PROCEEDINGS OF SCIENCE

Challenges in DNA sequence analysis on a production grid

Barbera D. C. VAN SCHAIK[*†]
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
E-mail: b.d.vanschaik@amc.uva.nl

Mark SANTCROOS
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
E-mail: m.a.santcroos@amc.uva.nl

Vladimir KORKHOV
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
Faculty of Applied Math and Control Processes, St.Petersburg State University, Russia
E-mail: v.korkhov@amc.uva.nl

Aldo JONGEJAN
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
E-mail: a.jongejan@amc.uva.nl

Marcel WILLEMSEN
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
E-mail: a.m.willemsen@amc.uva.nl

Antoine H. C. VAN KAMPEN
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
Biosystems Data Analysis, Swammerdam Institute for Life Science, University of Amsterdam, Amsterdam, NL
E-mail: a.h.vankampen@amc.uva.nl

Sílvia D. OLABARRIAGA
Bioinformatics Laboratory, Department of Clinical Epidemiology Biostatistics and Bioinformatics, Academic Medical Center, Amsterdam, NL
E-mail: s.d.olabarriaga@amc.uva.nl

PoS(EGICF12-EMITC2)039

PROCEEDINGS OF SCIENCE

Modern DNA sequencing machines produce data in the range of 1-100 GB per experiment and with ongoing technological developments this amount is rapidly increasing. The majority of experiments involve re-sequencing of human genomes and exomes to find genomic regions that are associated with disease. There are many sequence analysis tools freely available, e.g. for sequence alignment, quality control and variant detection, and frequently new tools are developed to address new biological questions. Since 2008 we use workflow technology to allow easy incorporation of such software in our data analysis pipelines, as well as to leverage grid infrastructures for the analysis of large datasets in parallel.

The size of the datasets has grown from 1 GB to 70 GB in 3 years, therefore adjustments were needed to optimize these workflows. Procedures have been implemented for faster data transfer to and from grid resources, and for fault recovery at run time. A split-and-merge procedure for a frequently used sequence alignment tool, BWA, resulted in a three-fold reduction of the total time needed to complete an experiment and increased efficiency by a reduction in number of failures. The success rate was increased from 10% to 70%. In addition, steps to resubmit workflows for partly failed workflows were automated, which saved user intervention.

Here we present our current procedure of analyzing data from DNA sequencing experiments, comment on the experiences and focus on the improvements needed to scale up the analysis of genomics data at our hospital.

EGI Community Forum 2012 / EMI Second Technical Conference
26-30 March, 2012
Munich, Germany

*Speaker.
†Corresponding author

PoS(EGICF12-EMITC2)039

1. Introduction

Technological advances and decreasing costs in DNA sequencing have enabled large-scale genomics experiments to study disease. Examples of such experiments are whole genome and exome sequencing (sequencing of DNA), characterization of transcriptomes (sequencing of RNA), and meta-genomics (sequencing of DNA and RNA from multiple organisms in complex samples). In the last few years the scientific and clinical studies became larger in terms of studying complete genomes instead of limited sets of genes. Moreover, an increasing number of samples (e.g. several individuals) are included in one study.

Modern DNA sequencers, also known as "next generation" or "high-throughput" sequencers, work on the basis of massive parallelization of the sequencing process. One sequence run currently produces about 200 million short DNA fragments. Further scaling-up is achieved by using multiple DNA sequencers simultaneously. Many institutes use more than one DNA sequencer[1], and the largest sequencing center now uses more than 150 machines. Development of the sequencing process is swiftly evolving, and with each upgrade of sequencing machines the amount of data that is produced per experiment increases.

Storage and analysis of these data is a challenge [1, 2]. There are developments on reference-based compression for the storage of the data [3] and more efficient algorithms for the analysis. For computing, bioinformaticians resort to buying bigger clusters and exploring the possibilities of distributed computing on grids and clouds (see reviews by Stein [4] and Schadt *et al.* [5]).

Another challenge is to cope with the variety of tools available for data analysis. Since the high-throughput DNA sequencers came to the market, more than 500 new analysis methods and tools were developed and published[2]. To keep up with advances in the field it is necessary to be flexible in integrating or replacing new software in analysis pipelines. Workflow technology is suitable to meet this requirement, because it allows the composition of workflows from modular components that can be reused and shared.

And finally, DNA sequence analysis involves several experimental steps, involving collaboration between various experts. In our organization, an academic hospital, biomedical researchers working at various laboratories prepare samples for DNA sequencing experiments, and subsequently send them for sequencing to a central DNA sequencing facility or to an external service provider. Analysis of the data is performed by the biomedical researchers themselves when knowledge, suitable software and sufficient computing capacity are available "off-the-shelf". In other cases biomedical researchers collaborate with the bioinformatics and e-Bioscience researchers, especially for the analysis of large experiments.

To manage the growing demands for data analysis, at the Academic Medical Center (AMC) we started a pilot in 2008 to use the e-BioInfra platform [6] for the analysis of DNA experiments on grid resources. BLAST and pre- and post-processing steps were integrated in a workflow and several datasets were analyzed in parallel on the Dutch e-science grid[3] as a proof-of-concept. This pilot showed that a 30 fold speed-up could be obtained compared to serial execution [7]. This

[1] See a map of worldwide sequencing facilities: http://omicsmaps.com/

[2] See a list of existing software: http://seqanswers.com/wiki/Software

[3] http://www.biggrid.nl/

BLAST workflow has been made available to end-users from the laboratory, and it is still regularly used today [8].

Since this pilot we have integrated many other bioinformatics tools for DNA sequence analysis in a similar fashion, including: BWA [9], Samtools [10], Varscan [11], the Picard toolkit[4], the GATK package [12] and many more. These components are integrated into workflows that are used mainly by bioinformaticians, and sometimes also by biomedical users directly. Until this date around 40 studies have been performed with the aid of the e-BioInfra platform.

Since 2008, however, the data throughput from the sequencing facility has increased. The data volume increased from 1 GB to 70 GB per file and the number of experiments per study increased as well. This raised several challenges that needed to be addressed. The methodology used in our pilot worked well for some time, but for current and future experiments we needed to revise the strategy. Not only scaling the data became a challenge, but it also became necessary to take into account the complete process, from data generation to transfer to the grid, processing in parallel, and delivery of results to the biomedical researchers.

