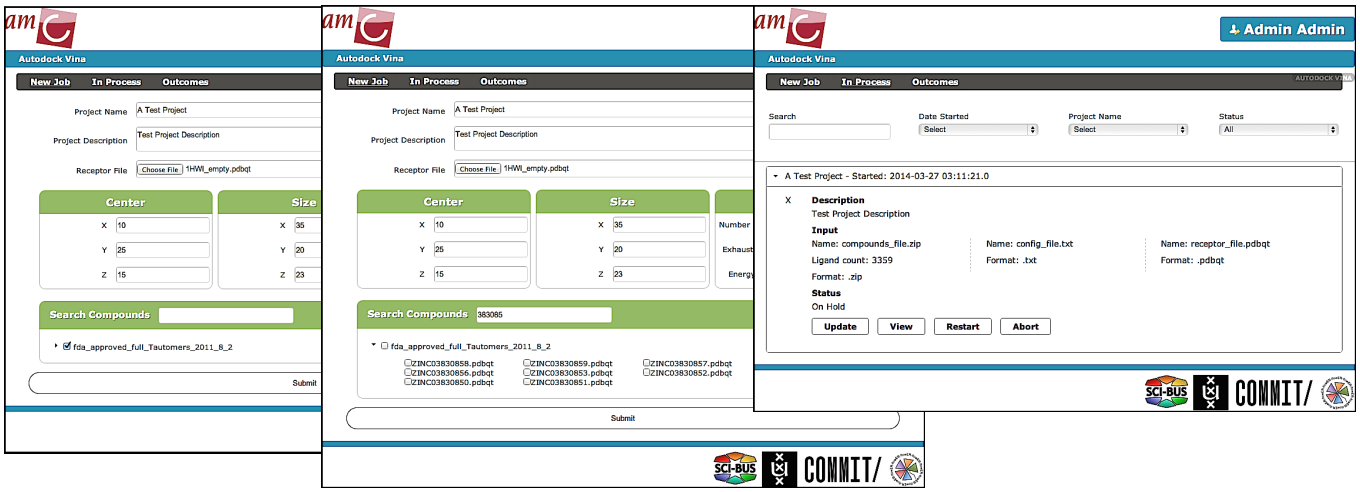

Fig. 4: Design that has been implemented after three iterations of the user-centered design cycle. The 'New Job', 'New Job with expanded ligand library', and 'In Process' pages are shown.

Output design: The `Outcomes` page only shows the jobs that have successfully finished and whose output data are ready. The incomplete and unsuccessful jobs are shown on the `In Process` page. Both pages further expose textual search on job name and description, and can present results sorted by date or by job name. On the `In Process` page, jobs can be filtered on their status (i.e., in progress, on hold, etc.). The `Outcomes` page contains a table and graph portraying the sorted list of binding energies (i.e., ligand affinities) generated by the workflow. In the table, ligand names are linked to the ZINC database [**zinc**] for detailed descriptions on the ligands. Such graphical representation of the data lowers the workload of the user. When the user wants to download the AutoDock Vina output, s/he is able to select the percentage of top binding energies to include in the output file. The GSG back-end extracts the requested ligands from the workflow output, and packs them into a single file for download.

C. Workflow

The workflow used in the GSG (see Fig. 5) is originally based on the workflow developed at the University of Westminster for desktop grid [**UoW2013Azure**], which was later adapted to run on the Dutch grid[2]. In this work, we further extended the workflow to accommodate user requirements regarding input and output file handling.

The workflow starts with the `Prepare` component, which takes as input, the receptor file (port 0), the configuration file (port 1) and a list of ligands as a zipped file (port 2). The `Prepare` component splits the ligands into distinct groups and passes them to the `AdVina` component, along with the receptor and configuration files. Several instances of the `AdVina` component are executed to process these ligand groups in parallel. Finally, the `Collect` component collects

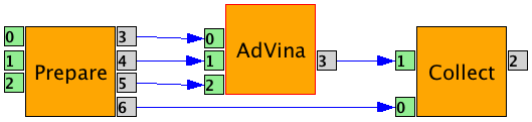

Fig. 5: The workflow for distributed execution of AutoDock Vina (available from SHIWA workflow repository)

all the outputs, creates a sorted list of the binding energies, and packs the outputs into a single compressed file.

