Master Thesis
Software Engineering
Thesis no: MSE-2010:06
April 2010

On using Desktop Grid Computing in software industry

Andreas Cederström

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Andreas Cederström
E-mail: andreas.cederstrom@gmail.com

External advisor:
Robert Reineck
Bitsec Consulting AB
Phone: +46 8 411 90 10

University advisor:
Kennet Henningsson
School of Engineering at Blekinge Institute of Technology

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 271 25

ABSTRACT

Context. When dealing with large data sets and heavy calculations the common solution is clusters, supercomputers or Grids of these two. However, there are ways of gaining large computational power by utilizing the unused cycles of regular home or office computers, this are referred to as Desktop Grids.
Objectives. In this study we review the current field of solutions for open source Desktop Grid computing capable of dealing with a heterogeneous set of clients and dynamic size of the Desktop Grid. We investigate current use, interest of use and priority of key attributes of Desktop Grids. Finally we want to show how time effective Desktop Grids are compared to execution on a single machine and in the process show effort needed to setup a Desktop Grid and start computing. The overall purpose of this study is to provide a path for industry organizations to take when taking the first step into Desktop Grid computing.
Methods. We use a systematic review to collect information of existing open source Desktop Grid solutions. Studies are selected based on inclusion criterions and a quality assessment. A survey questioner is used to assess industry usage, interest and prioritization of attributes of Desktop Grids. We will conduct an experiment to show execution speedup as well as setup effort.
Results. We found ten open source Desktop Grids fulfilling our requirements. The survey shows that Desktop Grids is used to a very little extent within industry while a majority of the participants state that there is an interest for Desktop Grids. As result of the experiment, we can say that we achieved very high speedup and that effort needed to setup a Desktop Grid is about 40 hours for one person with no prior experience to the selected Desktop Grid system.
Conclusions. We conclude that industry organizations have a possible need for Desktop Grids but in order to be more successful, Desktop Grid developers must put more effort into areas as automated testing and code compilation.

Keywords: Desktop Grid, distributed computing, BOINC.

Acknowledgments

I would like to thank my supervisor, Kennet Henningsson, for his guidance and help through this thesis. I would also like to thank Bitsec Consulting AB for giving me the opportunity to do this thesis with them. Last but not least, I would like to thank my girlfriend, family and friends for their support during this thesis. This thesis is dedicated to Bosse Ericsson, the best of friends.

Table of Contents

1 Introduction

1.1 Background

Today, a lot of large organizations are in need of high-throughput computing (HTC) to perform large scale calculations in order to understand our reality. These calculations can be anything from predicting earthquakes [1] or screening for new vaccines [2].

It is understandable for the research community to require solutions for calculations on large sets of data, but there is also a need in industry for such tools. For example, a software development company could benefit from having automated test suites running on several machines to produce performance and stability measurements.

The need to perform large scale calculations like the above mentioned in a simple and effective way has become one of the many challenges for the research community today [3]. But if the same need exists within industry, then this is something that must be looked in to.

A common way to manage large calculation problems is to use either supercomputers or clusters. However, in the past 20 years, the evolution of wide-area networking and major research on system distribution [4] has made it possible to build systems connecting computers worldwide to share resources such as data space and CPU power. These systems are referred to as *computational Grids* or just *Grids* [5]. In this report they will be referred to as Grids.

A Grid is defined as a solution for *"controlled and coordinated resource sharing and resource use in dynamic, scalable virtual organizations"* [5] by Foster (the "father" of Grid computing) et al. Grids has been used for many things in the past, from advanced networking to artificial intelligence. However, nowadays the research community has adopted the above stated definition. It is the sharing and use of resources that is the essential part in this definition, the ability to do something with the resources that otherwise would have gone to waste.

A lot of organizations have a large number of work-stations in use almost every day of the year. A high amount of their computational power goes to waste since they rarely use their full potential. Not only is this a waste of computational power, it is also a waste of electric power. This does not go well hand-in-hand with today's environmental thinking and the pursuit of "green" products.

One of the problems with utilizing recourses of an existing computer arsenal is the variety of architectures, platforms and operating systems. As stated by Foster et al [6], most Grid middleware focus only on a specific platform or operating system. This becomes a difficulty when you want to use all your workstations as one great computational unit.

The heterogeneity of computers in a Grid is indeed a problem of great magnitude to overcome, according to [7] many solutions to the problem exits such as various forms of object systems, Web technologies, problem-solving environments, the CORBA programming language, workflow systems, and compiler-based systems, but, we have yet no standards for how to deal with heterogeneous Grid computing.

With the technology of the Internet and a tool such as the ones listed above we have the possibility to use multiple computers as one unified computing resource [8]. It is now possible to connect regular workstations, laptops and supercomputers together into one large Grid and execute code as if it where run on a single machine.

Now, a new problem emerges. If we want to connect multiple workstations, or *clients*, in a single computational Grid, we are never sure of whether they are online or not. The need for a Grid that can handle both the heterogeneity and the dynamic size emerges. This need has also been identified by Karonis et al [7], Casanova [4] and Abramson et al [9]. Grids with these attributes are often referred to as *Desktop Grids*.

There are some commercial Desktop Grid solutions on the market today, like Entropia[32], Harmony[33] and Alchemi[34]. These systems utilize the power of regular office workstations as well as clusters. According to [30] they all run in isolation with little or no ability to communicate with other Grids and they most likely do not follow the Grid standards provided by Open Grid Services Architecture (OGSA)[35].

As far as we know, there have been no efforts in bringing open source Grids to industry. Open source software is often cheaper and more customizable than commercial products. We think that there is a market for open source Grid solutions in industry today, in this thesis we will try to clarify this.

1.2 Terminology

Grid computing is in this work the ability to use computers idle CPU time and memory to perform small operations from a greater context. This must not be confused with Data Grids which is a way of managing very large amounts of data dispersed over multiple computers, perhaps geographically spread. When we use the term Grid in this work we mean any kind of computational Grid, including Desktop Grids.

Desktop Grid is a Grid where the nodes mainly are regular desktop computers or laptops. The purpose of a Desktop Grid is therefore to scavenge the idle CPU cycles and memory from such machines. In this work the term Desktop Grid is used to point to the fact that the Grid targets desktop machines. In this work, when we are talking about Desktop Grids we imply that these are open source unless otherwise is stated.

Heterogeneity in this work means a blend of various operating systems and hardware architectures. An extension to this could also include heterogeneity of location, i.e. nodes in a Grid exist in different networks and can be widely spread across the Internet. Many articles talk about heterogeneity of data but that is not the case in this work.

Dynamic size of a Grid solution is the ability for nodes to join and leave the Grid as they wish during runtime without that other nodes is affected (other than the amount of work per node will change). This requires a high level of fault tolerance that must be implemented in the solution.

Grid of Grids is a Grid solution that connects existing Grids into a larger Grid. The Grids that is connected into the Grid of Grids can be built with different middleware.

Client is a machine which is connected to the Grid and shares its resources there. It can be anything from a desktop computer to a cluster. Other names for a client in related work can be *node*, *worker*, *worker node*, *grid machine* or *peer* just to mention a few. In this work we choose to call it a client.

Work unit is a job to be done on a Grid. A work unit consists of application code to be executed and input of some kind. A work unit produces a result which is mapped to the work unit.

1.3 Aims and objectives

This thesis project was triggered by the Stockholm based security company Bitsec Consulting AB when presenting the need to speed up the process of large scale calculations. They were looking for a solution that should be scalable, cheap and possible to distribute on their and their customers already existing arsenal of computers.

The aim of this research is to compile information of how heterogeneous and dynamical computational Desktop Grids is, or could be, used within industry. We want to find Desktop Grid solutions that meets the requirements formulated by Bitsec Consulting AB and show how one of these Desktop Grids could be set up. This Desktop Grid setup will show how much effort is needed to get a Desktop Grid up and running.

Bitsec Consulting AB mentioned that they want to be able to provide a Desktop Grid as a service for their customers. Given this we decided to do a survey on finding industry interest and areas of use. The survey will give indications on how much and to what a Desktop Grid can be used in industry.

In Table 1 you find the objectives that we strive to meet in this thesis. They are each mapped to a corresponding research questions to give a better understanding of this thesis.

Objective	Research question
Find a Desktop Grid middleware that supports heterogeneous and dynamical Desktop Grids.	RQ 1
Identify the interest and areas of use for computational Desktop Grids within industry.	RQ 2
Measure how time effective a Desktop Grid is compared to one-machine computing.	RQ 3

Table 1 Aims and objectives mapped to research questions.

1.4 Research questions

In Table 2 we have listed our main research questions (level one), their sub questions (level two) and the corresponding alternative questions (level three).

ID	Research question
RQ 1	Which open source Desktop Grid middleware is on the market today?
RQ 1.1	Is there any Grid middleware that supports a dynamic sized Grid?
RQ 1.1.1	If no on RQ 1.1; how could Desktop Grid middleware be modified to support this?
RQ 1.2	Is there any Desktop Grid middleware that supports a heterogeneous set of nodes?
RQ 1.2.1	If no on RQ 1.2; how could Desktop Grid middleware be modified to support this?
RQ 1.3	Is there any Desktop Grid middleware that supports both a dynamic sized Desktop Grid and a heterogeneous set of nodes?
RQ 2	Could a Desktop Grid middleware be used in industry?
RQ 2.1	Is this kind of Grid middleware of interest to use in industry?
RQ 2.2	Which areas of application could be covered by such a Grid middleware?
RQ 3	How time efficient is a small, heterogeneous and dynamic sized Desktop Grid?
RQ 3.1	How does execution time of a linear calculation differ between a Desktop Grid like this and a single machine?

Table 2 Research questions.

1.5 Expected outcomes

The work done in this thesis is expected to present:

- a summary of Grid technologies on the market today best suited to our research questions and the requirement of Bitsec Consulting AB,
- insight about how computational Grids is used and can be used in industry,
- data describing how calculations conducted on a computational Grid compares to calculations on a single machine in terms of time to finish jobs.

1.6 Research methodology

A mixture of qualitative and quantitative [11] research methodologies will be used to reach the goals of this thesis.

1.6.1 Qualitative methodology

RQ1 will be answered with a qualitative systematic literature review where we will identify which solutions exist today, how they are built and what purposes they have. We will follow the guidelines described by Kitchenham et al [10] for this work.

The systematic review is also where the work in this thesis is narrowed down in order to suite our other research questions. Since the area of Grid computing is rapidly evolving with the improvements of both software and hardware it is important to justify our topic and relay to relevant work. Reason for answering RQ1 with a systematic review is that it is a very thorough method and with the given input it will give a complete and objective overview of the topic.

Output of the systematic review will be based on the results of the quality assessment. The final selection of literature will be analyzed and compiled into a table where the discussed Grid solutions are classified.

1.6.2 Quantitative methodology

Both RQ2 and RQ3 will be answered using quantitative methods. To answer RQ2 we need insight into industry, we need to get data on how Grid computing is or could be used. As we see it, this data could be acquired in two ways: qualitative interviews or a quantitative survey. The survey approach takes less time and would thus generate more data than interviews given a fixed period of time.

We choose to answer RQ2 with a survey questionnaire which was sent out to people working with software engineering in industry. The questionnaire was design to provide data on the following questions regarding Grid computing:

- current usage?
- interest of use?
- priority of key attributes?
- areas of application?

RQ3 aims to first showing how time effective calculations can be on a Grid and second to show how much effort is needed to setup a Grid and build an application for it. Data showing how time effective a Grid is could have been found using a systematic review, interviews or any other methodology that grants a direct link to the results of such an experiment. However, in order to provide answer to both the time effectiveness and the setup effort we choose to conduct our own experiment. This gives us full control of all the input data which easily could have been overseen in for example an interview. Which Grid solution that will be used in this experiment will be decided through the systematic review.

For the experiment we will construct a test application to run on the Grid and measures execution time with various numbers of clients. The same application will then be executed on a single machine and the same measurement taken. This will allow us to compare execution time between a computational Grid and a single machine. Our experience from this experiment will provide data on how much effort is needed to setup a Grid and build an application for it.

1.7 Limitations

It would of course be good to cover all Grid solutions, both open source and commercial, in this study. Due to time and budget we do not have the resources to go into commercial solutions, instead we will focus on open source Grids in this work.

This thesis aims at bringing Desktop Grid computing into industry organizations. However, focus has been put on software industry at this stage since the initiator of this thesis, Bitsec Consulting AB, mainly deals with software related industry.

The experiment we conducted only focuses on one Desktop Grid middleware, BOIINC, which is a limitation. We cannot rule out that other solutions would have performed differently but we assume that they have similar performance.

1.8 Structure of this report

The structure of this report follows the same order as the research questions. From here on it is divided into four detailed sections:

1. systematic review,
2. survey,
3. experiment,
4. conclusions.

Each one of these sections contains purpose, method, results, discussion and validity threats.

2 Systematic review

In order to answer the first research question of this thesis we choose to conduct a systematic review. A systematic review is a powerful tool used to find proper literature on a topic. The methods used in this work are based on the framework created by Kitchenham et al [10].

A systematic reviews starts by defining research question, search terms, resources to be searched, quality assessment checklist and answers and a method for data extraction. The systematic review then serves as a tool to find as much relevant literature as possible with the given definitions as input.

A set of primary studies is the output of the literature search. This literature is then evaluated according to the quality assessment checklist. The quality assessment generates a kind of grade on each primary study.