In this paper we describe the procedures and tools adopted to perform DNA analysis experiments on the Dutch Grid from data generation to sharing of results. Secondly, we describe how the original strategy used in the pilot phase was improved to cope with larger datasets, and we measured the time gain and reduction of errors resulting from these improvements (see section 4). Finally, we discuss the current procedure and suggest further improvements.

2. DNA sequence analysis on the grid

In this section we describe the set-up and challenges faced for performing DNA sequence analysis on the Dutch grid using the e-BioInfra platform. To facilitate understanding of the improvements described in this paper, we present a brief overview of the platform (section 2.1) and how it was used during the pilot phase for processing DNA sequencing data (section 2.2). Finally we identify and discuss in section 2.3 the various problems faced in the production phase due to the data growth, which motivate the improvements described in section 3.

2.1 e-BioInfra platform: short introduction

The e-BioInfra platform provides generic services for executing and monitoring data analysis workflows on a distributed computing infrastructure (DCI). A detailed description of the platform can be found in Olabarriaga *et al.* [6] and Shahand *et al.* [13].

The workflow components are implemented using the Generic Application Service Wrapper (GASW) [14]. Command-line data analysis tools (executables and dependencies) and their in- and output parameters are described in an XML description file that is used by the workflow system to automatically generate grid jobs. The components are combined in workflows using the graphical user interface of MOTEUR [15]. Originally the SCUFL language [16] was used, but currently the workflows are defined in the GWENDIA language [17]. This is a rich, data-oriented language with sophisticated constructions such as multi-dimensional arrays, condition, loop, merge and filter operations. In addition, embedded Java code can be incorporated into the workflow using

[4]http://picard.sourceforge.net/

`beanshells` and executed on the workflow engine host. The workflow description also defines how data and parameter sweeps should be performed on the input parameters. Workflow descriptions are shared between bioinformaticians and biomedical researchers by making them accessible on the Virtual Organization's (VO) shared storage space on the LCG File Catalogue (LFC).

When a workflow is instantiated with input data and parameters, the MOTEUR workflow management system coordinates data and parameter sweeps and translates the workflow components into jobs that run on grid resources or as local `beanshell` executions. In 2008 the gLite WMS[5] was used by MOTEUR for direct job submission, but currently the DIANE[6] pilot framework [18] is used instead. The user who analyses the data can choose to run an existing workflow with a command-line client, desktop client (VBrowser with MOTEUR plug-in) or web-based interface (e-BioInfra gateway) [13].

The progress of the workflow runs can be examined from the e-BioInfra monitoring pages, which provide status updates and error information when jobs or workflows fail. Improvements to the e-BioInfra monitoring have been made to detect errors and distinguish their type [19]. Initially, information was stored in unstructured log files, therefore it was difficult to track events that occur in several layers of the platform. Currently, events are collected from every level of the workflow execution, stored in a provenance database, and presented in a user interface.

2.2 Pilot phase

The general approach used for development of workflows for DNA experiments is described in detail in Luyf *et al.* [7].

During the pilot phase most actions were operated via the Virtual Resource Browser, or VBrowser[7] [20]. The VBrowser is a Java program that provides the user with a graphical user interface and explorer look-and-feel to manipulate files on various types of storage resources (FTP, gridFTP, LFC). For example, it was used to transfer the data from the sequencing server, where the data became available after acquisition, to grid storage, where it was processed by the workflows. It was also used to fetch results from grid storage to a local analysis server for further processing. The VBrowser was used from the user's laptop, where the user could select files from the sequencing server and copy them to grid storage with a drag-and-drop action. By default only one replica is created on grid storage for each transferred file.

The VBrowser also has a plug-in to submit MOTEUR workflows. Bioinformaticians operated the workflow execution and shared the analysis results with biomedical researchers via a data analysis server located at the AMC. Workflows were submitted interactively by selecting a workflow file from a central repository, after which the input parameters could be filled in one-by-one via a form. Alternatively an XML file could be generated by the user and uploaded to avoid typing many input parameters interactively. At the time the data size per experiment did not exceed 1 GB, and the frequency of new sequence experiments was around one every three or four weeks.

2.3 Production phase

Since the successful pilot experiments in 2010, the e-BioInfra has been increasingly adopted

[5] http://web.infn.it/gLiteWMS/

[6] http://cern.ch/diane

[7] http://www.vl-e.nl/vbrowser

for DNA sequencing data analysis. Between January 2011 and February 2012 alone, a total of 9600 workflows were executed, including production data analysis, workflow development and troubleshooting. With the increasing adoption of the workflows in production, we faced various problems due to the increasing number and size of the files containing data and analysis results. The most important problems can be categorized as follows:

Data transfer to grid storage Originally the VBrowser was executed on the user's own machine to transfer data from the sequencing server to grid storage. However, when the VBrowser runs on the laptop of the user, the files are first transferred to the laptop and then to the target resource, which means that the data is transferred twice and the network connection of the user becomes the limiting factor. The VBrowser is also capable of doing third-party transfers, so that the users laptop would not become the bottleneck, but the transfer protocol of the sequencing server does not support third-party transfers. This is not a problem for smaller files, but for large datasets with file sizes up to 70 GB the data transfer takes too long to complete.

Sharing of results After the analysis the results are made available to the biomedical researcher for further analysis. In the production phase the number of sequence experiments increased, as well as the size of the individual files, and there were more users to share the results with. Biomedical researchers usually do not posses a grid certificate, with a few exceptions. Moreover, not all researchers are keen on installing grid-enabled utilities, such as the VBrowser, on their laptops. Therefore we had to find alternatives to more easily share large result files without requiring all researchers to obtain grid credentials or install additional clients on their workstations.

Data access bottlenecks We observed data transfer timeouts when many jobs were submitted at the same time. Some files are used by all jobs, such as the reference database for sequence alignment (e.g. Human Genome) and the workflow component executables. Since by default only one replica was created, too many jobs were trying to access the same file and therefore created a bottleneck.

Insufficient resources on the worker node Another cause of job failures was related to insufficient disk space for temporary files and crashes due to lack of memory. The sequence files became larger and used up all local disk space for temporary files. Memory shortage occurred since jobs run on shared hardware and in some cases need more memory than normally can be assigned to a job. In the Dutch grid a job typically runs on a worker node with a multicore CPU using shared memory. We observed that memory-heavy jobs either compete for memory leading to significant degrading performance, "out-of-memory" exceptions or even get killed by a resource monitoring system.