The workflow dynamically maximizes parallelization granularity based on two restrictions: a minimum number of ligands per job is needed to minimize the overhead versus computation time; and, a maximum splitting factor is used to lower the overhead on gUSE. Furthermore, the workflow employs the split-merge workflow motif [**Works2013**]: the `Prepare` component passes (via its output port 6) the number of parallel instances of `AdVina` to the `Collect` component. This enables the `Collect` to determine if all the `AdVina` instances have successfully finished. Otherwise, gUSE would allow the workflow to collect partial results even if some of the `AdVina` instances finished in error or were cancelled.

D. Data Management and Provenance

An important added value of the GSG is automatic storage and organization of the input and output data related to each processing. This will enable future reference to the results and allow for reproducibility of the experiment by the scientists.

On the `Outcomes` page, when the output of a job is downloaded, the user gets the related input files that were created by the back-end when the job was submitted (the ligands zip file and the configuration file). This satisfies the user requirements and assures a complete overview of the performed jobs. Furthermore, a special HTML element (accordion, from the jQuery UI library[3]) is used to display the input and output data on each job; this element stacks headers on top of each other. The headers in this case contain

[2]Both workflows are available publicly on the Shiwa repository via the links http://repo.shiwa-workflow.eu/shiwa-repo/public/edit-application.xhtml?appid=2956 and http://repo.shiwa-workflow.eu/shiwa-repo/public/edit-application.xhtml?appid=4256.

[3]Example and description on: http://jqueryui.com/accordion/

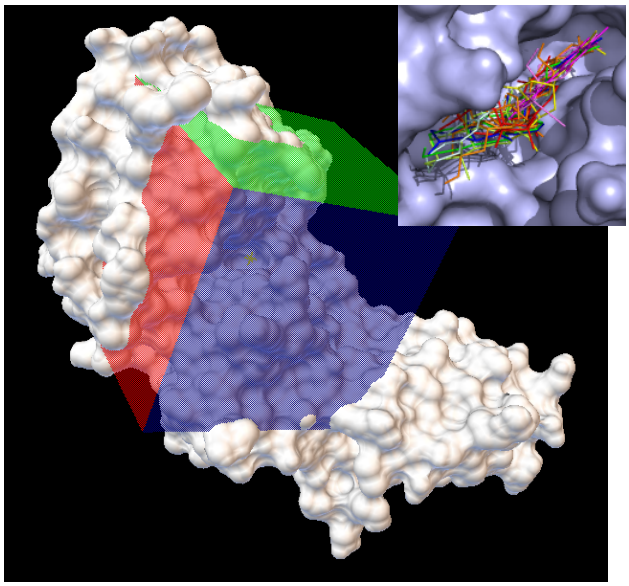

Fig. 6: Search space as defined with AutoDock Tools that were used to prepare the receptor file for virtual screening experiments. A box of 32x32x32 Å was sampled for ligand binding. The inset graphics zooms on the receptor showing 10 compounds with the highest binding energy (-9 to -10 kcal/mol).

the job name and the date the job was started, which is used for sorting. When one header is selected, the content belonging to it (detailed information about each output generated by the workflow) is shown, while all the contents from other headers are hidden.

For long term reference and provenance, the input and output data need to be stored in a permanent, reliable and secure storage. Firstly, the system has a centralized copy of the ligand libraries that are used by the users; these libraries can be easily extended by the administrators if needed for future use cases. Secondly, the GSG keeps track of all input and output data related to each processing. We use a local storage inside the AMC network as a trusted environment. Access to this data is restricted to the user who has submitted the job, and it is only reachable via the GSG.

V. Case Study and Initial Evaluation

As our first case study, we used virtual (in silico) screening to identify putative inhibitors of FTO (fat mass and obesity related protein) catalytic activity. This can help scientists understand better how FTO activity influences body weight, as it has been shown that variants in the gene coding for the 2OG dioxygenase FTO are the strongest hereditary predictors of obesity [**frayling2007common**].

We repeated the experiment a few times with two ligand libraries which, based on user experience, have a high chance of generating good bindings. Fig. 6 illustrates part of the output of this case study, visualized offline using PyMOL [**pymol**]. In this section we present and evaluate the results of this experiment regarding the computational performance and user experience, rather than the biochemical interpretation of the results.

Performance: To assess performance we measured the actual computation time (accumulated CPU time for all jobs) and the response time (as observed by the end-user to complete the experiment). Note that, for practical reasons, we have not compared the performance with local execution times. We also calculate the gained speed-up as the ratio between computation and response time. The response time, and therefore also the speed up, are affected by system overhead, including job queueing and the time it takes before the system administrator resumes a failed workflow. Table II summarizes the obtained results that are discussed below.