2.1 Literature search

We started off with a broad search (iteration one) in order to test search terms and databases. Based on the results, we revisited the search terms and excluded the databases whit low relevance to the topic. The relevance was judged based on how well the articles topic fits the search term, some hits were articles just referring to Grid computing. High relevance was given when 60% or more of the hits regarded the search term, medium relevance when 20-60% regarded the search terms, medium relevance when less than 20% regarded the search terms and no relevance when none of the hits, if any, regarded the search terms.

This first iteration also gave us insight in the topic and it became clearer what to look for and what to not look for. With this in mind we created our inclusion criterions (Table 3) and exclusion criterions (Table 4). The search terms, databases and the results of iteration one can be found in Appendix A.

ID	Inclusion criteria	Clarification
IC1	Articles concerning Grid computing solutions and implementations	Research focused on developing or enhancing Grid middleware.
IC2	Articles concerning Grid issues and challenges	Research that aim to shed light on the difficulties and their solutions within Grid computing.
IC3	Articles concerning heterogeneous Grid solutions	See section 1.4 Terminology.
IC4	Articles concerning dynamic sized Grid solutions	See section 1.4 Terminology.
IC5	Articles concerning Grid experiments	Research concerned with experiments on how to improve Grid solutions.

Table 3 Inclusion criterions.

ID	Exclusion criteria	Clarification
EC1	Articles produced prior to January 1th 2004	Figure 1 shows the total number of hits published concerning Grid computing up until a given year in our five databases used for the systematic review. As you can see the curve beds a lot more at 2003, meaning that the number of articles published increased drastically in 2004.
EC2	Articles only concerning security solutions	E.g. an article that focus its contributions in the field of security within Grid computing. This is not relevant in this research.
EC3	Articles concerning so called Grid of Grids	Since a *Grid of Grids* is one level higher than a regular Grid, this is out of the scope of this research.
EC4	Articles concerning homogeneous Grid solutions	A Grid solution where the nodes have to be identical in terms of hardware and operating system.
EC5	Articles only concerning Grids computing in another research area	E.g. an article that focus its contributions on how to find a cure for malaria with the help of Grid computing. They might have specified what Grid solution the use but the research concerns an other topic than Grid computing.
EC6	Articles only concerning proprietary Grid computing solutions	A solution where it is not possible to gain access to source code and low level solutions without paying for it. We did not have a budget for this research and therefor an experiment with such a solution would have been impossible.
EC7	Articles only concerning high-level architectural solutions to Grid computing	These articles is often just thoughts and wishes of how a new type of Grid solution could look like. Again, it would not be possible to implement such a Grid for an experiment due to the time limits of this research.
EC8	Articles not written in English language	This is necessary for both the authors of this paper and its reviewers.
EC9	Articles that have not been subject to peer-review	Needed in order to have trust in our sources.
EC10	Articles only concerning comparisons (e.g. performance or usability) between Grid solutions	If an article only compares two exciting solutions without going into how they function it is not relevant to this research.
EC11	Articles mainly concerning application deployment	E.g. an article that focus its contribution on how to optimize the process of partitioning and deploying application code onto the Grid, not how the Grid is built.
EC12	Article that is not free to obtain through the library at Blekinge Institute of Technology	We did not have a budget for this research and therefor not the possibility to purchase articles.
EC13	Articles mainly concerning resource discovery, task scheduling or load balancing	These three are related to each other. Indeed, they are needed in a Grid for it to function at all but this research focuses on how an entire Gird solution is built and therefor this limitation must be made.

Table 4 Exclusion criterions.

One thing that became clear after iteration one was that the kind of Grid solution we were looking for is commonly referred to as a *Desktop Grid*. A Desktop Grid has the same purpose as a Grid but targets regular workstations or home computers.

With the set of search terms crystallized and the number of databases reduced from iteration one we did a new search, iteration two. At each search we applied our inclusion criterions on the first one hundred hits in order to find appropriate articles. The number of hits for each search term in iteration two can be found in Appendix B. The articles found during this search can be found in Appendix D.

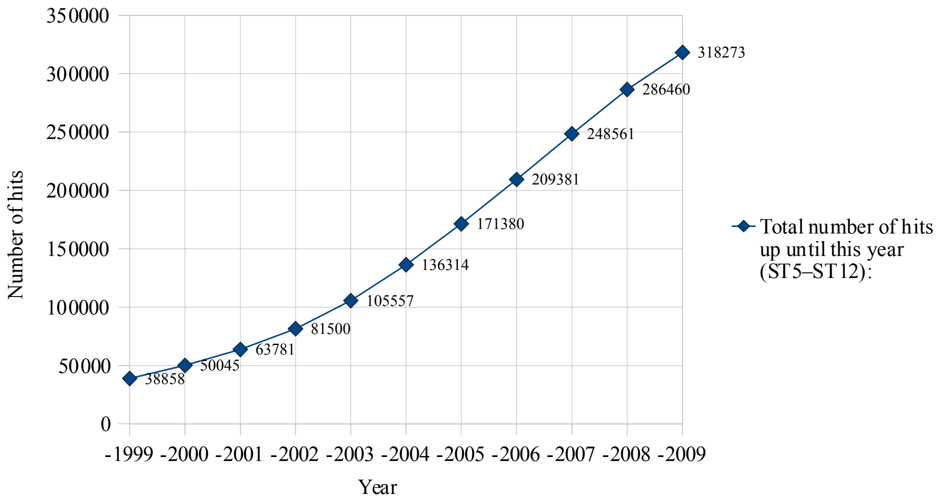

Figure 1 Growth in number of articles concerning Desktop Grid computing. (Definitions of ST5-ST12 can be found in Appendix B and Appendix C.)

2.2 Quality assessment

A test run of the quality assessment on ten random articles from the primary selection showed that it could be useful to do a third iteration of searching. New search terms was created and applied on the same databases as in iteration two, a list of search terms for iteration three can be found in Appendix C.

As a result of the test run we decided to add quality assessment question number four (QAQ4) since a lot of articles concerned Grid solutions that build upon existing open-source solutions, this without stating if the discussed solution itself was open-source or not.

We assessed each article from iteration two and three with the quality assessment questions (Table 5) and the exclusion criterions (Table 4). After processing the first one hundred articles from the initial selection we were already way past deadline for the systematic review. It turned out to be a very time consuming undertaking and we realized that a lot of these initial articles could have been excluded. Disregarding this, we did not want the quality assessment to become bias so we decided to proceed with the primary selection as it was regardless that the time plan had to suffer.

ID		Question / answer
QAQ1	Article concerns Grid computing solution(s)?	
	1.0:	Yes, article discusses solution(s).
	0.5:	Partly, article discusses part(s) of solution.
	0:	No.
QAQ2	Article concerns Grid computing solution(s) built for a dynamic sized set of nodes?	
	1.0:	Yes, the amount of nodes can vary during execution.
	0.5:	Partly, the amount of nodes can vary between executions.
	0:	No, the amount of nodes is fixed.
	0:	Not specified.
QAQ3	Article concerns Grid computing solution(s) built for a heterogeneous set of nodes?	
	1.0:	Yes, solution(s) or part(s) built to be neither hardware nor software specific.
	0.5:	Partly, solution(s) or part(s) built to be either hardware or software specific.
	0:	Not specified.
QAQ4	Article concerns open-source Grid computing solution(s)?	
	1.0:	Yes, solution(s) or part(s) is open-source.
	0.5:	Partly, solution(s) or part(s) is based on open-source components.
	0:	Not specified.
QAQ5	The Grid computing solution(s) or part(s) has been implemented and used?	
	1.0:	Yes, the solution(s) or part(s) has been used within industry or academia.
	0.5:	Partly, the solution(s) or part(s) has been implemented as a proof-of-concept.
	0:	No, the solution(s) or part(s) has not been implemented.

Table 5 Quality assessment questions and answers.

2.2.1 Assumptions of the Quality Assessment

The assumptions mentioned below has created to clarify things in the articles that was not explicitly stated. Whenever we were uncertain about any answer we first tried to find the correct answer on the Internet or in other articles but when this failed we had to apply these assumptions.

If the solution discusses in the article has been implemented and tested but it is not clearly stated if this has been done on a real set of computers or it has been simulated; we assume that it has been simulated.

If it just loosely stated that the solution is heterogeneous; we assume that this refers to heterogeneity of hardware, not operating system.

2.3 Results and discussion

Our primary selection from iteration two and three resulted in a total of 162 articles, 54 of these was rejected according to the exclusion criterions. From the quality assessment we had a document with all the articles ordered by their score from the assessment (see Appendix D).

A total of ten articles scored five (maximum) in the quality assessment, we decided to select these ten articles for our final selection. A closer look on the final selection revealed that the articles deal with eight different Grid solutions. Inspired by the work done in [13] we have categorized the Grid solutions discussed in these articles. The result can be found in Table 6.

The result of this systematic review will help organizations in selecting the Desktop Grid best suited for their organization. As far as we know this type of compilation has not been done for Desktop Grids earlier.

	Alice [14]	BOINC [15][16]	Globus+ BOINC [17]	GridBASE [18]	Ibis+Satis [19][20][21]	MyGrid [22]	OurGrid [23]	SZDG [13][24][25]
Architecture	Three-tier	Client/server	Client/server	Client/server	P2P	Client/server	P2P	Client/server
Target environment	Clusters	Home and office computers	Home and office computers + clusters	Home and office computers + clusters	Home and office computers + clusters	Home and office computers	Home and office computers	Home and office computers
Node distribution	Global	Global	Global	Global	Global	Global	Global	Global
Node connectivity	None	None	None	None	P2P	None	P2P	None
Security mechanism	Node info is undisclosed to others + node authenticatio n + data encryption + sandboxing	Application trust + signature based code verification + sandboxing	Application trust + signature based code verification + certificate identification + sandboxing	Node authenticatio n against database	Node authenticatio n + sandbox	Signature based code verification	Credibility based sabotage detection + sandboxing	Application trust + certificate based code verification + redundant task allocation
Task submission	Any node	Restricted by server	Restricted by server	Any node	Any node	Any node	Any node	Any node
Task distribution	Pull model	Pull model	Pull model	Pull model	Push model	Push or pull model	Push model	Pull model
End-user interface	Command line	Graphical	Graphical	Command line	Graphical	Graphical	Graphical or command line	Graphical (web portal)
Supported programing languages	Java, C, C++	C, C++ (with wrappers: FORTRAN, Java, Python)	C, C++	Any language	Any language supported in Java Virtual Machine	Java, .NET (Microsoft's . NET and Mono)	Any with support for SOAP	C, C++ (with wrappers: FORTRAN, Java, Python)
Supported operating systems	Any Java compatible OS (Java Virtual Machine is used)	Mac OS X, Windows, Linux and other Unix systems	Mac OS X, Windows, Linux and other Unix systems	Any (a task has specific code for specific operating systems)	Any Java compatible OS (Java Virtual Machine is used)	Mac OS X, Windows, Linux and other Unix systems	Any Java compatible OS	Mac OS X, Windows, Linux and other Unix systems
Implemented in	Java	C	C	Java, SQL	Java	C# + Java	Java	C

Table 6 Categorization of selected Desktop Grids.

2.4 Validity threats of Systematic Review

We choose to have a very inclusive way of selecting primary articles to evaluate in our quality assessment. We saved every article that passed the inclusion criterions to our reference management software without recording from which database or search term they originated. This way of working is indeed a difficulty in a complete reproduction of our work. However, if the same method is used it will result in the same final selection since all the primary articles was found in the databases listed in Appendix B and Appendix C.

The fact that the number of inclusion criterions is fewer than the exclusion criterions could be seen as an indicator to that the initial selection was too large. It would have been better to construct more inclusion criterions and thus reduce the amount of literature in the primary selection.

3 Survey

The systematic review of this thesis was conducted with such search criterions that we would be able to find articles concerning or leveraging the use of Desktop Grids in industry. However, no such reports were found. For this we see two possible explanations: (i) Desktop Grids is used in industry but we could not find any signs of this, (ii) Desktop Grids have little to none recognition in industry. This confirms the need for RQ2.1.

3.1 Survey design

This survey was designed using the method described by JW. Creswell in [12]. Using this method one must identify a purpose for the survey, determine its population and how to sample the population, choose and describe the instruments used and state how data will be extracted.

3.1.1 Survey purpose

We conducted a survey with the purpose to see what interest industry have for using Desktop Grids and to what extent it is actually used. The participants were also asked to prioritize some key factors of a Desktop Grid in order to see which factors is of highest importance if they were to start using Desktop Grid computing. These factors are all found in the description of Desktop Grids in the final set of articles from the systematic review.

In the last part of the survey the participants was asked to think of areas of implementation for a Desktop Grid within their organization. This information can help the Desktop Grid developers finding new target areas for development to satisfy industry needs. The questions from the survey can be found in Appendix E.

3.1.2 Survey population

The target population for this survey was people working in software industry, these persons are the ones that have the ability to see whether Desktop Grid computing is of interest or not and to what it could be used. Educational and ethnical background is not relevant to this survey.

The survey was spread to clients of Bitsec Consulting AB dealing with software development and through industry contacts of the authors of this thesis. Bitsec Consulting AB has most of their customers in banking and financing, government, energy industry and the Swedish department of defense. An e-mail with brief information about the survey as well as a link was sent out asking people to participate. This was the best method for this thesis given the short time span a thesis project stretches.

Due to the method used for spreading the questionnaire it is impossible to know how many it reached and therefore we have no data on how many of the receivers actually answered it.

Everyone who answered the questionnaire was anonymous and nothing but their answers and IP was stored. The IP's was stored to prevent the same person taking the questionnaire multiple times.