Long run time of jobs The processing time for computations such as sequence alignment increases linearly with the input data size. So with the growth of data size, the processing time also increased and sometimes took longer than the time slot available on the worker nodes, thus leading to jobs getting aborted.

Workflow submission and monitoring for large numbers of files The MOTEUR workflow engine has a built-in retry mechanism for failed jobs, but frequently workflows were only partly

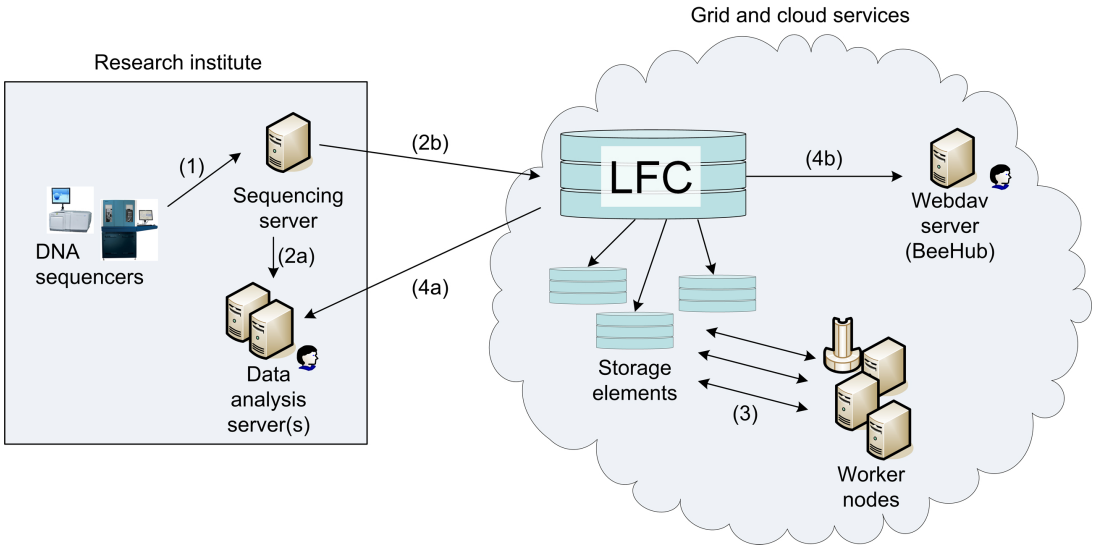

Figure 1: Data flow from data generation to sharing of results. The DNA sequencer stores sequence data on a sequencing server (1), from where it is transferred to a data analysis server (2a) and/or to grid storage (2b). During workflow execution data is transferred to/from worker nodes (3). The result files are transferred to a data analysis server (4a) or WebDAV server (4b) from where they can be accessed by biomedical researchers.

finished, even after retries. Job failures occurred because of too large input files, too memory-intensive jobs, application failures and grid related errors, e.g. data transfer errors. As the run time of analysis steps is relatively long (in the order of 2-24 hours per step), it is important to be able to resume a workflow run, skipping datasets that already have been analyzed. This functionality is not offered by MOTEUR.

3. Improvements

After the pilot phase we have implemented several improvements to deal with the problems identified in section 2.3: to facilitate data transfer to/from grid storage (section 3.1); handling of results (section 3.2); to reduce data bottlenecks (section 3.3); to split the work into smaller chunks (section 3.4); and to manage more files and fault handling in workflow submission section 3.5).

3.1 Data transfer to and from grid storage

Fig. 1 illustrates the set-up that is currently adopted for data transfers. To avoid an overload of the sequencing server, only a very limited number of people have access to it. Downstream data analysis takes place on a separate data analysis server (Fig. 1, 2a). The biomedical researchers interact with the data analysis servers.

Initially the transfer between the servers and grid storage was operated remotely from the user's laptop. Currently the VBrowser is installed on the sequencing server and the data analysis servers, enabling direct transfer between these servers and grid storage (Fig. 1, 2b, 4a). The VBrowser is an application that runs in user space, so no administrator rights are needed to install the software, and its installation does not have a big impact on the server.

In addition to avoiding double transfer to/from the user's laptop, this approach is more efficient because the sequencing server has a faster network connection to the outside world than a regular laptop on the slower intranet or wifi connection.

3.2 Handling of results

When the analysis is completed, results are transferred to a data analysis server and that can be accessed locally by users who do not have a grid certificate. This was initially done using the VBrowser from a laptop, but currently either the graphical user interface or the command-line tools of the VBrowser from the data analysis server (Fig. 1, 4a) are used.

Recently, we explored WebDAV for making the results available for collaborators (Fig. 1, 4b) using a system[8] deployed at BiG Grid[9]. We publish the URLs, which point to the result files. The benefits of using this system are easy file sharing and access from other programs running on the user's workstation. For example, the URLs can be used by a Genome Viewer application used by the researchers to view and further analyse alignment results. Moreover, this server can be reached from outside the AMC firewall, which allows data to be shared with external collaborators to our institute.

3.3 File replication

The grid files are manually replicated with the VBrowser or with the gLite command-line tools to avoid bottlenecks and to prevent failure when a particular storage element is unreachable. Files that are needed by many jobs, e.g. the human genome reference database [21], are replicated on each storage site of the Dutch e-science grid where the data analysis will take place to reduce data transfer time during the analysis. At the time of writing the Dutch e-science grid had storage resources located at 15 sites. During workflow execution the nearest storage site is chosen automatically by the GASW wrapper.

3.4 Adaptation of workflows for larger datasets

Since we observed several job failures due to the large data size of sequencing experiments, we decided to reduce the processing chunks by splitting the sequencing data.

At first we focused on the most common task in DNA sequence analysis, which is the alignment of the DNA sequences to a reference database. In this task the data is processed sequentially and independently, so parallelization can be achieved by splitting the data [22]. Data and reference database splitting has been successfully done for various sequence alignment programs, such as BLAST [23] and BFAST [24]. Here we use the Burrows-Wheeler Transform Alignment program (BWA) [9] as an example of our strategy, but this approach can be applied to several other types of tools as well.