For *Library 1 (78 ligands)*, we observed that analysis of small ligand libraries on the grid did not offer much speed-up [0.15, 1.2]. The response time may even be longer than the time actually needed for the computation. Nevertheless, based on these results, we adjusted the parallelism parameters to allow for smaller grid jobs. The end-user still benefited from data management incorporated in the system.

For *Library 2 (2.5K ligands)*, we observed that the larger amount of parallelism makes it almost inevitable that some of the grid jobs fail. The response time observed by the end-user, therefore, depends partly on the delay of the GSG administrator to manually resume the failed jobs. As a result, although the speed-up (in this case 3.9 and 4.8) is not in the same order as the parallelism (in this case 49), still there is considerable gain. This is due to instability and unreliability of the underlying software stack, including gUSE, gLite, etc. By fine tuning these layers, we expect the analysis of bigger ligand libraries will deliver even better speed-up.

User experience: Feedback from the user has been collected concerning the value of computational speed-up for the scientific task and ease of use. Results are presented below.

The possibility to access the Dutch grid to execute the calculations has allowed for faster ligand screening than previously possible on stand-alone desktop computers. This advantage is especially noticeable for virtual screening studies using the medium-sized library (2462 compounds) and is expected to be even higher for larger screening libraries. However, the user also mentioned the need to optimize the implementation to achieve larger speed-ups.

Secure storage and easy access to provenance data greatly simplifies data management. Automation and streamlining of data evaluation is considered extremely useful to lure new users to this portal. Previously, sorting of data on binding energy was performed manually via a command-line interface and was difficult for inexperienced users. In addition to this, direct web-links to information on the docked ligands via the ZINC repository simplifies the interpretation of the results and identification of bona fide binding compounds.

The detailed inspection of the mode of binding of predicted high affinity ligands still requires a separate software package to visualize the outcome of the virtual screening experiment. However, prospective users are expected to have some experience in using molecular viewers because similar software is required to prepare input files, analyze the receptor molecule, and define screening parameters such as the search space.

TABLE II: Initial evaluation results. Response time is since submission until the result is calculated; compute time is the sum of all execution times for all parallel runs. The number of resubmissions shows how many times the system admin had to intervene and resume failed workflows.

	#ligands	parallelism	response time (rt)	compute time (ct)	speed-up (ct/rt)	#resubmissions
Library 1	78	1	19h	3h	0.15	1
		1	3h	3h	1	0
		3	3h	3.5h	1.2	0
Library 2	2462	49	19h	74h	3.9	2
		49	15.5h	74h	4.8	1

VI. Discussion and Conclusion

We described a web-based gateway for virtual screening enabling execution of AutoDock Vina on a grid infrastructure. With an extensive requirements analysis and iterative user-interface design following ISO 9241-210:2010, we created a user interface that fully exploits the capabilities of AutoDock Vina and presents outputs to the end-user in a tailored and flexible manner. The provenance information for the experiments that is automatically maintained by the GSG greatly simplifies data management for the user. As a result, the researchers can focus on the interpretation of the screening results rather than on running the software tools. The GSG however can still be improved in many ways. For example, we plan to integrate the preparation of input data, as well as post-processing of the output, including visualization.

From a performance perspective, we observed that speed-up is much smaller than expected because jobs fail and the experiment needs to be resumed manually. This is partly due to system overload and has made so far impractical to perform the most interesting experiment for the user ($100K^+$ ligands). We are collaborating with gUSEdevelopers on improving execution of workflows with high parallelism.

From an operational perspective, we improved the science gateway administration experience by using the processing manager as a web service. There are two advantages in doing that. Firstly, the development of new gateways has been technologically decoupled from PM and its internal tasks. Secondly, it is now possible to have a single point of reference for administration activities that can be shared by various gateways. This will simplify system operation and maintenance significantly, because in the future only one installation of gUse will be necessary for multiple gateways.

The development of a new gateway in the future will essentially boil down to adapting domain knowledge to the the generic data model of the gateway, besides creating the user interface and workflows for the new applications. The time to develop new gateways is therefore minimized and they get extra features for free, like provenance, automatic email notification, administration panel, etc.

Acknowledgment

We would like to thank A. Benabdelkader, J. L. Font, M. Santcroos, and S. Shahand for their contributions to the workflow and software development. This work performed computations using resources of the Dutch e-Science Grid. The AMC gateway activities are financially supported by various projects: COMMIT “e-Biobanking with imaging for healthcare” is funded by the Dutch Organisation for Scientific Research (NWO); SCI-BUS is funded by European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 28348; and ER-Flow is funded by FP7 INFRASTRUCTURES-2012-1 call under contract no 312579.