3.1.3 Survey instrument

The survey was conducted using a web questionnaire form provided by SurveyMonkey [36]. The design of the questions was tested on two external persons before released. Once the questionnaire was released it was open for response collecting for three weeks.

3.2 Results and discussion

It is hard to estimate how many that actually was asked to participate in this survey since it could have been forwarded by those we sent it to. We sent the questionnaire to approximately 100 persons, give or take 20 due to the loose restrictions of spreading the questionnaire. We received 24 responses on the questionnaire, all of who works with software development.

The results of the survey is divided into three parts, they are listed below under separate headlines. The detailed answers from the survey can be seen in Appendix E.

3.2.1 Current usage

From the systematic review we assumed that Desktop Grid computing is not widely used within industry. This is also the case when looking at the results from the survey. Only one participant say that his/her organization uses Desktop Grid computing and as many as twenty two say that their does not. These numbers confirms our assumption from the systematic review.

When the results were collected and analyzed we realized that it would have been useful to ask if the participants of the questionnaire use clusters or supercomputers today. If this question would have been included we could have drawn deeper conclusions about the need for Desktop Grid computing.

3.2.2 Usage interests

Out of the twenty three participants who did not answered that their organization use Desktop Grid computing, fourteen of them say that it would be of some interest to use it. They all work with software development in some way and therefore have a valid opinion in this survey.

The conclusions from these numbers is not as clear as the ones in the previous section but if we generalize it a bit, it is probably fair to say that fifty percent of software organizations today could have some use of Desktop Grid computing.

Given this, it is obvious that Desktop Grids have the potential to become something very useful as long as they can reach their end users.

3.2.3 Future directions

Now, the survey was constructed so it would tell us about what organizations need for them to take the step into using Desktop Grid computing. First we asked them to prioritize seven criterions of a Desktop Grid system in an order of importance. There are variables but the results can still give some indications on what is prioritized.

The highest priority for an organization willing to start using Desktop Grid computing is a *low operating cost*. This criterion is closely followed by that the system must have *low purchase price*.

It is hard to see from the data in the table provided as the result of this question so to make it a bit clearer we provide a figure (Figure 2) with one graph for each criterion. In the graphs we have included the linear regression for each attribute to show in which direction each data collection in heading. If the line climbs upwards, this means that the criterion is generally prioritized low. If the line drops downwards, this means that the criterion is in generally prioritized high. How fast a line changes is also of importance, a fast climbing line indicates that the described criterion is generally prioritized very low and vice versa for a fast dropping line. We have included the function for each line to make the values of the lines more obvious.

The last question of the survey concerned areas of use for Desktop Grids. We got a lot of results and managed to compile them to nine main areas as can be seen in the results in Appendix E. The top three areas are *automated test*, *compilation* and *heavy calculations*. These are all interesting areas and it would make it easier for Desktop

Grids to gain ground within industry if such solutions were provided.

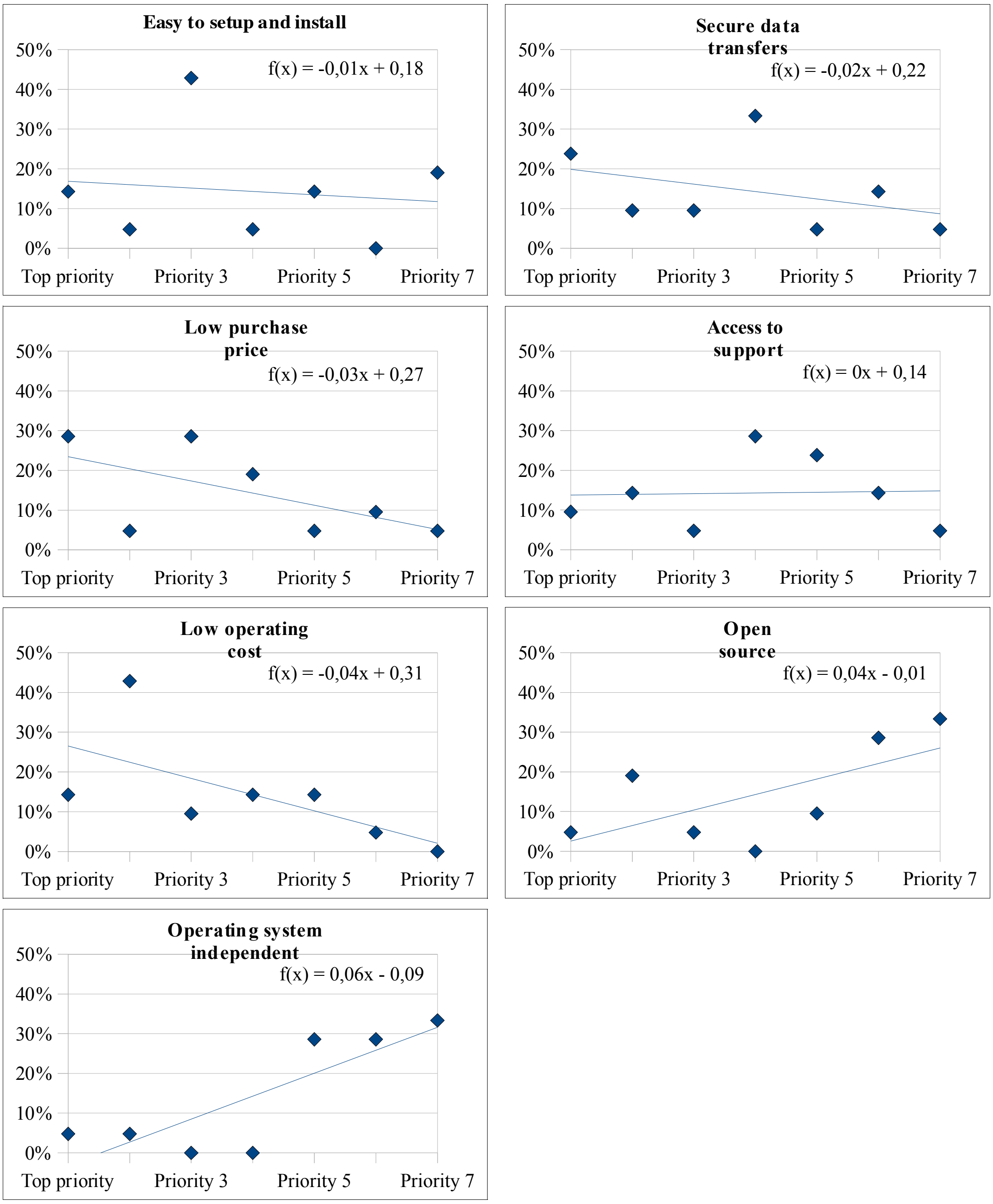

Figure 2 Regression lines for each criterion.

3.3 Validity threats of survey

First, a bad sample of participants could affect the results of the survey. We were very specific in the text where we asked people to participate in the questionnaire that they must be software engineering practitioners.

Second, a low amount of participants threatens the validity of statistical conclusion. This limits the value of the results.

Third, there was a threat that the participants would try to guess the purpose of the survey and thus answer accordingly without considering their organizations actual situation. To minimize this threat we explained the purpose of the survey in the beginning of the questionnaire.

The following three validity threats are general for most surveys and thus we choose to include them here.

Content validity threats refer to fully assess and understand the context of the survey. The questionnaire of this survey contained a brief description of Desktop Grid computing was formulated after the systematic review done in this work. This prevents us, the authors, and the survey population to have a faulty apprehension of the term Desktop Grid computing.

Face validity regards how the survey, questionnaire in this case, is presented in terms of language and language levels. The appearance of the questionnaire was tested on two external persons before sent out in order to find language issues.

Construct validity refers to the quality of the tools used for conducting the survey. In this work we used a web based questionnaire which by all means must be seen as a valid tool of conducting a survey.

4 Experiment

The third research question of this thesis aims at showing how time efficient a Desktop Grid can be compared to run calculations on a single machine. To do this we conducted an experiment with the purpose to show to you, the reader, how you can setup your own Desktop Grid and start using it within your organization.

We have shown in our systematic review that there are a lot of capable Desktop Grid solutions available today. Despite this, our survey clearly states that there is an interest for Desktop Grid computing but very few organizations use it.

4.1 Setup

We choose to conduct an experiment using BOINC[15] (Berkeley Open Infrastructure for Network Computing) and record our setup and results in order to show how much effort is needed to get a Desktop Grid server up and running and start doing computations on a small set of workstations.

The reason for choosing BOINC was mainly the large base of supported operating systems and programming languages. BOINC also use a client/server architecture which gives the owner more control over the server and applications. Another reason was that BOINC is widely used in research projects such as SETI@home[26], Einstein@home[27] and many more[28].

4.1.1 BOINC

BOINC is a product that sprung out of the SETI@home[26] project. BOINC was developed as an open infrastructure to reach millions of home computers to participate in computational projects that serves the general public. Users can participate in a BOINC project by installing a single client application and then selecting which project they want to donate their idle computational power to. BOINC was in 2007 the largest Desktop Grid system with more than 250.000 users participating in various hosted projects[30].

BOINC uses a pull model when assigning work to clients. This in combination with a deadline for each work unit gives BOINC a natural way of dealing with clients that becomes unreachable. If a client that is executing work goes down for some reason the server will wait until the deadline for the assigned work unit is reached before giving the work to another client.

The BOINC client has functionality for storing checkpoints of the execution of a work unit. This means that if the work unit for some reason is interrupted or killed in while in progress the client can resume work and finish the work unit anyway.

BOINC does not perform any matchmaking between work units and clients, instead the client has settings for how much CPU and memory it is allowed to use.

The client application for BOINC is open source and is available pre-built for the following operating systems:

- Windows 2000/XP/Vista/7 (32-bit),
- Windows XP/Vista/7 (64- bit),
- Mac OS X version 10.4.0+,
- Linux x86,
- Linux x64.

The developer of a BOINC project is responsible for providing the project with application code that can run on these different operating systems and their respective architecture.

4.1.2 Server

For the server setup we choose to use the virtual machine image that the BOINC team provides on their website. The image was then run on a MacBook with an Intel C2D processor at 2.0 GHz and 4 GB DD3 RAM at 1067 MHz with VMware Fusion 3.0.1. The virtual machine was allowed to use one processor core and 1 GB of RAM.

The virtual machine ran Debian 2.6.18 as operating system and BOINC server at build revision 20737 and the provided MySQL and Apache server.

We also downloaded the server source code, built and ran it on a machine identical to the ones described in section 4.1.3. The setup of this server took about two hours to setup from a fresh installation of the operating system. This shows how fast an inexperienced BOINC user can setup an own server. Due to the low amount of memory on our machines we did not use this server for the experiment but used the virtual machine instead.

4.1.3 Clients

All of our client machines were regular workstations that do not differ from one that could be found on a regular workplace in industry. All clients had Intel Pentium 4 processor running at 2.00 GHz and 512 MB DDR SDRAM at 266 MHz. They all ran Debian 5.0.4 as operating system and BOINC client 6.10.17 for Linux.

We had a total of six test machines at hand and they all had the exact same setup and configuration on them.

4.1.4 Networking

The server and the clients were connected over network with a HP Procurve 408 switch with 100 Mbit/s interfaces.

The server and clients had 100 Mbit/s networking interfaces directly connected to the switch.

4.1.5 Our application

With BOINC come scripts to build skeletal projects, including web page and database. It was very easy to build one of the provided test applications after reading up on the documentation of BOINC. Once that was done we could connect clients and test the configuration. Getting to know how BOINC works took us about 20 hours of reading documentation and exploring the functionality.

After getting to know the infrastructure we created our own project and application in C++. The application takes a file, read it line-by-line and performs an MD5 hash of each line 10 million times. This repetition has no actual purpose except using a lot of CPU operations. For each read line the application writes the hash to an output file and when all lines are processed the output is returned to the server.

The application mimics the nature of a common Desktop Grid application. On most available BOINC projects a work unit for the application consists of very little input and output but the calculations consume a large amount of CPU power.

Our application took about 20 hours to develop from scratch with no prior experience of developing for BOINC projects. The code was supported by and tested on Windows Vista (32-bit), Mac OS X version 10.6 and Debian 5.0.4. These were the operating systems we had pre installed and available when testing. Many other operating systems could have been supported if more time was spent on development.

The test data consisted of a total of 700 lines and was divided into 70 equally sized work units.

4.2 Execution

Since we had six machines at hand for the experiment we created six scenarios for execution.

Before using the BOINC infrastructure we ran the application in standalone mode on one of the clients. The input data was the 700 lines of string put into one input file. This showed us how long time our application would take on a regular machine without the overhead created by BOINC or networking. The result from this scenario serves as a baseline for the experiment.

Next, we included the BOINC infrastructure and the server to performed measurements on various numbers of clients. The clients are continuously, as they finish old work units, assigned new work units. These are downloaded to the client and then calculated and reported back to the server. The only setting we changed in the BOINC clients from the default values was the amount of CPU and memory they were allowed to use, we changed these values to 100%.

For each number of clients used in the experiment we did three runs and recorded the execution time.

4.3 Results

The collected data from the various executions of the experiment can be found in Table 7 and Figure 3. We used this data to calculate the speedup acquired for each number of used machines. Speedup tells us how much faster the application performs with N machines available for execution. Speedup (S_N) is calculated with

$$S_N = \frac{T_1}{T_N}$$

where S_N is the measured speedup in execution time compared to running in standalone mode, T_1 is the execution time for the application running on a single machine and T_N is the average execution time for the application on N machines.