BWA builds an index of the reference database and stores it in memory. The sequences of the experiment are then read from disk and aligned to the reference. The run time for aligning the sequences to the genome scales almost linearly with the amount of sequences (and file size). Since

[8] http://www.beehub.nl/

[9] http://www.biggrid.nl/

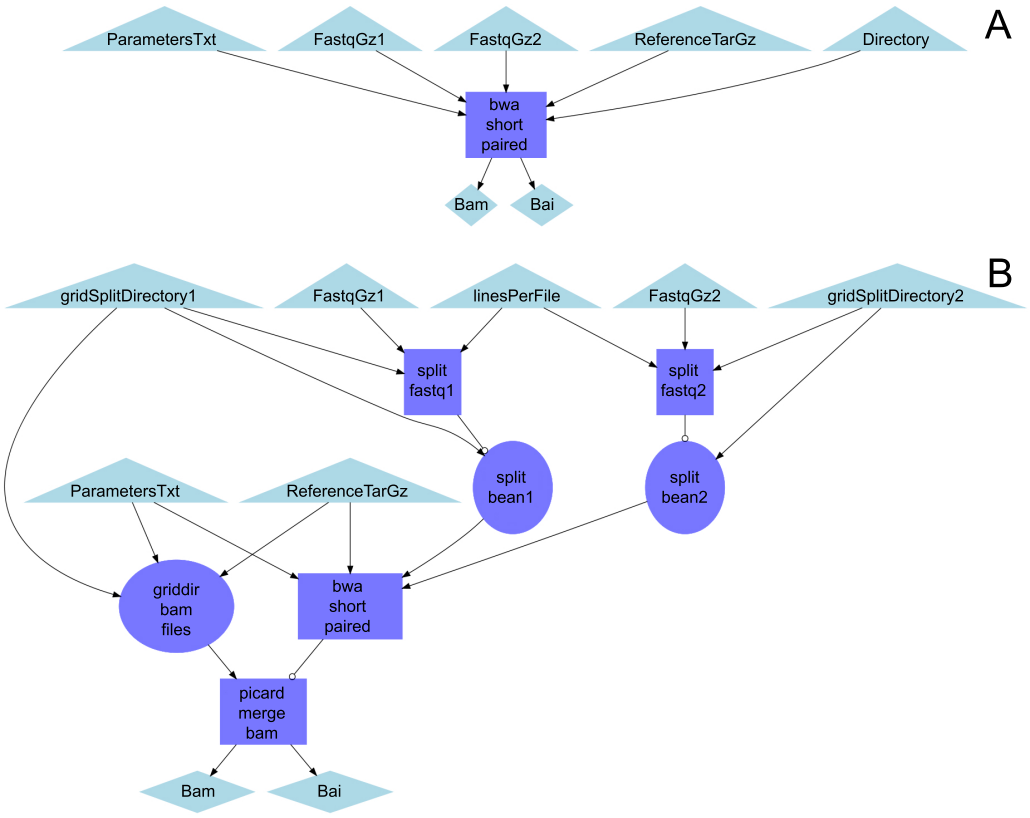

Figure 2: BWA workflows. A) Original workflow where the complete dataset is processed by one component. B) Modified workflow where the input dataset is split, processed in parallel, and the results are merged.

the sequences are aligned independently to the reference, they can processed in parallel without changing the analysis outcome.

The original workflow is depicted in Fig. 2A. It contains one component for BWA and takes a complete dataset as input (`FastqGz1` and `FastqGz2`) to align against a reference database (`ReferenceTarGz`) with the BWA parameters supplied via `ParametersTxt`. The directory for final results is specified via the `Directory` parameter.

The adapted workflow implementing the split-and-merge strategy is depicted in Fig. 2B. It has an additional parameter where the size of chunks can be defined in terms of lines per file (`linesPerFile`). A workflow component was developed for splitting the dataset (components`split-fastq1,2`), after which all smaller datasets are processed in parallel (`bwa-short-paired-reads`). When all the alignments are ready, the results are merged by the `picard-merge-bam` workflow component.

The split and merge components themselves need to process the complete dataset, demanding unusual amounts of memory or local disk space. For these type of analyses with large datasets we currently make a (manual) selection of resources with sufficient free disk space for the job. The target resource to run these components is defined in the workflow component description (GASW descriptor), being used by DIANE to create the job. Currently there are no automatic

means to select the site based on the amount of free disk because the resources do not publish this type of information. The selection is currently made based on personal communication with grid administrators.

3.5 Workflow submission and fault handling for numerous datasets

Workflows are submitted interactively in the e-BioInfra platform. However this becomes inconvenient when too many files need to be indicated. We have automated the workflow submission by 1) listing the names of the files and parameters to be processed (comma-separated file); 2) converting this list into valid XML input descriptors for the MOTEUR web service; and 3) submitting the workflows using a command-line interface utility.

For one specific experiment a database was used to keep track of input files and the expected output files generated by successful workflow execution. Output files are written in subdirectories at the location where the input files are stored using well-defined rules, so one knows beforehand which output files to expect by the execution of each workflow component on each input. A tool (`workflow-results-verifier`) was developed to check the results generated on the LFC, storing in the database the successful ones. After this, a new list of inputs was generated for the missing files by comparing the expected and the generated output file names. An input XML file for MOTEUR was generated with this information, and a new workflow was started with this subset of inputs. When steps further down the workflow failed, another workflow was started without the earlier succeeded steps.

This process could be automated by executing the `workflow-results-verifier` tool via, e.g., a cron job. However, to avoid endless resubmissions in case of permanent failures, the tool was started manually after inspection of an earlier workflow run. Note that this strategy was meant to reduce the manual steps needed to verify the output generated by the workflow, and resulted in a considerable reduction of time to start and resume partially failed workflows. The described solution was specific for one experiment, but such functionality would be interesting for the execution of large workflows that process many inputs.

4. Results

In this section we demonstrate how the improvements presented in section 3.3 and section 3.4 affected the performance of workflow execution. Controlled experiments were performed where the BWA workflow was executed repetitively using various settings.

The experiments used a dataset where the exome was sequenced and which represents about 1% of the complete human genome. These types of experiments are conducted to identify mutations involved in rare disorders and are regularly performed in our institute. We chose a recent representative dataset. The input dataset for the workflow consists of two compressed files containing the sequence fragments and base quality scores (3.5 and 4.8 GB). The human reference genome is a compressed archive containing the genome sequence and index files for the BWA program (5.2 GB).