We also used the measured values to calculate how much of the code that ran in parallel. Percentage of code runs in parallel (P_N) is, according to Amdahl's law [29], calculated with

$$P_N = \frac{\frac{1}{S_N} - 1}{\frac{1}{N} - 1} \quad .$$

	Standalone	BOINC				
Number of CPUs (N)	1	2	3	4	5	6
Measured time 1	9325	5366	3389	2599	2056	1798
Measured time 2	9879	5334	3356	2596	2070	1844
Measured time 3	10211	5325	3378	2586	2047	1820
Average execution time (T)	9805	5342	3374	2594	2058	1821
Speedup (S)		1,84	2,91	3,78	4,77	5,39

Table 7 Experiment results.

The average execution time is represented graphically in Figure 3. The power regression curve is was automatically generated by our word processing software and calculated with this formula

$$T(N) = 9946 * N^{-0,96} \quad .$$

This equation gives accurate result for low values of N but we do not know how accurate it is for values of N higher than six.

The same way execution time is predicted we can predict speedup, naturally the same restrictions on N applies here. Speedup is estimated with

$$S(N)=\frac{9325}{9946}*N^{-0,96} \quad .$$

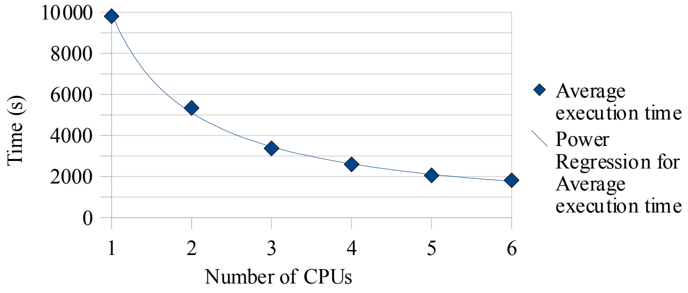

Figure 3 Measured execution time.

4.4 Discussion

The results suggest a high improvement in execution time of our application. The maximum speedup for N machines is N and as can be seen in Table 7 we collected data close to the maximum speedup.

The speedup is, according to Amdahl's law[29] dependent on how the application is built. If the application take 100 hours to run on a single machine and 5 of those cannot be parallelized the minimum execution time for the application, disregarding the number of machines dedicated to the application, cannot be lower than 5 hours i.e. a speedup of 20. This is represented in Figure 4 where we illustrate the maximum speedup when 50%, 90% and 95% of the application is parallelized.

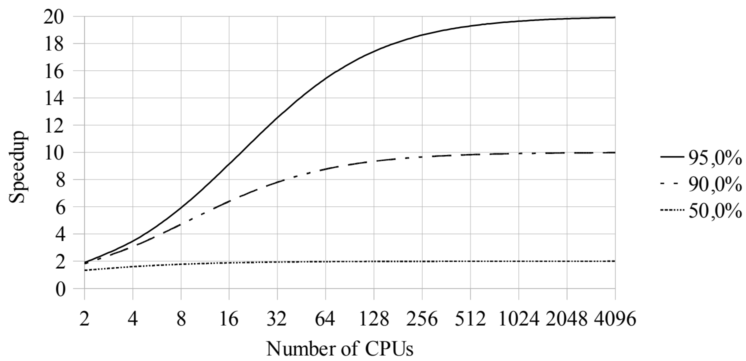

Figure 4 Maximum speedup in parallel computing according to Amdahl's law.

Experience from the setup and outcome of this experiment speaks for itself, it does not take much time to setup a running Desktop Grid with a simple application. Industry can defiantly benefit from using Desktop Grid computing. Most of all those that use or plan to use clusters or supercomputers. Instead of buying expensive new hardware, Desktop Grids offers the same power but without new hardware. This makes Desktop Grid computing ideal for smaller organizations that cannot afford large hardware costs or just has a temporary need for computational power.

4.5 Validity threats of experiment

There is a risk that the network used in the experiment to connect the clients to the server is overloaded and thus the results may be inaccurate. This was prevented by isolating the network so the only things on it were the clients and the server.

The selection of equipment for the experiment poses as a threat to validity. If the faulty hardware and/or software is selected the result may be inaccurate. To prevent this we carefully selected which Desktop Grid solution to use and thoroughly tested the hardware. The selection of BOINC was based on the initial requirements from Bitsec Consulting AB and the fact that BOINC was the largest Desktop Grid system in 2007 which indicates that it is well tested with real-world projects. Selecting another Desktop Grid solution would have given the same results as BOINC did.

5 Summary

In this section we will conclude the work done in this thesis and the results of it. We will then follow by discussing the given conclusions and finally end with some directions for future work.

5.1 Conclusion

This thesis started out with a systematic review where we collected literature regarding Grid computing and mainly Desktop Grid computing. We found a considerable amount of work done in this field only between 2004 and 2009. Desktop Girds is a relatively new term but the literature we found shows that a lot of existing solutions exist today.

We wanted to find a solution that would support heterogeneous and dynamically sized Desktop Grids, thus we created the search terms and the quality assessment questions accordingly. The result of the systematic review consisted of ten articles regarding eight different open source Desktop Grid solutions, all of them meet the requirements defined by Bitsec Consulting AB.

After the systematic review followed a survey with the aim to identify current usage, interest in usage, key attributes and areas of application for a Desktop Grid. The survey was conducted using a web questionnaire and samples people working with software engineering in industry. Due to the population, the survey only got 24 responses out of about 100, which is not enough to statistically prove anything but it serves as an indicator just as well.

The survey indicates that Desktop Grid computing is used to a very small extent in industry. At the same time the majority of the participants express interest in using Desktop Grid computing within their organization. As we have found no sign of work on bringing open source Desktop Grid computing into the world of industry, we see the reason to why only one of the participants state that his/her organization uses Desktop Grid computing.

The survey also generated a list of high prioritized attributes of a Desktop Grid in order for an organization to start utilizing this technology. The highest prioritized attributes is, in this order, *a low operating costs*, *low purchase price and secure data transfers*. In our opinion, these attributes must be fulfilled in order for Desktop Grids to successfully make its way into the modern organization. The second highest prioritized attribute, *a low purchase price*, is automatically meet with an open source Desktop Grid solution while we can assume that the other two are the same for commercial and open source solutions.

Some areas of use for a Desktop Grid were identified from the survey. The top three areas are *automated test*, *compilation* and *heavy calculations*. These are all interesting areas and it would make it easier for Desktop Grids to gain ground within industry if such functionality were provided or leveraged.

The third and last part of this thesis was an experiment which aimed to show how fast a Desktop Grid is compared to a single machine and also to show the effort needed to setup a Desktop Grid and build an application for it. We choose to setup the Desktop Grid with BOINC, one of the solutions found in the final selection of literature after the systematic review.

Setting up a BOINC server can be done in two hours for an inexperienced person with the documentation at hand. Getting to know the basic functionality of BOINC and build our test application took about 40 hours with no prior experience of developing for BOINC.

The application we developed was tested on a single machine without the BOINC infrastructure and then run on the Desktop Grid with various number of clients connected to the server. The execution speedup in this test can be estimated with

$$S(N)=\frac{9325}{9946}*N^{-0,96}$$

where N is the number of clients connected to the server. This formula is based on hardware and software specifications provided in section 4.1.

Our experience from setting up a BOINC server and building an application shows that little effort is needed in order to gain a large amount of computational power.

Finally, the work done in this thesis shows that Desktop Grids exists in various forms and is readily available for industry to utilize it. Desktop Grids can serve as a replacement for expensive clusters or supercomputers and is therefore an adequate alternative to these solutions. Open source Desktop Grids has a lower initial cost than clusters and supercomputers since no new hardware is needed to get started.

5.2 Discussion

The impact of Desktop Grid computing in an organization has the potential to be great. If a need for large computational power exists Desktop Grid computing is a better alternative in terms of initial costs for software and hardware compared to clusters or supercomputers. The selection of Desktop Grids instead of more traditional ways of gaining more computational power could be a great benefit to the organization.

The cheer computational power gained is one thing, but other aspects exist as well. One of the main differences between a Desktop Grid and let's say a cluster, is that the upgrading of hardware is done continuously in a Desktop Grid while it has to be done with a lot more effort and costs with a cluster. It is also a much more environmental friendly alternative since Desktop Grids utilizes the power that would have gone to waste anyway.

We think that Desktop Grid computing could change the way organizations deal with large scale calculations and computational tasks once it gains more ground in industry.

5.3 Future work

As the survey conducted in this work shows, there are some key attributes and areas of use the Grid developers could focus on to gain more ground within industry. Future work could go in to more detail on how Desktop Gird software can evolve to better serve industry purposes such as automated testing and release building.

Another direction to go with future work would be to compare performance and cost between Desktop Grids and clusters or supercomputers. This would show the actual business value of using Desktop Grid compared to more conventional models. Such work is needed for industry to really get their eyes opened for Desktop Grid computing.

As a final direction of future work we suggest comparing costs of using open source Desktop Grids compared to commercial ones. This work could also include other pros and cons with the two. Such work would certainly serve industry organizations when they want to take the leap and start using Desktop Grid computing.

6 References

[1] Callaghan et al. “Reducing Time-to-Solution Using Distributed High-Throughput Mega-Workflows: Experiences from SCEC CyberShake”. 4th IEEE International Conference on e-Science (eScience'08), to appear (2008).

[2] Jacq et al. “Grid-enabled virtual screening against malaria”. Journal of Grid Computing (2008) vol. 6 (1) pp. 29-43.

[3] Altintas et al. “A framework for the design and reuse of grid workflows”. Lecture Notes in Computer Science (2005) vol. 3458 pp. 120-133.

[4] Casanova. “Distributed computing research issues in grid computing”. ACM SIGAct News (2002) vol. 33 (3) pp. 50-70.

[5] Foster et al. “The anatomy of the grid: Enabling scalable virtual organizations”. International Journal of High Performance Computing Applications (2001) vol. 15 (3) pp. 200.

[6] Foster et al. “Grid services for distributed system integration”. Computer (2002) vol. 35 (6) pp. 37-46.

[7] Karonis et al. “MPICH-G2: A Grid-enabled implementation of the Message Passing Interface”. J Parallel Distr Com (2003) vol. 63 (5) pp. 551-563.

[8] Baker et al. “The Grid: International efforts in global computing”. Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (2000).

[9] Abramson et al. “A computational economy for grid computing and its implementation in the Nimrod-G resource broker”. Future Generation Computer Systems (2002) vol. 18 (8) pp. 1061-1074.

[10] Kitchenham B and Charters S. Guidelines for performing Systematic Literature Reviews in Software Engineering (2007) (EBSE-2007-01).

[11] W. Dawson, “Projects in Computing and Information Systems: A Student's Guide”. Pearsson Education Limited (2009).

[12] JW. Creswell. “Research Design: Qualitative, Quantitative, and Mixed Methods Approaches”. Sage Publications (CA) (2008).

[13] Kacsuk et al. SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System. Journal of Grid Computing (2009) pp. 1-23.

[14] Teo and Wang. Alice: A scalable runtime infrastructure for high performance grid computing. Lecture Notes in Computer Science (2004) pp. 101-109.

[15] Anderson. BOINC: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (2004) pp. 10.

[16] http://boinc.berkeley.edu/trac/wiki/ProjectMain
Visited on 2009-12-14.

[17] Myers et al. Expanding the reach of Grid computing: Combining Globus-and BOINC-based systems. Grid Computing for Bioinformatics and Computational Biology (2007) pp. 71.

[18] Sterck et al. Database-Driven Grid Computing with GridBASE. Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on (2007) vol. 1.

[19] Bal et al. Ibis: Real-world problem solving using real-world grids. Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing-Volume 00 (2009) pp. 1-8.

[20] van Nieuwpoort et al. Satin: Simple and efficient java-based grid programming. Scalable Computing: Practice and Experience (2005) vol. 6 (3) pp. 19-32.

[21] http://www.cs.vu.nl/ibis/
Visited on 2009-12-22.

[22] Costa et al. MyGrid–A complete solution for running Bag-of-Tasks Applications. Proc. of the SBRC 2004–Salao de Ferramentas (22nd Brazilian Symposium on Computer Networks–III Special Tools Session) (2004).
[23] Andrade et al. Peer-to-peer grid computing with the OurGrid Community. Proceedings of the SBRC (2005) pp. 1-8.
[24] Kacsuk et al. Scalable desktop grid system. Lecture Notes in Computer Science (2007) vol. 4395 pp. 27.
[25] http://www.desktopgrid.hu/
Visited on 2009-12-22.
[26] Anderson et al. SETI@home: an experiment in public-resource computing. Communications of the ACM (2002) vol. 45 (11).
[27] Abbot et al. Einstein@Home search for periodic gravitational waves in early S5 LIGO data. ArXiv (2009).
[28] http://boinc.berkeley.edu/projects.php
Visited on 2010-03-07.
[29] Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. AFIPS Conference Proceedings (30) (1967): pp. 483–485.
[30] Balaton et al. SZTAKI desktop grid: a modular and scalable way of building large computing grids (2007).
[31] Naik et al. Harmony: A Desktop Grid for Delivering Enterprise Computations. GRID '03: Proceedings of the 4th International Workshop on Grid Computing (2003).
[32] Chien A et al. Entropia: architecture and performance of an enterprise desktop grid system. Journal of Parallel and Distributed Computing (2003) vol. 63 (5) pp. 597-610.
[33] Naik et al. Harmony: A Desktop Grid for Delivering Enterprise Computations. GRID '03: Proceedings of the 4th International Workshop on Grid Computing (2003).
[34] Luther et al. Alchemi: A .NET-based enterprise grid computing system. Proceedings of the 6th International Conference on Internet Computing (ICOMP'05) (2005) pp. 27-30.
[35] I. Foster, C. Kesselman, J.M. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Internet, 2002. http://www.globus.org/research/papers/ogsa.pdf.
[36] http://www.surveymonkey.com
Visited on 2010-03-23.
[37] S. Litwin. How to measure survey reliability and validity. SAGE Publications, Inc. 1995.