Three different workflows were executed twice per day on this dataset between 6 and 20 July 2012 under different conditions. In total we have performed 180 workflow runs, i.e. 30 repeated runs with six different settings to explore three types of variations:

Table 1: Results for workflow executions under varying settings: experiment identifier (Id; see text), dataset (complete or split), number of replicas of the reference database, site selection for the split-merge process, number of successful/failed workflows, success rate (successful/total), and workflow duration (time to complete): mean, standard deviation, minimum and maximum.

Id	Dataset	Replicas	Site selection	# (ok/failed)	Success (%)	Mean $\pm$ Std.Dev. (hrs)	Min, Max (hrs)
A	complete	one		3/27	10.0	55.3 ± 28.2	35.1, 95.3
B	complete	multiple		6/24	20.0	55.5 ± 15.0	34.1, 69.7
C	split	one	no	14/16	46.7	19.1 ± 10.2	9.7, 54.3
D	split	one	yes	17/13	56.7	24.0 ± 10.8	12.2, 51.4
E	split	multiple	no	18/12	60.0	18.5 ± 10.6	8.1, 58.6
F	split	multiple	yes	21/9	70.0	22.7 ± 11.1	14.3, 59.0

- complete or split dataset, where in the second case the dataset was split into 10 smaller files and processed in parallel;
- with single or thirteen replicas of the reference database (replicas on each site); and
- with or without site selection to run the split and merge components. The site was manually selected based on the available space for temporary files.

We measured workflow execution time of the successful workflows as the elapsed time from the submission of the first job to the completion of the last one, including data transfer of result files. We assessed the success rate as the percentage of successfull workflows compared to the total number of workflows.

Three different workflow implementations were used in the experiments, see also Fig. 2. The first is the original BWA workflow with one component that processes the input data as a complete set (table 1, runs A and B). The second workflow splits the input data and runs every job on a random site (table 1, runs C and E). The third workflow is essentially the same as the second, except that the split and merge steps are configured to run on a selected site (table 1, runs D and F) by defining a Job Requirement on the GASW descriptor. Each workflow was executed twice; once using a single replica reference database (table 1, runs A, C and D) and once with the reference database replicated on every site of the Dutch e-science grid (table 1, runs B, E and F).

With these experiments we were able to assess the effect on workflow execution time and success rate of the following strategies: (a) splitting the input data; (b) replication of the reference database and with replicas on all available sites; and (c) site selection for the split and merge steps.

Table 1 summarizes the results. Splitting the input dataset results in workflows that run faster: from 55 hours on average (runs A and B) to an average of 20 hours for the other runs. More importantly, a much larger success rate is observed for the runs using split datasets: from 10 and 20% for runs A and B, up to 46% and 60% for runs C and E. When a particular site is selected for the split and merge steps, the total run time of the workflow is slightly longer on average (from 18.5 to 24 hours), possibly because it takes longer to obtain the selected resource. However the success rate is clearly higher, from 46% and 60% in runs C and E to 56% and 70% for D and F. Therefore the proposed improvements enabled us to go from a situation where only 10% of the workflows

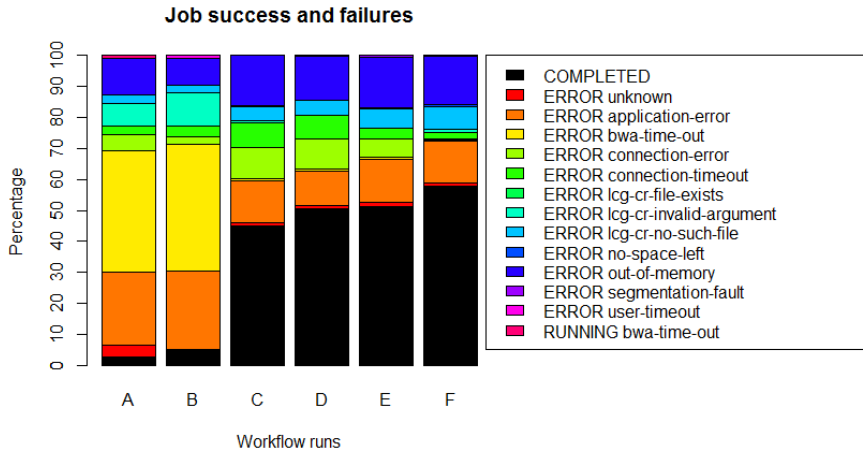

Figure 3: Completed and failed jobs in BWA workflows for runs A-F, showing error distribution. See also table 2.

ran successfully with an average run time of 55.3 hours, to an average of 22 hours with a success rate of 70%.

To get insight in the type and amount of errors, we have filtered the standard error log files for a list of terms that were frequently observed in error messages. Errors were due to application faults (*application-error, segmentation-fault*), timeouts (*bwa-time-out, connection-time-out, user-time-out*), problems with connection to systems (*connection-error*), related to data transfers back to grid storage (*lcg-cr-**), full disk (*no-space-left*), insufficient memory (*out-of-memory*). In some cases the error could not be identified (*unknown*). The distribution of error types in the various runs are shown in Fig. 3 and presented in more detail in table 2. For BWA workflows where the complete dataset was processed as a whole the most observed errors were job timeouts. Although the BWA workflows where the alignment was parallelized perform much better, there were still several failing jobs. The most observed reason for failure were out-of-memory issues. Fig. 3 clearly shows that the percentage of errors decreased drastically after implementing the improved workflow.

5. Discussion

The e-BioInfra platform provides generic services, such as workflow submission, monitoring, and provenance services, which are used on a daily basis via easy-to-use interfaces by a growing number of users. At the end of 2008 we started a pilot for using this infrastructure for DNA sequence analysis. Since then around 40 studies have been performed using the e-BioInfra platform and around 80 workflows have been developed for sequence analysis alone. The number of implemented workflow components from existing and newly developed software for DNA sequence analysis has grown and the components are now used by bioinformatics and biomedical researchers. New discoveries were done in these areas and are verified with complementary laboratory experiments, such as virus discovery [25], (partial) human genome re-sequencing [26], whole transcriptome sequencing [27] and small RNA expression profiling [28].