Appendix A Systematic review, iteration one

ID	Search term	Years covered	Score	Action
ST1	"distributed computing" AND heterogeneous	1999 – 2009	70	Include
ST2	"distributed computing" AND scalable	1999 – 2009	95	Include
ST3	"distributed computing" AND "dynamic size"	1999 – 2009	5	Exclude
ST4	"open source" AND "distributed computing" AND heterogeneous	1999 – 2009	35	Exclude
ST5	"open source" AND "distributed computing" AND scalable	1999 – 2009	75	Include
ST6	"open source" AND "distributed computing" AND "dynamic size"	1999 – 2009	5	Exclude
ST7	"open source" AND "distributed computing" AND heterogeneous AND scalable	1999 – 2009	70	Include
ST8	"open source" AND "distributed computing" AND heterogeneous AND "dynamic size"	1999 – 2009	5	Exclude
ST9	grid AND heterogeneous	1999 – 2009	70	Include
ST10	grid AND scalable	1999 – 2009	80	Include
ST11	grid AND "dynamic size"	1999 – 2009	5	Exclude
ST12	"open source" AND grid AND heterogeneous	1999 – 2009	135	Include
ST13	"open source" AND grid AND scalable	1999 – 2009	40	Exclude
ST14	"open source" AND grid AND "dynamic size"	1999 – 2009	0	Exclude
ST15	"open source" AND grid AND heterogeneous AND scalable	1999 – 2009	95	Include
ST16	"open source" AND grid AND heterogeneous AND "dynamic size"	1999 – 2009	0	Exclude
ST17	"grid computing" AND heterogeneous	1999 – 2009	65	Include
ST18	"grid computing" AND scalable	1999 – 2009	130	Include
ST19	"grid computing" AND "dynamic size"	1999 – 2009	5	Exclude
ST20	"open source" AND "grid computing" AND heterogeneous	1999 – 2009	100	Include
ST21	"open source" AND "grid computing" AND scalable	1999 – 2009	75	Include
ST22	"open source" AND "grid computing" AND "dynamic size"	1999 – 2009	5	Exclude
ST23	"open source" AND "grid computing" AND heterogeneous AND scalable	1999 – 2009	85	Include
ST24	"open source" AND "grid computing" AND heterogeneous AND "dynamic size"	1999 – 2009	5	Exclude

Rating

Relevance	Score
High	50
Medium	10
Low	5
None	0

Database score limits

Action	Score
Included	>=120
Excluded	<120

Search term score limits

Action	Score
Included	>=50
Excluded	<50

	ACM Digital Library http://portal.acm.org			ArXiv http://arxiv.org		
	Date of search	# hits	Relevance	Date of search	# hits	Relevance
ST1	2009-10-21	2 105	0	2009-10-21	377	10
ST2	2009-10-21	3 272	0	2009-10-21	446	50
ST3	2009-10-21	7	0	2009-10-21	4	0
ST4	2009-10-21	284	0	2009-10-21	80	10
ST5	2009-10-21	414	0	2009-10-21	77	50
ST6	2009-10-21	0	0	2009-10-21	3	0
ST7	2009-10-21	156	0	2009-10-21	48	50
ST8	2009-10-21	0	0	2009-10-21	0	0
ST9	2009-10-21	2 601	10	2009-10-21	475	10
ST10	2009-10-21	3 915	10	2009-10-21	527	10
ST11	2009-10-21	4	0	2009-10-21	6	0
ST12	2009-10-21	411	0	2009-10-21	90	10
ST13	2009-10-21	517	0	2009-10-21	100	10
ST14	2009-10-21	0	0	2009-10-21	4	0
ST15	2009-10-21	201	10	2009-10-21	56	10
ST16	2009-10-21	0	0	2009-10-21	0	0
ST17	2009-10-21	843	5	2009-10-21	193	10
ST18	2009-10-21	879	5	2009-10-21	309	10
ST19	2009-10-21	2	0	2009-10-21	6	0
ST20	2009-10-21	165	0	2009-10-21	70	10
ST21	2009-10-21	181	10	2009-10-21	64	10
ST22	2009-10-21	0	0	2009-10-21	4	0
ST23	2009-10-21	92	0	2009-10-21	46	10
ST24	2009-10-21	0	0	2009-10-21	0	0
		16049	**50**		**2985**	**270**
			Excluded			**Included**

	CiteSeer http://citeseerx.ist.psu.edu			Compendex + Inspec http://www.engineeringvillage2.org		
	Date of search	# hits	Relevance	Date of search	# hits	Relevance
ST1	2009-10-23	2 828	10	2009-10-22	1 919	10
ST2	2009-10-23	2 956	5	2009-10-22	2 082	5
ST3	2009-10-23	28	0	2009-10-22	2	0
ST4	2009-10-23	251	0	2009-10-22	15	5
ST5	2009-10-23	260	5	2009-10-22	13	5
ST6	2009-10-23	3	5	2009-10-22	0	0
ST7	2009-10-23	125	10	2009-10-22	0	0
ST8	2009-10-23	3	5	2009-10-22	0	0
ST9	2009-10-23	4 467	10	2009-10-22	5 502	5
ST10	2009-10-23	5 187	10	2009-10-22	4 531	10
ST11	2009-10-23	34	5	2009-10-22	1	0
ST12	2009-10-23	384	5	2009-10-22	42	0
ST13	2009-10-23	431	5	2009-10-22	47	0
ST14	2009-10-23	4	0	2009-10-22	0	0
ST15	2009-10-23	183	10	2009-10-22	7	0
ST16	2009-10-23	4	0	2009-10-22	0	0
ST17	2009-10-23	1 320	5	2009-10-22	2 830	5
ST18	2009-10-23	1 162	5	2009-10-22	2 078	10
ST19	2009-10-23	2	5	2009-10-22	1	0
ST20	2009-10-23	178	5	2009-10-22	29	0
ST21	2009-10-23	158	5	2009-10-22	21	0
ST22	2009-10-23	1	5	2009-10-22	0	0
ST23	2009-10-23	95	5	2009-10-22	4	0
ST24	2009-10-23	1	5	2009-10-22	0	0
		20065	**125**		**19124**	**55**
			Included			**Excluded**

	IEEE Xplore http://ieeexplore.ieee.org/			ISI Web of knowledge http://apps.isiknowledge.com		
	Date of search	# hits	Relevance	Date of search	# hits	Relevance
ST1	2009-10-22	518	5	2009-10-22	481	0
ST2	2009-10-22	391	0	2009-10-22	233	0
ST3	2009-10-22	1	0	2009-10-22	1	0
ST4	2009-10-22	2	0	2009-10-22	4	5
ST5	2009-10-22	0	0	2009-10-22	3	0
ST6	2009-10-22	0	0	2009-10-22	0	0
ST7	2009-10-22	0	0	2009-10-22	0	0
ST8	2009-10-22	0	0	2009-10-22	0	0
ST9	2009-10-22	1 124	5	2009-10-22	2 247	5
ST10	2009-10-22	628	10	2009-10-22	883	0
ST11	2009-10-22	0	0	2009-10-22	0	0
ST12	2009-10-22	6	50	2009-10-22	14	5
ST13	2009-10-22	3	10	2009-10-22	4	0
ST14	2009-10-22	0	0	2009-10-22	0	0
ST15	2009-10-22	1	50	2009-10-22	0	0
ST16	2009-10-22	0	0	2009-10-22	0	0
ST17	2009-10-22	827	10	2009-10-22	507	5
ST18	2009-10-22	368	10	2009-10-22	225	10
ST19	2009-10-22	0	0	2009-10-22	0	
ST20	2009-10-22	5	50	2009-10-22	7	5
ST21	2009-10-22	3	10	2009-10-22	1	5
ST22	2009-10-22	0	0	2009-10-22	0	0
ST23	2009-10-22	1	50	2009-10-22	0	0
ST24	2009-10-22	0	0	2009-10-22	0	0
		3878	**260**		**4610**	**40**
			Included			**Excluded**

	SCOPUS http://www.scopus.com			SpringerLink http://www.springerlink.com		
	Date of search	# hits	Relevance	Date of search	# hits	Relevance
ST1	2009-10-23	501	5	2009-10-25	4 516	10
ST2	2009-10-23	280	10	2009-10-25	4 779	10
ST3	2009-10-23	501	5	2009-10-25	8	0
ST4	2009-10-23	3	5	2009-10-25	548	5
ST5	2009-10-23	2	0	2009-10-25	556	5
ST6	2009-10-23	0	0	2009-10-25	0	0
ST7	2009-10-23	0	0	2009-10-25	288	5
ST8	2009-10-23	0	0	2009-10-25	0	0
ST9	2009-10-23	2 710	5	2009-10-25	13 933	5
ST10	2009-10-23	1 178	10	2009-10-25	6 974	5
ST11	2009-10-23	1	0	2009-10-25	14	0
ST12	2009-10-23	23	50	2009-10-25	1 013	10
ST13	2009-10-23	7	0	2009-10-25	893	5
ST14	2009-10-23	0	0	2009-10-25	0	0
ST15	2009-10-23	1	0	2009-10-25	444	5
ST16	2009-10-23	0	0	2009-10-25	0	0
ST17	2009-10-23	951	5	2009-10-25	2 646	5
ST18	2009-10-23	420	50	2009-10-25	2 260	10
ST19	2009-10-23	1	0	2009-10-25	3	0
ST20	2009-10-23	12	10	2009-10-25	456	10
ST21	2009-10-23	2	5	2009-10-25	410	10
ST22	2009-10-23	0	0	2009-10-25	0	0
ST23	2009-10-23	0	0	2009-10-25	238	10
ST24	2009-10-23	0	0	2009-10-25	0	0
		6593	**160**		**39979**	**110**
			Included			**Excluded**

	Wiley Interscience http://www3.interscience.wiley.com			Google Scholar http://scholar.google.com		
	Date of search	# hits	Relevance	Date of search	# hits	Relevance
ST1	2009-10-23	134	10	2009-10-25	18 300	10
ST2	2009-10-23	295	5	2009-10-25	18 700	10
ST3	2009-10-23	0	0	2009-10-25	45	0
ST4	2009-10-23	3	0	2009-10-25	4 550	5
ST5	2009-10-23	5	5	2009-10-25	5 090	5
ST6	2009-10-23	0	0	2009-10-25	5	0
ST7	2009-10-23	1	0	2009-10-25	2 880	5
ST8	2009-10-23	0	0	2009-10-25	3	0
ST9	2009-10-23	459	5	2009-10-25	16 900	10
ST10	2009-10-23	2 495	5	2009-10-25	33 800	10
ST11	2009-10-23	0	0	2009-10-25	116	0
ST12	2009-10-23	4	0	2009-10-25	6 980	5
ST13	2009-10-23	15	0	2009-10-25	7 820	10
ST14	2009-10-23	0	0	2009-10-25	9	0
ST15	2009-10-23	1	0	2009-10-25	3 750	10
ST16	2009-10-23	0	0	2009-10-25	2	0
ST17	2009-10-23	68	10	2009-10-25	12 200	5
ST18	2009-10-23	136	10	2009-10-25	12 700	10
ST19	2009-10-23	0	0	2009-10-25	8	0
ST20	2009-10-23	1	0	2009-10-25	2 630	10
ST21	2009-10-23	4	10	2009-10-25	2 930	10
ST22	2009-10-23	0	0	2009-10-25	2	0
ST23	2009-10-23	0	0	2009-10-25	1 870	10
ST24	2009-10-23	0	0	2009-10-25	1	0
		3621	**60**		**151291**	**125**
			Excluded			**Included**

Appendix B Systematic review, iteration two

ID	Search term
ST1	"distributed computing" AND heterogeneous
ST2	"distributed computing" AND scalable
ST3	"open source" AND "distributed computing" AND heterogeneous
ST4	"open source" AND "distributed computing" AND scalable
ST5	grid AND heterogeneous
ST6	grid AND scalable
ST7	"open source" AND grid
ST8	"grid computing" AND heterogeneous
ST9	"grid computing" AND scalable
ST10	"open source" AND "grid computing"

	ArXiv http://arxiv.org			**CiteSeer** http://citeseerx.ist.psu.edu		
	Advanced search			Advanced search		
	Search in:	Full record		Search in:	Text	
	Subject areas:	Computer Science		Subject areas:	All	
	Years:	2004 – 2009		Years:	2004 – 2009	
	Sort by:	-		Sort by:	Relevance	
	Document type:	All		Document type:	All	
	Date of search	**# hits**	**Considered**	**Date of search**	**# hits**	**Considered**
ST1	2009-10-26	7	7	2009-10-26	921	100
ST2	2009-10-26	12	12	2009-10-29	1016	100
ST3	2009-10-26	0	0	2009-10-29	125	125
ST4	2009-10-26	0	0	2009-10-29	134	134
ST5	2009-10-26	33	33	2009-10-29	2143	100
ST6	2009-10-26	22	22	2009-10-29	2511	100
ST7	2009-10-26	8	8	2009-10-29	715	100
ST8	2009-10-26	9	9	2009-10-29	753	100
ST9	2009-10-26	8	8	2009-10-29	663	100
ST10	2009-10-26	3	3	2009-10-29	219	100
		102	**102**		**9200**	**1059**