Table 2: Job status and observed error types in BWA workflows for runs A-F: number of jobs (percentage of jobs)

Job status	A	B	C	D	E	F
COMPLETED	3 (2.7)	6 (5.2)	258 (44.9)	264 (50.7)	259 (51.4)	295 (58.0)
ERROR unknown	4 (3.6)		6 (1.0)	4 (0.8)	7 (1.4)	4 (0.8)
ERROR application-error	26 (23.6)	29 (25.2)	79 (13.7)	58 (11.1)	69 (13.7)	69 (13.6)
ERROR bwa-time-out	43 (39.1)	47 (40.9)	4 (0.7)	5 (1.0)	4 (0.8)	3 (0.6)
ERROR connection-error	6 (5.5)	3 (2.6)	57 (9.9)	50 (9.6)	30 (6.0)	1 (0.2)
ERROR connection-timeout	3 (2.7)	4 (3.5)	46 (8.0)	39 (7.5)	17 (3.4)	11 (2.2)
ERROR lcg-cr-file-exists						
ERROR lcg-cr-invalid-argument	8 (7.3)	12 (10.4)	5 (0.9)	1 (0.2)		4 (0.8)
ERROR lcg-cr-no-such-file	3 (2.7)	3 (2.6)	25 (4.3)	24 (4.6)	32 (6.3)	37 (7.3)
ERROR no-space-left			1 (0.2)		1 (0.2)	4 (0.8)
ERROR out-of-memory	13 (11.8)	10 (8.7)	94 (16.3)	75 (14.4)	82 (16.3)	79 (15.5)
ERROR segmentation-fault				1 (0.2)	3 (0.6)	2 (0.4)
ERROR user-timeout		1 (0.9)				
RUNNING bwa-time-out	1 (0.9)					

We observed that the setup used by the pilot did not scale well with the increasing amount of data files and data size, so we have implemented various improvements to address challenges. These include a procedure for data transfer to and from grid, modifications to a workflow for the most common task in sequence analysis and implementation of tools to manage larger datasets.

5.1 Data transfer and results sharing

Although the data is now more rapidly transferred, the procedure still includes manual steps like selecting files to transfer and performing the data transfer. Ideally, the data transfer to and from distributed resources should be completely automated and there are several options to accomplish this. For example, the data can be automatically pushed to grid storage, such as done by the e-BioInfra platform for workflows that are provided via the web portal [13]. In the portal users upload their data via sFTP to a data staging server which is then synchronized with grid storage. DNA sequencing datasets are too large to upload from the user's laptop, but in a similar way the data could be uploaded directly from the sequencing server to the grid storage. Another possibility is to coordinate the data transfer in the workflow itself by addition of data transfer components before and after the analysis steps.

5.2 Data splitting

So far the split-and-merge approach has only been implemented for the most commonly used analysis step: sequence alignment with BWA. This resulted in a higher success rate of the workflow executions and speed-up for the complete analysis of a dataset. We have not yet investigated whether further gain can be obtained by optimizing the split size of the data. We hypothesize that the split size should be determined dynamically, depending on the availability of resources and their workload.

Not all errors were solved by splitting the data, and several jobs still fail due to insufficient memory. Next to splitting the experiment dataset, there are possibilities for splitting the reference

database, but extra processing is needed to merge results. For example, the human genome reference database can be split for processing multiple chromosomes in parallel. However when the human genome is split, information gets lost about sequences that align to multiple chromosomes, and this needs to be recovered or corrected in the merge procedure.

We have noticed in the experiments of section 4 that there was one site where jobs failed more often than on others (Additional file 1). Excluding these sites could reduce the number of failures even more.

Other analysis steps after the alignment can be parallelized as well, to improve on run time and success rate, as long as these steps operate on genomic intervals (e.g. per chromosome) and do not need the complete dataset. Examples are sequence realignment [12], marking duplicate reads[10], base score recalibration [12] and variant calling [11, 12]. In these cases the files could be split per chromosome, processed in parallel, and merged afterwards using similar workflow components as for the alignment workflow.

5.3 Automatic workflow submission and monitoring

A drawback of the implemented tool to resume failed workflows is that there are still manual steps involved, which slow down the analysis. First the user who analyzes the data needs to wait until the workflow is finished, then check if all analysis results are complete, and manually start new workflow(s) for the missing result files. These steps can now be performed with one command that checks what is complete and what is still missing, but human intervention is still needed. Moreover, the implemented solution is specific for a given experiment, therefore we are investigating ways to generalize it. One of the alternatives is to implement a resume functionality at the workflow level. At the moment the workflows are being extended to implement a fault recovery mechanism, which has already been successfully implemented in other workflows for medical imaging. At the end of a workflow a component will check whether all output files were produced, and then restart the analysis for unprocessed files. The workflow will iterate until all results are complete or until a given maximum number of iterations is reached.

6. Related work

Various bioinformatics tools, including tools for DNA sequence analysis, have been implemented, and sometimes optimized for grid infrastructures. We discuss below three selected examples from directly related work, but many others exist.

Aparicio and co-workers [29] implemented BLAST for metagenomics experiments on the EGEE grid within the Biomed virtual organization. They implemented customized tools for job submission, job monitoring, and for re-submission of failed jobs. They point out that careful planning is needed to select suitable resources, distribute data, and create replicas, to make optimal use of the available resources. This is in accordance with our experience described in this paper.

Mirto *et al.* have optimized BLAST on grid by splitting experiment data as well as the reference database. They have modified the BLAST source code to account for recalculation of the statistics of alignments and avoid post processing at the merging phase [23].

[10] http://picard.sourceforge.net/

Kim and colleagues have investigated how BFAST could be optimized with the DARE framework on grid, cloud and local infrastructures [24]. Since the execution time scales almost linearly with the amount of sequences, speed-up can be achieved by splitting the experiment sequences. Splitting the reference database (splitting the genome per chromosome) only showed a two-fold speed-up, but still has advantages because the disk space and memory used per job is reduced. BWA has similarities to BFAST, therefore it is worth to investigate whether we can increase the success rate by splitting the reference database in addition to the experiment sequences.

In contrast to the three examples above, we have made improvements on the workflow level without changing the application itself. This has been our approach because we focussed on porting many different tools as workflow components to create complete pipelines for various sequence analysis experiments, without touching the original code of these tools. For other applications it might be necessary to change the code, but since applications are updated very frequent (often multiple updates per year) it will be time-consuming to synchronize all updates.

In addition to distributed execution of specific tools, also several generic platforms and tools have been developed and used to integrate and run sequence data analysis pipelines on distributed resources, some of which are mentioned below.

A notable example is Galaxy [30], which presents bioinformatics tools in a uniform user interface and can be installed on a local cluster or accessed via a public server. Galaxy does not support grid infrastructures, but several Cloud implementations of Galaxy exist now. Although pipelines can be constructed in Galaxy, it has limited possibilities for parameter sweeps. Another popular platform for bioinformatics is Taverna [16], however this does not have facilities for distributing computing.

MOTEUR, on the other hand, supports grid infrastructures and offers rich constructs to perform data and parameter sweeps without additional programming. Other comparable workflow systems that intrinsically support computing on distributed systems are Pegasus [31] and WS-PGRADE [32] (see Deelman *et al.* for a review [33]). We use MOTEUR in the e-BioInfra platform because it is a mature environment, that is used in production at other institutes, and because we have a close working relationship with the developers.

Ferreira da Silva and colleagues [34] use a comparable approach to ours to run large scale computations on the EGI infrastructure. Although the application area is mostly medical imaging, their framework is similar, and also based on MOTEUR. They recently performed a thorough error analysis and implemented mechanisms that can blacklist sites when jobs show an increased error rate on these systems and resubmit failed jobs. It would be interesting to explore how these or similar solutions could be integrated in the e-BioInfra platform.

7. Conclusion

The characteristics of implemented tools for DNA sequence analysis on the e-BioInfra platform vary between tools with short and long run times, few or several dependencies, low or high memory usage, and different sizes of the input data. Especially the larger datasets raised challenges for data transfer to and from the grid, as well as successfully executing alignments and performing other analysis steps. A procedure to transfer data from the DNA sequencer to grid storage is now in place, which speeds up the data transfer. We have also invested in a procedure to share the results

with researchers that do not have a grid certificate, and to keep track of all analysis steps for a particular experiment. The run times of individual jobs are relatively long (in the order of hours), which can cause jobs to fail due to wall clock timeouts.

The BWA application is used the most in our institute for DNA sequence analysis and we have made improvements on the workflow level without changing the code of the application. It is important to reduce failures to avoid an increase of the total run time of a workflow due to resubmission of failed jobs. We have implemented a split-and-merge procedure for the most common task (sequence alignment) to decrease the total run time of a workflow and to reduce failures due to the large file size. The procedure resulted in a 2-3 fold speed-up for the analysis of an exome sequencing experiment and the success rate of the workflows increased from 10% to 70%. Steps are taken to be able to resume workflows from the point where jobs failed.

Although the improvements already resulted in increased performance both in time-to-complete and success rate, further improvements are still required, such as automatic data transfer before and after a workflow run, optimization of the split size for paralellization, paralellization of other tools, and a generic approach for fault recovery.

Since the developments in next generation sequencing are progressing fast, we use generic solutions to keep up with the field. We have to search for a balance between including new applications in our analysis workflows and optimization of the most frequently used ones. Improvements and optimization of current strategies, from data generation till interpretation of results, are needed to prepare for the many and larger DNA experiments that will be performed in the near future.

Acknowledgments

We thank the members of the e-bioscience team who contribute to the development, operation and support of the e-BioInfra. The sequence data used in the reported experiments was provided by the laboratory division of the Academic Medical Center (Frank Baas, Raoul Hennekam). We thank Kyriacos Neocleous for feedback on the manuscript. This work is financially supported by BiG Grid and SHIWA EU FP7. The work uses BiG Grid resources, the Dutch e-Science Grid, which is financially supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organisation for Scientific Research, NWO).

References

[1] J. McPherson, *Next-generation gap*, Nature methods, 2009, vol. 6(11s), pp. S2-S5.

[2] M. Baker, *Next-generation sequencing: adjusting to data overload*, Nature methods, 2010, vol. 7(7), pp. 495-499.

[3] M. Hsi-Yang Fritz, R. Leinonen, G. Cochrane and E. Birney, *Efficient storage of high throughput DNA sequencing data using reference-based compression*, Genome research, 2011, vol. 21(5), pp. 734-740.

[4] L. Stein, *The case for cloud computing in genome informatics*, Genome Biology, 2010, vol. 11(5), 207.

[5] E. Schadt, M. Linderman, J. Sorenson, L. Lee, G. Nolan, *Computational solutions to large-scale data management and analysis*, Nature Review Genetics, 2010, vol. 11(9), pp. 647-657.

[6] S. Olabarriaga, T. Glatard and P. de Boer, *A Virtual Laboratory for Medical Image Analysis*, IEEE Transactions on Information Technology In Biomedicine (TITB), 2010, vol. 14(4), 979-985.

[7] A. Luyf, B. van Schaik, M. de Vries, F. Baas, A. van Kampen, S. Olabarriaga, *Initial steps towards a production platform for DNA sequence analysis on the grid*, BMC Bioinformatics, 2010, vol. 11, 598.

[8] S. Coelho, *Hunting for viruses: finding a needle in a haystack*, ISGTW, January 2012.

[9] H. Li and R. Durbin, *Fast and accurate short read alignment with Burrows-Wheeler Transform*, Bioinformatics, 2009, vol. 25, pp. 1754-1760.

[10] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin and 1000 Genome Project Data Processing Subgroup, *The Sequence alignment/map (SAM) format and SAMtools*, Bioinformatics, 2009, vol. 25, pp. 2078-2079.

[11] D. Koboldt, K. Chen, T. Wylie, D. Larson, M. McLellan, E. Mardis, G. Weinstock, R. Wilson and L. Ding, *VarScan: variant detection in massively parallel sequencing of individual and pooled samples*, Bioinformatics, 2009, vol. 25(17), pp. 2283-2285.

[12] M. DePristo, E. Banks, R. Poplin, K. Garimella, J. Maguire, C. Hartl, A. Philippakis, G. del Angel, M. Rivas, M. Hanna, A. McKenna, T. Fennell, A. Kernytsky, A. Sivachenko, K. Cibulskis, S. Gabriel, D. Altshuler and M. Daly, *A framework for variation discovery and genotyping using next-generation DNA sequencing data*, Nature Genetics, 2011, vol. 43(5), pp. 491-498.

[13] S. Shahand, M. Santcroos, Y. Mohammed, V. Korkhov, A. Luyf, A. van Kampen and S. Olabarriaga, *Front-ends to Biomedical Data Analysis on Grids*, Proceedings of HealthGrid, June 2011, Bristol. UK.

[14] T. Glatard, J. Montagnat, D. Emsellem and D. Lingrand, *A Service-Oriented Architecture enabling dynamic services grouping for optimizing distributed workflows execution*, Future Generation Computer Systems, 2008, vol. 24(7), pp. 720-730.