	IEEE Xplore http://ieeexplore.ieee.org/		
	Advanced search Search in: All fields Subject areas: All Years: 2004 – 2009 Sort by: Relevance Document type: No books or educational courses		
	Date of search	**# hits**	**Considered**
ST1	2009-10-29	361	100
ST2	2009-10-29	248	100
ST3	2009-10-29	1	1
ST4	2009-10-29	0	0
ST5	2009-10-29	1004	100
ST6	2009-10-29	534	100
ST7	2009-10-29	66	66
ST8	2009-10-29	780	100
ST9	2009-10-29	350	100
ST10	2009-10-29	47	47
		3391	**714**

	SCOPUS http://www.scopus.com		
	Basic search Search in: Article Title, Abstract, Keywords Subject areas: Physical Sciences Years: 2004 – 2009 Sort by: Relevance Document type: All		
	Date of search	**# hits**	**Considered**
ST1	2009-10-29	373	100
ST2	2009-10-29	218	100
ST3	2009-10-29	3	3
ST4	2009-10-29	2	2
ST5	2009-10-29	2313	100
ST6	2009-10-29	1071	100
ST7	2009-10-29	213	100
ST8	2009-10-29	911	100
ST9	2009-10-29	402	100
ST10	2009-10-29	86	86
		5592	**791**

	Google Scholar http://scholar.google.com		
	Advanced search Search in: All Fields Subject areas: All Years: 2004 – 2009 Sort by: Relevance Document type: All		
	Date of search	**# hits**	**Considered**
ST1	2009-10-30	15900	100
ST2	2009-10-30	16300	100
ST3	2009-10-30	3240	100
ST4	2009-10-30	4070	100
ST5	2009-10-30	27600	100
ST6	2009-10-30	17000	100
ST7	2009-10-30	16100	100
ST8	2009-10-30	10700	100
ST9	2009-10-30	11200	100
ST10	2009-10-30	4390	100
		126500	**1000**

Appendix C Systematic review, iteration three

ID	Search term
ST11	"desktop grid*" AND heterogeneous
ST12	"desktop grid*" AND scalable

	ArXiv		
	http://arxiv.org		
	Advanced search		
	Search in: Full record		
	Subject areas: Computer Science		
	Years: 2004 – 2009		
	Sort by: -		
	Document type: All		
	Date of search	**# hits**	**Considered**
ST11	2009-11-12	0	0
ST12	2009-11-12	0	0
		0	**0**

	CiteSeer		
	http://citeseerx.ist.psu.edu		
	Advanced search		
	Search in: Text		
	Subject areas: All		
	Years: 2004 – 2009		
	Sort by: Relevance		
	Document type: All		
	Date of search	**# hits**	**Considered**
ST11	2009-11-12	34	34
ST12	2009-11-12	0	0
		34	**34**

	IEEE Xplore		
	http://ieeexplore.ieee.org/		
	Advanced search		
	Search in: All fields		
	Subject areas: All		
	Years: 2004 – 2009		
	Sort by: Relevance		
	Document type: No books or educational courses		
	Date of search	**# hits**	**Considered**
ST11	2009-11-12	14	14
ST12	2009-11-12	11	11
		25	**25**

	SCOPUS		
	http://www.scopus.com		
	Basic search		
	Search in: Article Title, Abstract, Keywords		
	Subject areas: Physical Sciences		
	Years: 2004 – 2009		
	Sort by: Relevance		
	Document type: All		
	Date of search	**# hits**	**Considered**
ST11	2009-11-12	43	43
ST12	2009-11-12	30	30
		73	**73**

	Google Scholar		
	http://scholar.google.com		
	Advanced search		
	Search in: All Fields		
	Subject areas: All		
	Years: 2004 – 2009		
	Sort by: Relevance		
	Document type: All		
	Date of search	**# hits**	**Considered**
ST11	2009-11-12	381	100
ST12	2009-11-12	396	100
		777	**200**

Appendix D Systematic review, quality assessment

#	Article	QAQ 1	2	3	4	5	Sum
1	Anderson. BOINC: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (2004) pp. 10	1	1	1	1	1	5
2	Sterck et al. Database-Driven Grid Computing with GridBASE. Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on (2007) vol. 1	1	1	1	1	1	5
3	Costa et al. MyGrid–A complete solution for running Bag-of-Tasks Applications. Proc. of the SBRC 2004–Salao de Ferramentas (22nd Brazilian Symposium on Computer Networks–III Special Tools Session) (2004)	1	1	1	1	1	5
4	Andrade et al. Peer-to-peer grid computing with the OurGrid Community. Proceedings of the SBRC (2005) pp. 1-8	1	1	1	1	1	5
5	Kacsuk et al. SZTAKI Desktop Grid (SZDG): A Flexible and Scalable Desktop Grid System. Journal of Grid Computing (2009) pp. 1-23	1	1	1	1	1	5
6	Kacsuk et al. Scalable desktop grid system. Lecture Notes in Computer Science (2007) vol. 4395 pp. 27	1	1	1	1	1	5
7	Teo and Wang. Alice: A scalable runtime infrastructure for high performance grid computing. Lecture Notes in Computer Science (2004) pp. 101-109	1	1	1	1	1	5
8	Myers et al. Expanding the reach of Grid computing: Combining Globus-and BOINC-based systems. Grid Computing for Bioinformatics and Computational Biology (2007) pp. 71	1	1	1	1	1	5
9	Bal et al. Ibis: Real-world problem solving using real-world grids. Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing-Volume 00 (2009) pp. 1-8	1	1	1	1	1	5
10	van Nieuwpoort et al. Satin: Simple and efficient java-based grid programming. Scalable Computing: Practice and Experience (2005) vol. 6 (3) pp. 19-32	1	1	1	1	1	5
11	Constantinescu-Fülöp. A Desktop Grid Computing Approach for Scientific Computing and Visualization. idi.ntnu.no (2008)	1	1	1	1	0,5	4,5
12	Genaud and Rattanapoka. A Peer-to-Peer Framework for Robust Execution of Message Passing Parallel Programs on Grids. Lecture Notes in Computer Science (2005) vol. 3666 pp. 276	1	1	1	1	0,5	4,5
13	Therning and Bengtsson. Jalapeno: decentralized grid computing using peer-to-peer technology. Proceedings of the 2nd conference on Computing frontiers (2005) pp. 65	1	1	1	1	0,5	4,5
14	Carvalho et al. Squid – a simple bioinformatics grid. BMC bioinformatics (2005) vol. 6 (1) pp. 197	1	1	1	1	0,5	4,5
15	Iosup et al. Synthetic grid workloads with Ibis, KOALA, and GrenchMark. Proceedigs of the CoreGRID Integrated Research in Grid Computing, Pisa, Italy (2005)	1	1	0,5	1	0,5	4
16	Keane et al. A high-throughput bioinformatics distributed computing platform. (2005)	1	1	1	0	1	4
17	Cahon et al. An enabling framework for parallel optimization on the computational grid. Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on (2005) vol. 2 pp. 702- 709 Vol. 2	1	1	0,5	1	0,5	4
18	Sun et al. CROWN Node Server: An Enhanced Grid Service Container Based on GT4 WSRF Core. Grid and Cooperative Computing Workshops (2006)	1	0	1	1	1	4
19	Schulz et al. Orbweb—A Network Substrate for Peer-to-Peer Desktop Grid Computing Based on Open Standards. Journal of Grid Computing (2009) pp. 1-31	1	1	1	0,5	0,5	4
20	Lacour et al. A Software Architecture for Automatic Deployment of CORBA Components Using Grid Technologies. arXiv (2004) vol. cs.NI	1	1	1	0,5	0,5	4
21	De Sterck et al. A lightweight, scalable grid computing framework for parallel bioinformatics applications. High Performance Computing Systems and Applications, 2005. HPCS 2005. 19th International Symposium on (2005) pp. 251- 257	1	1	1	0,5	0,5	4
22	Choi et al. Characterizing and classifying desktop grid. Seventh IEEE International Symposium on Cluster … (2007)	1	1	1	1	0	4
23	Thain et al. Distributed computing in practice: The Condor experience. Concurrency and Computation Practice and Experience (2005) vol. 17 (2-4) pp. 323-356	0,5	1	0,5	1	1	4
24	Foster. Globus toolkit version 4: Software for service-oriented systems. Journal of Computer Science and Technology (2006) vol. 21 (4) pp. 513-520	1	0,5	0,5	1	1	4
25	Mateescu and Sosonkina. IMAGE: An approach to building standards-based enterprise Grids. Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International (2006) pp. 8	0,5	1	1	1	0,5	4
26	Amoretti et al. SP2A: a Service-oriented Framework for P2P-based Grids. Proceedings of the 3rd international workshop on Middleware for grid computing (2005) pp. 6	1	1	0,5	1	0,5	4

27	Buyya and Venugopal. The gridbus toolkit for service oriented grid and utility computing: An overview and status report. Proceedings of the First IEEE International Workshop on Grid Economics and Business Models (2004) pp. 19-36	1	0	1	1	1	4
28	Tanaka et al. A case study of a linux grid on windows using virtual machines. Proceedings of the 22nd International Conference on Advanced Information Networking and Applications-Workshops (2008) pp. 195-200	1	0,5	0,5	1	0,5	3,5
29	Cheng et al. A parallel loop self-scheduling on grid computing environments. Proceedings of the 2004 IEEE International Symposium on Parallel Architectures, Algorithms and Networks pp. 409-414	1	0,5	1	0,5	0,5	3,5
30	Denis and Perez. Network communications in grid computing: at a crossroads between parallel and distributed worlds. Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International (2004) pp. 95 EP -	1	0	1	1	0,5	3,5
31	Caromel et al. Peer-to-peer for computational grids: mixing clusters and desktop machines. Parallel Computing (2007) vol. 33 (4-5) pp. 275-288	1	1	1	0	0,5	3,5
32	Amin et al. Abstracting the grid. Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2004) (2004) pp. 11-13	1	0	1	1	0,5	3,5
33	Miyazwa et al. Development of grid middleware CyberGRIP and its applications. e-Science and Grid Computing (2005)	1	0	1	0,5	1	3,5
34	Crespo et al. GridQTC: A Desktop Client for the Computational Chemistry Grid Infrastructure. Proceedings of the 2008 The Second International … (2008)	1	0	1	1	0,5	3,5
35	Abdennadher et al. Initializing a National Grid Infrastructure—Lessons Learned from the Swiss National Grid Association Seed Project. Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (2008) pp. 169-176	1	1	0,5	0,5	0,5	3,5
36	Gadgil et al. Managing Grid messaging middleware. Challenges of Large Applications in Distributed Environments (CLADE) (2006) pp. 83-91	0,5	1	1	0,5	0,5	3,5
37	Breuer et al. Scientific computing with UNICORE. NIC Symposium (2004) pp. 429–440	1	0	0,5	1	1	3,5
38	Streit et al. UNICORE-from project results to production grids. Grid Computing: The New Frontiers of High Performance Processing, Advances in Parallel Computing (2005) vol. 14 pp. 357–376	1	0	0,5	1	1	3,5
39	Pullen et al. Using Web services to integrate heterogeneous simulations in a grid environment. Future Generation Computer Systems (2005) vol. 21 (1) pp. 97-106	1	0	1	1	0,5	3,5
40	Li and Parashar. A Decentralized Computational Infrastructure for Grid-Based Parallel Asynchronous Iterative Applications. Journal of Grid Computing (2006)	0,5	1	1	0	0,5	3
41	Yongh et al. Development of Scalable Service-Oriented Grid Computing Architecture. (2009)	0,5	1	1	0	0,5	3
42	Zhou and Lo. Wavegrid: A scalable fast-turnaround heterogeneous peer-based desktop grid system. Proceedings of the 20th International Parallel & Distributed Processing Symposium (2006)	1	1	0,5	0	0,5	3
43	Pathak et al. A framework for dynamic resource management on the grid. HP Laboratories Palo Alto (2005)	1	1	1	0	0	3
44	Khanli and Analoui. Grid-JQA a new architecture for QoS-guaranteed grid computing system. Parallel, Distributed, and Network-Based Processing, 2006. PDP 2006. 14th Euromicro International Conference on (2006) pp. 4	0,5	1	1	0	0,5	3
45	Grace et al. Gridkit: Pluggable overlay networks for grid computing. Lecture Notes in Computer Science (2004) pp. 1463-1481	0,5	1	0,5	0,5	0,5	3
46	Endsuleit and Calmet. Introducing robust and private computation into grid technology. Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. 13th IEEE International Workshops on (2004) pp. 303-308	1	1	0,5	0	0,5	3
47	Matsuoka et al. Japanese computational grid research project: NAREGI. Proceedings of the IEEE (2005) vol. 93 (3) pp. 522-533	0,5	0,5	1	0,5	0,5	3
48	Eerola et al. Roadmap for the ARC Grid middleware. Lecture Notes in Computer Science (2007)	1	0	0	1	1	3
49	Sundaram et al. Sun Grid Engine Package for OSCAR - A Google Summer Of Code 2005 Project. High-Performance Computing in an Advanced Collaborative Environment, 2006. HPCS 2006. 20th International Symposium on (2006) pp. 41- 41	1	0	0,5	1	0,5	3
50	Calder et al. The entropia virtual machine for desktop grids. VEE '05: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments (2005)	0,5	1	0,5	0	1	3
51	Smith et al. Towards a service-oriented ad hoc grid. Proceedings of the 3rd International Symposium on Parallel and Distributed Computing (2004) pp. 201-208	0,5	1	1	0	0,5	3
52	Huang et al. Ws-messenger: A web services based messaging system for service-oriented grid computing. 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid06) (2006)	0,5	0	1	1	0,5	3
53	Saltz et al. caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid. Bioinformatics (2006) vol. 22 (15) pp. 1910	1	0	0	1	1	3
54	Kuo et al. Construct a grid computing environment for bioinformatics. Parallel Architectures, Algorithms and Networks, 2004. Proceedings. 7th International Symposium on (2004) pp. 339-344	1	0	0,5	0,5	0,5	2,5
55	Qi Huang et al. Development of an MPI for Power System Distributed Parallel Computing in Wide Area Network with P2P Technology. Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on (2006) pp. 411-416	0,5	0	1	0,5	0,5	2,5