[15] T. Glatard, J. Montagnat, D. Lingrand and X. Pennec, *Flexible and Efficient Workflow Deployment of Data-Intensive Applications On Grids With MOTEUR*, International Journal of High Performance Computing Applications, 2008, vol. 22(3), pp. 347-360.

[16] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. Pocock, A. Wipat and P. Li, *Taverna: A tool for the composition and enactment of bioinformatics workflows*, Bioinformatics, 2004, vol. 20(17), pp. 3045-3054.

[17] J. Montagnat, B. Isnard, T. Glatard, K. Maheshwari and M. Fornarino, *A data-driven workflow language for grids based on array programming principles*, WORKS '09: Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, 2009.

[18] J. Moscicki, M. Lamanna, M. Bubak and P. Sloot, *Processing moldable tasks on the grid: Late job binding with lightweight user-level overlay*, Future Generation Computer Systems, 2011, vol. 27(6), pp. 725-736.

[19] S. Madougou, M. Santcroos, A. Benabdelkader, B. van Schaik, S. Shahand, V. Korkhov, A. van Kampen and S. Olabarriaga, *Provenance for distributed biomedical workflow execution*, Proceedings of HealthGrid, June 2012, Amsterdam, NL.

[20] S. Olabarriaga, P. de Boer, K. Maheshwari, A. Belloum, J. Snel, A. Nederveen, M. Bouwhuis, *Virtual Lab for fMRI: Bridging the Usability Gap*, Proceedings of Second IEEE International Conference on e-Science and Grid Computing, 2006.

[21] International Human Genome Sequencing Consortium, *Initial sequencing and analysis of the human genome*, Nature, 2001, vol. 409, pp. 860-921.

[22] Y. Mohammed, S. Shahand, V. Korkhov, A. Luyf, B. van Schaik, M. Caan, A. van Kampen, M. Palmblad and S. Olabarriaga, *Data Decomposition in Biomedical e-Science Applications*, IEEE eScience conference Workshop on Computing Advances in Life Sciences (CALS), 2011, Stockholm, Sweden.

[23] M. Mirto, S. Fiore, I. Epicoco, M. Cafaro, S. Mocavero, E. Blasi and G. Aloisio, *A bioinformatics grid alignment toolkit*, Future Generation Computer Systems, 2008, vol. 24, pp. 752-762.

[24] J. Kim, S. Maddineni and S. Jha, *Characterizing Deep Sequencing Analytics Using BFAST: Towards a Scalable Distributed Architecture for Next-Generation Sequencing Data*, ACM Proceedings of the second international workshop on Emerging computational methods for the life sciences (ECMLS), 2011.

[25] M. de Vries, M. Deijs, M. Canuti, B. van Schaik, N. Faria, M. van de Garde, L. Jachimowski, M. Jebbink, M. Jakobs, A. Luyf, F. Coenjaerts, E. Claas, R. Molenkamp, S. Koekkoek, C. Lammens, F. Leus, H. Goossens, M. Ieven, F. Baas and L. van der Hoek, *A sensitive assay for virus discovery in respiratory clinical samples*, PLoS One, 2011, vol. 24 6(1), e16118.

[26] J. Van Houdt, B. Nowakowska, S. Sousa, B. van Schaik, E. Seuntjens, N. Avonce, A. Sifrim, O. Abdul-Rahman, M-J. van den Boogaard, A. Bottani, M. Castori, V. Cormier-Daire, M. Deardorff, I. Filges, A. Fryer, J-P. Fryns, S. Gana, L. Garavelli, G. Gillessen-Kaesbach, B. Hall, D. Horn, D. Huylebroeck, J. Klapecki, M. Krajewska-Walasek, A. Kuechler, M. Lines, S. Maas, K. Macdermot, S. McKee, A. Magee, S. de Man, Y. Moreau, F. Morice-Picard, E. Obersztyn, J. Pilch, E. Rosser, N. Shannon, I. Stolte-Dijkstra, P. Van Dijck, C. Vilain, A. Vogels, E. Wakeling, D. Wieczorek, L. Wilson, O. Zuffardi, A. van Kampen, K. Devriendt, R. Hennekam and J. Vermeesch, *Heterozygous missense mutations in SMARCA2 cause Nicolaides-Baraitser syndrome*, Nature Genetics, 2012, vol. 44(4), pp. 445-449

[27] R. Huis In 't Veld, M. Willemsen, A. van Kampen, T. Bradley, F. Baas, Y. Pannekoek and A. van der Ende, *Deep Sequencing Whole Transcriptome Exploration of the s Regulon in Neisseria meningitidis*, PLoS One, 2011, vol. 6(12), e29002.

[28] N. Schopman, M. Willemsen, Y. Liu, T. Bradley, A. van Kampen, F. Baas, B. Berkhout and J. Haasnoot, *Deep sequencing of virus-infected cells reveals HIV-encoded small RNAs*, Nucleic Acids Research, 2011, vol. 40(1), pp. 414-427.

[29] G. Aparicio, I. Blanquer, V. Hernandez, *A Highly Optimized Grid Deployment: the Metagenomic Analysis Example*, Proceedings of HealthGrid, Studies in Health Technology and Informatics, 2008, vol. 138.

[30] J. Goecks, A. Nekrutenko, J. Taylor and The Galaxy Team, *Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences*, Genome Biology, 2010, vol. 11, R86.

[31] S. Callaghan, E. Deelman, D. Gunter, G. Juve, P. Maechling, C. Brooks, K. Vahi, K. Milner, R. Graves, E. Field, D. Okaya, T. Jordan, *Scaling up workflow-based applications*, J Comput System Sci, 2010, vol. 76(6), pp. 428-446.

[32] P. Kacsuk, *P-GRADE portal family for Grid infrastructures*, Concurrency and Computation: Practice and Experience journal, 2011, vol. 23(3), pp. 235-245.

[33] E. Deelman, D. Gannon, M. Shields, I. Taylor, *Workflows and e-Science: An overview of workflow system features and capabilities*, Future Generation Computer Systems, 2009, vol. 25, pp. 528-540.

[34] R. Ferreira da Silva, T. Glatard, F. Desprez, *Self-Healing of Operational Workflow Incidents on Distributed Computing Infrastructures* 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2012) (pp. 318-325).