56	Liu et al. Experimental Investigation of a Peer-to-Peer-Based Architecture for Emerging Consumer Grid Applications. Journal of Optical Communications and Networking (2009) vol. 1 (1) pp. 57-68	1	1	0	0	0,5	2,5
57	Deak. Grid service execution for JOpera. (2005)	1	0	1	0	0,5	2,5
58	Kenny et al. Heterogeneous grid computing: Issues and early benchmarks. International Conference on Computational Science (3) (2005)	1	0	1	0	0,5	2,5
59	Pichitlamken et al. High performance spreadsheet simulation on a desktop grid. Proceedings of the 40th Conference on Winter Simulation (2008) pp. 663-670	1	0,5	0,5	0	0,5	2,5
60	Karlsen and Vinter. Minimum intrusion grid–the simple model. Proceedings of ETNGRID-2005 (2005)	1	1	0	0	0,5	2,5
61	Tanaka et al. Parallel Computing of CG Using an Open Source Windows Grid. Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on (2008) pp. 173-180	1	0	0,5	0,5	0,5	2,5
62	Foster et al. The Grid2003 production Grid: Principles and practice. 13th International Symposium on High Performance Distributed Computing (2004)	1	0	0	0,5	1	2,5
63	Podvinec et al. The SwissBioGrid Project: objectives, preliminary results and lessons learned. Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006)–Workshop on Production Grids. IEEE Computer Society Press (2006) pp. 148-56	1	0	0,5	0	1	2,5
64	Vinter. The architecture of the minimum intrusion Grid (MiG). Communicating process architectures 2005: WoTUG-28: proceedings of the 28th WoTUG Technical Meeting, 18-21 September 2005, Technische Universiteit Eindhoven, The Netherlands (2005) pp. 189	1	1	0	0	0,5	2,5
65	Huibin et al. A Grid Workflow Process Engine: Architecture and Simulation. (2009)	0,5	0	1	0,5	0,5	2,5
66	He et al. BLAST Application with Data-Aware Desktop Grid Middleware. Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid-Volume 00 (2009) pp. 284-291	0,5	1	0	0,5	0,5	2,5
67	Kim et al. Creating a Robust Desktop Grid using Peer-to-Peer Services. Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International (2007) pp. 1-7	1	1	0	0	0,5	2,5
68	Benson et al. Evaluation of UDDI as a provider of resource discovery services for OGSA-based grids. Parallel and Distributed Processing Symposium (2006) vol. IPDPS 2006 (20th International) pp. 9-21	0,5	1	0	0,5	0,5	2,5
69	Altintas et al. A framework for the design and reuse of grid workflows. Lecture Notes in Computer Science (2005) vol. 3458 pp. 120-133	0,5	1	1	0	0	2,5
70	Pheatt. An easy to use distributed computing framework. Proceedings of the 38th SIGCSE technical symposium on Computer science education (2007) pp. 575	1	0,5	0,5	0	0,5	2,5
71	Tsaregorodtsev et al. Dirac: A scalable lightweight architecture for high throughput computing. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (2004) pp. 19-25	0,5	0	0,5	0,5	1	2,5
72	JT Moscicki et al. Ganga: a tool for computational-task management and easy access to Grid resources. arXiv (2009) vol. cs.DC	0,5	0	1	0	1	2,5
73	Amin et al. Gridant: A client-controllable grid workflow system. 37th Hawai'i International Conference on System Science (2004) pp. 5-8	0,5	0	1	0,5	0,5	2,5
74	Li et al. Heterogeneous grid computing for energy constrained mobile device. Lecture Notes in Computer Science (2004)	1	0	1	0	0,5	2,5
75	Amendolia et al. MammoGrid: A Service Oriented Architecture based Medical Grid Application. arXiv (2004) vol. cs.DC	1	0,5	0	0,5	0,5	2,5
76	Charbonneau and Terskikh. SpectroGrid: Providing Simple Secure Remote Access to Scientific Instruments. High Performance Computing Systems and Applications, 2008. HPCS 2008. 22nd International Symposium on (2008) pp. 76-82	1	0	0	0,5	1	2,5
77	Ellahi et al. Transparent Migration of Multi-Threaded Applications on a Java Based Grid. arXiv (2006) vol. cs.DC	0,5	1	0	0,5	0,5	2,5
78	Wei et al. A Scalable Grid Information Service Framework for Engineering and Scientific Computation. Information Technology Journal	0,5	1	0	0	0,5	2
79	Luther et al. Alchemi: A .NET-based enterprise grid computing system. Proceedings of the 6th International Conference on Internet Computing (ICOMP'05) (2005) pp. 27-30	1	0	0,5	0	0,5	2
80	Sosa and Grimshaw. Bringing the Grid home. Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing-Volume 00 (2008) pp. 152-159	0,5	0	0,5	0,5	0,5	2
81	Al Zain et al. Evaluating a High-Level Parallel Language (GpH) for Computational GRIDs. IEEE Transactions on Parallel and Distributed Systems (2008) vol. 19 (2) pp. 219	0,5	0	1	0	0,5	2
82	Georgiou et al. Evaluations of the Lightweight Grid CIGRI upon the Grid5000 Platform. Proceedings of the Third IEEE International Conference on e-Science and Grid Computing (2007) pp. 279-286	0,5	0	0	1	0,5	2
83	Xiaohui et al. GDIA: A Scalable Grid Infrastructure for Data Intensive Applications. 2006 IEEE International Conference on Hybrid Information Technology (2006) pp. 9-11	0,5	0	0,5	0,5	0,5	2
84	Zhang et al. GridSM: a Norm-driven State Machine Model of Grid Workflow. Computer Supported Cooperative Work in Design, 2006. CSCWD'06. 10th International Conference on (2006) pp. 1-5	0,5	0	1	0	0,5	2
85	Ali et al. Heterogeneous Relational Databases for a Grid-enabled Analysis Environment. arXiv (2005) vol. cs.DC	0,5	0	0,5	0,5	0,5	2

86	Al Zain et al. Managing heterogeneity in a grid parallel haskell. J. Scalable Comp.: Practice and Experience (2006) vol. 6 (4)	0,5	0	0,5	0,5	0,5	2
87	Marsh et al. Matchmaking and implementation issues for a P2P desktop grid. IEEE International Symposium on Parallel and Distributed Processing, 2008. IPDPS 2008 (2008) pp. 1-5	0,5	1	0	0	0,5	2
88	Aumage et al. Netibis: An efficient and dynamic communication system for heterogeneous grids. Proc. of the Cluster Computing and Grid 2005 Conference (CCGrid 2005), Cardiff, UK (2005)	1	0	0,5	0	0,5	2
89	Cannataro et al. Proteus, a Grid based Problem Solving Environment for Bioinformatics: Architecture and Experiments. (2004)	0,5	0	0,5	0,5	0,5	2
90	Reddy et al. Vishwa: A reconfigurable P2P middleware for Grid Computations. Proc. of the 2006 International Conference on Parallel Processing. IEEE Press (2006)	1	0	0,5	0	0,5	2
91	Weishuai Yang et al. Automatic Clustering for Self-Organizing Grids. Cluster Computing, 2006 IEEE International Conference on (2006) pp. 1-9	0,5	1	0	0	0,5	2
92	Shang et al. A Grid Workflow Framework with High Scalability and Usability. Grid and Cooperative Computing (2008)	0,5	0,5	0	0,5	0,5	2
93	Zhou et al. An Open Source Toolkit for Grid Resources Scheduling in Digital Library. Hybrid Intelligent Systems, 2009. HIS '09. Fifth International Conference on (2009) vol. 3 pp. 121-124	1	0	0,5	0	0,5	2
94	Jin. ChinaGrid: Making grid computing a reality. Digital Libraries: International Collaboration and Cross-Fertilization-Lecture Notes in Computer Science (2004) vol. 3334 pp. 13-24	1	0	0	0	1	2
95	Youn et al. GEONGrid portal: design and implementations. CONCURRENCY AND COMPUTATION (2007) vol. 19 (12) pp. 1597	0,5	0	0,5	0,5	0,5	2
96	GuiLing et al. Service-oriented grid architecture and middleware technologies for collaborative e-learning. 2005 IEEE International Conference on Services Computing (2005) vol. 2	1	0	0	0	1	2
97	Alameda et al. The Open Grid Computing Environments collaboration: portlets and services for science gateways. CONCURRENCY AND COMPUTATION (2007) vol. 19 (6) pp. 921	0,5	0	0,5	0,5	0,5	2
98	Huang. VIRGO: Virtual Hierarchical Overlay Network for Scalable Grid Computing. Lecture Notes in Computer Science (2005)	0,5	1	0	0	0,5	2
99	Xu et al. Vega: A computer systems approach to grid computing. Journal of Grid Computing (2004) vol. 2 (2) pp. 109-120	1	0	0	0,5	0,5	2
100	Riedel et al. Web Services Interfaces and Open Standards Integration into the European UNICORE 6 Grid Middleware. EDOC Conference Workshop, 2007. EDOC '07. Eleventh International IEEE (2007) pp. 57-60	1	0	0	1	0	2
101	Aloisio et al. iGrid, a novel grid information service. Lecture Notes in Computer Science (2005) vol. 3470 pp. 506	0,5	0	0	1	0,5	2
102	Yu and Law. Grid Computing on Massively Multi-User Online Platform. Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on (2007) pp. 135-140	1	0	0	0	0,5	1,5
103	Surridge et al. Experiences with GRIA—Industrial Applications on a Web Services Grid. Proceedings of the First International Conference on e-Science and Grid Computing (2005) pp. 98-105	0,5	0	0	0	1	1,5
104	Wang et al. Flexible Grid Portlets to Access Multi Globus Toolkits. GCC2006: CVRE06 Workshop. Changsha, China (2006) pp. 21-23	0,5	0	0	0,5	0,5	1,5
105	Guan et al. Grid-Flow: a Grid-enabled scientific workflow system with a Petri-net-based interface. CONCURRENCY AND COMPUTATION (2006) vol. 18 (10) pp. 1115	0,5	0	1	0	0	1,5
106	Shivle et al. Mapping of subtasks with multiple versions in a heterogeneous ad hoc grid environment. 3rd International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (HeteroPar 2004) in the Proceedings of the 3rd International Symposium on Parallel and Distributed Computing in association with HeteroPar'04 (ISPDC/HeteroPar 2004)	0,5	0	0	0	0,5	1
107	Ruiz-Alvarez et al. BES++: HPC Profile open source C implementation. Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing-Volume 00 (2008) pp. 41-48	0,5	0	0	0	0,5	1
108	Cybok. A Grid workflow infrastructure. Concurrency and Computation (2006) vol. 18 (10) pp. 1243	0,5	0	0	0	0,5	1
109	Bickson et al. Peer-to-Peer Secure Multi-Party Numerical Computation. arXiv (2008) vol. cs.CR						Rejected EC2
110	Wen et al. Crossing Heterogeneous Grid Systems with a Single Sign-On Scheme Based on a P2P Layer. IEEE Asia-Pacific Services Computing Conference (2008)						Rejected EC3
111	Patoli et al. An open source Grid based render farm for Blender 3D. Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES (2009) pp. 1-6						Rejected EC4
112	Park and Fujimoto. A scalable framework for parallel discrete event simulations on desktop grids. Proceedings of the 8th IEEE/ACM International Conference on Grid Computing (2007) pp. 185-192						Rejected EC5
113	Duarte et al. Using the computational grid to speed up software testing. Proceedings of the 19th Brazilian Symposium on Software Engineer (2005)						Rejected EC5
114	Silva et al. Design and Evaluation of a Grid Computing Based Architecture for Integrating Heterogeneous IDSs. IEEE Global Telecommunications Conference (2007)						Rejected EC5
115	Xie et al. CNGrid Software 2: Service Oriented Approach to Grid Computing. the proceedings of the UK e-Science All Hands Meeting (2005) pp. 701-708						Rejected EC6

116	Wong et al. A Global Criminal and Terrorist Tracking Framework Using Semantic Grid Technologies. IEEE Congress on Services Part II, 2008. SERVICES-2 (2008) pp. 157-162	Rejected EC7
117	Holliday et al. A geographically-distributed, assignment-structured undergraduate grid computing course. Proceedings of the 36th SIGCSE technical symposium on Computer science education (2005) pp. 206-210	Rejected EC7
118	Shi et al. A grid computing integrated distributed CAD system. Computer Supported Cooperative Work in Design (2004)	Rejected EC7
119	Lican et al. A scalable semantic grid framework--VDHA/spl I. bar/Grid. 2004 IEEE International Conference on Systems, Man and Cybernetics (2004) vol. 5	Rejected EC7
120	El-Darieby and Krishnamurthy. A scalable wide-area grid resource management framework. ICNS'06. International conference on Networking and Services (2006) pp. 76–86	Rejected EC7
121	Corcho et al. An overview of s-ogsa: a reference semantic grid architecture. Web Semantics: Science, Services and Agents on the World Wide Web (2006) vol. 4 (2) pp. 102-115	Rejected EC7
122	Li et al. Concurrent Negotiations for Agent-Based Grid Computing. 5th IEEE International Conference on Cognitive Informatics, 2006. ICCI 2006 (2006) vol. 1	Rejected EC7
123	Lehman et al. DRAGON: A framework for service provisioning in heterogeneous grid networks. IEEE Communications Magazine (2006) vol. 44 (3) pp. 84-90	Rejected EC7
124	Jian-Hua Li et al. Framework of grid computing based on semantic web service. Machine Learning and Cybernetics, 2009 International Conference on (2009) vol. 3 pp. 1523-1527	Rejected EC7
125	Andrzejak et al. Grid Architectural Issues: State-of-the-art and Future Trends. CoreGRID (2008) (WP-0004)	Rejected EC7
126	Amin et al. Grid computing for the masses: An overview. Lecture Notes in Computer Science (2004) pp. 464-473	Rejected EC7
127	Berlich et al. Grid computing in Europe: from research to deployment. Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44 (2005) pp. 27	Rejected EC7
128	Freund et al. Health-e-Child : An Integrated Biomedical Platform for Grid-Based Paediatric Applications. arXiv (2006) vol. cs.DC	Rejected EC7
129	Cannataro and Talia. Semantics and knowledge grids: building the next-generation grid. IEEE Intelligent Systems (2004) pp. 56-63	Rejected EC7
130	Amin et al. Toward an architecture for ad hoc grids. 12th International Conference on Advanced Computing and Communications (ADCOM 2004) (2004) pp. 15-18	Rejected EC7
131	Coulson et al. Towards a component-based middleware framework for configurable and reconfigurable grid computing. Proc. Workshop on Emerging Technologies for Next Generation Grid (ETNGRID-2004), associated with 13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE-2004), Modena, Italy (2004)	Rejected EC7
132	Al Zain. Implementing high-level parallelism on computational GRIDs. (2006)	Rejected EC9
133	Arefin et al. Alchemi vs Globus: a performance comparison. 4th International Conference on Electrical and Computer Engineering (2006) pp. 19-21	Rejected EC10
134	Peng et al. Benchmark performance on cluster grid with NGB. Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International (2004) pp. 275	Rejected EC10
135	Asadzadeh et al. Global grids and software toolkits: A study of four grid middleware technologies. High Performance Computing: Paradigm and Infrastructure (2005)	Rejected EC10
136	Wu et al. Grid middleware in China. International Journal of Web and Grid Services (2007)	Rejected EC10
137	Manset et al. Managing Separation of Concerns in Grid Applications Through Architectural Model Transformations. arXiv (2007) vol. cs.SE	Rejected EC10
138	Flissi and Merle. A Generic Deployment Framework for Grid Computing and Distributed Applications. arXiv (2007) vol. cs.DC	Rejected EC11
139	Goscinski and Abramson. Application deployment over heterogeneous grids using distributed ant. E-SCIENCE (2009) vol. 5 pp. 361–368	Rejected EC11
140	Byun and Kim. DynaGrid: A dynamic service deployment and resource migration framework for WSRF-compliant applications. Parallel Computing (2007) vol. 33 (4-5) pp. 328-338	Rejected EC11
141	Lin et al. CoreP2P: a tailored group communication scheme for P2P-grid GIS. Proceedings of SPIE (2009)	Rejected EC12
142	Huang et al. Grid architecture for scalable e-science and its prototype. Ruan Jian Xue Bao(J. Softw.) (2005)	Rejected EC12
143	Bendjoudi et al. P2P design and implementation of a parallel branch and bound algorithm for grids. International Journal of Grid and Utility Computing (2009)	Rejected EC12
144	Chakravarti et al. The organic grid: Self-organizing computation on a peer-to-peer network. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans (2005) vol. 35 (3) pp. 373-384	Rejected EC13
145	Azab and Kholidy. An adaptive decentralized scheduling mechanism for peer-to-peer Desktop Grids. Computer Engineering & Systems, 2008. ICCES 2008. International Conference on (2008) pp. 364-371	Rejected EC13
146	Harvey et al. Design and performance of a heterogeneous grid partitioner. Algorithmica (2006) vol. 45 (3) pp. 509-530	Rejected EC13

147	Ismail. Dynamic Resource Allocation Mechanisms for Grid Computing Environment. Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on (2007) pp. 1-5	Rejected EC13
148	Al-Azzoni and Down. Dynamic scheduling for heterogeneous Desktop Grids. Proceedings of the 9th International Conference on Grid Computing (2008) pp. 136–143	Rejected EC13
149	Page and Naughton. Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing. Proceedings of the 19th International Parallel and Distributed Processing Symposium (2005)	Rejected EC13
150	Lazarevic et al. Enabling Adaptive Grid Scheduling and Resource Management. arXiv (2007) vol. cs.DC	Rejected EC13
151	Zhang et al. GeoGrid: A Scalable Location Service Network. Proceedings of 27th IEEE International Conference on Distributed Computing Systems (ICDCS 2007) (2007)	Rejected EC13
152	Lu et al. On the performance-driven load distribution for heterogeneous computational grids. Journal of Computer and System Sciences (2007)	Rejected EC13
153	Becker et al. Performance Evaluation and Optimization of Parallel Grid Computing Applications. Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on (2008) pp. 193-199	Rejected EC13
154	Cremonesi and Turrin. Performance models for desktop grids. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2007), San Diego, CA (2007) pp. 16-18	Rejected EC13
155	Sumathi and Gopalan. Priority Based Scheduling for Heterogeneous Grid Environments. 10th IEEE Singapore International Conference on Communication systems, 2006. ICCS 2006 (2006) pp. 1-5	Rejected EC13
156	Menasce and Casalicchio. Quality of service aspects and metrics in grid computing. Proc. 2004 Computer Measurement Group Conference (2004)	Rejected EC13
157	Shen and Li. SPPS: A scalable P2P-based proximity-aware multi-resource discovery scheme for grids. IEEE Military Communications Conference, 2008. MILCOM 2008 (2008) pp. 1-7	Rejected EC13
158	Banerjee et al. Scalable Grid service discovery based on UDDI. Proceedings of the 3rd international workshop on Middleware for grid computing (2005) pp. 6	Rejected EC13
159	Huang et al. Smartgrid: A fully decentralized grid scheduling framework supported by swarm intelligence. Proceedings of the 2008 Seventh International Conference on Grid and Cooperative Computing (2008) pp. 160-168	Rejected EC13
160	Shivle et al. Static mapping of subtasks in a heterogeneous ad hoc grid environment. 13th IEEE Heterogeneous Computing Workshop (HCW … (2004)	Rejected EC13
161	Rahmeh and Johnson. Towards scalable and reliable Grid Networks. Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on (2008) pp. 253-259	Rejected EC13
162	Gjermundrod et al. g-Eclipse - An integrated framework to access and maintain Grid resources. Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing-Volume 00 (2008) pp. 57-64	Rejected EC13
163	Gil and Choi. A Peer to Peer Grid Computing System Based on Mobile Agents. Lecture Notes in Computer Science (2008)	Rejected IC1
164	Zhao et al. A Scalable and Low-Cost Grid Portal. Grid and Cooperative Computing (2008)	Rejected IC1
165	Nami and Bertels. A Survey of Autonomic Computing Systems. Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on (2007) pp. 26-26	Rejected IC1
166	Ali et al. A Taxonomy and Survey of Grid Resource Planning and Reservation Systems for Grid Enabled Analysis Environment. arXiv (2004) vol. cs.DC	Rejected IC1
167	Ranjan et al. A Taxonomy of Peer-to-Peer Based Complex Queries: a Grid perspective. arXiv (2006) vol. cs.NI	Rejected IC1
168	Kouvakis and Georgatos. A Technical Report On Grid Benchmarking using ATLAS V.O. arXiv (2007) vol. cs.PF	Rejected IC1
169	Wang et al. A middleware framework for maximum likelihood evaluation over dynamic grids. Second IEEE International Conference on e-Science and Grid Computing (2006)	Rejected IC1
170	Androutsellis-Theotokis and Spinellis. A survey of peer-to-peer content distribution technologies. ACM Computing Surveys (CSUR) (2004) vol. 36 (4) pp. 371	Rejected IC1
171	Venugopal et al. A taxonomy of data grids for distributed data sharing, management, and processing. ACM Computing Surveys (CSUR) (2006) vol. 38 (1) pp. 3	Rejected IC1
172	Fahringer et al. ASKALON: a tool set for cluster and Grid computing. Concurrency and Computation: Practice & Experience (2005) vol. 17 (2) pp. 143-169	Rejected IC1
173	Wang et al. An Adaptive Multi-node Downloading Partitioning Algorithm of Distributed Virtual Environment Based on Grid Computing. Image and Graphics, 2007. ICIG 2007. Fourth International Conference on (2007) pp. 1026-1032	Rejected IC1
174	Elmroth et al. An OGSA-based bank service for grid accounting systems. Lecture Notes in Computer Science (2005) vol. 3732 pp. 1051	Rejected IC1
175	Dongarra and Lastovetsky. An overview of heterogeneous high performance and grid computing. Engineering the Grid: Status and Perspective (2006) pp. 1-25	Rejected IC1

176	Kondo et al. Characterizing and evaluating desktop grids: An empirical study. Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'04) (2004)	Rejected IC1
177	Litke et al. Computational workload prediction for grid oriented industrial applications: the case of 3D-image rendering. Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on (2005) vol. 2 pp. 962-969 Vol. 2	Rejected IC1
178	Plaza et al. Distributed Computing for Efficient Hyperspectral Imaging Using Fully Heterogeneous Networks of Workstations. Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on (2006) pp. 28- 28	Rejected IC1
179	Pan et al. Distributed parallel computing using navigational programming. International Journal of Parallel Programming (2004) vol. 32 (1) pp. 1-37	Rejected IC1
180	Clout et al. EHGRID: An emulator of heterogeneous computational grids. Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing-Volume 00 (2009) pp. 1-8	Rejected IC1
181	Jiménez-Peris et al. Enterprise Grids: challenges ahead. Journal of Grid Computing (2007) vol. 5 (3) pp. 283-294	Rejected IC1
182	Laforenza. European Strategies Towards Next Generation Grids. Parallel and Distributed Computing (2006)	Rejected IC1
183	Iosup and Epema. GRENCHMARK: A Framework for Analyzing, Testing, and Comparing Grids. Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on (2006) vol. 1 pp. 313- 320	Rejected IC1
184	Karmon and Liss. GWiQ-P: An EMicient Decentralized Grid-Wide Quota Enforcement Protocol. (2005)	Rejected IC1
185	Bolze et al. Grid'5000: a large scale and highly reconfigurable experimental Grid testbed. International Journal of High Performance Computing Applications (2006) vol. 20 (4) pp. 481	Rejected IC1
186	Ramamurthy. GridFoRCE: A Comprehensive Resource Kit for Teaching Grid Computing. IEEE Transactions on Education (2007) vol. 50 (1) pp. 10-16	Rejected IC1
187	Shah et al. Handling asymmetry in power heterogeneous ad hoc networks. Computer Networks (2007) vol. 51 (10) pp. 2594-2615	Rejected IC1
188	Thain et al. How to measure a large open-source distributed system. CONCURRENCY AND COMPUTATION (2006) vol. 18 (15) pp. 1989	Rejected IC1
189	Vanmechelen and Broeckhove. On using Jini and JXTA in lightweight grids. Lecture Notes in Computer Science (2005) vol. 3470 pp. 384	Rejected IC1
190	Graham et al. Open MPI: A high-performance, heterogeneous MPI. Proceeding of the conference HeteroPar (2006) vol. 6	Rejected IC1
191	Bukhari and Abbas. Optimization & performance analysis of grid protocol for scalable location services. Wireless Ad-Hoc Networks (2004)	Rejected IC1
192	Zhao et al. Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments. arXiv (2008) vol. cs.DC	Rejected IC1
193	Vaniachine and Collaboration. Scalable Database Access Technologies for ATLAS Distributed Computing. arXiv (2009) vol. physics.ins-det	Rejected IC1
194	Aldinucci et al. Structured implementation of component based grid programming environments. Future Generation Grids, CoreGRID series. Springer Verlag (2005)	Rejected IC1
195	Gietz et al. TextGrid and eHumanities. e-Science and Grid Computing, 2006. e-Science '06. Second IEEE International Conference on (2006) pp. 133-133	Rejected IC1
196	Bernholdt et al. The Earth System Grid: Supporting the Next Generation of Climate Modeling Research. arXiv (2007) vol. cs.CE	Rejected IC1
197	Xia et al. The microgrid: Using emulation to predict application performance in diverse grid network environments. Proc. of the Workshop on Challenges of Large Applications in Distributed Environments (CLADE'04) (2004)	Rejected IC1
198	Atkinson et al. Web Service Grids: an evolutionary approach. Concurrency and Computation: Practice & Experience (2005) vol. 17 (2) pp. 377-389	Rejected IC1
199	Malecot et al. XtremLab: A System for Characterizing Internet Desktop Grids. High Performance Distributed Computing, 2006 15th IEEE International Symposium on (2006) pp. 357-358	Rejected IC1
200	Cantino et al. A Low Cost Distributed Computing Approach to Pulsar Searches at a Small College. arXiv (2004) vol. cs.DC	Rejected IC1

Appendix E Survey on Desktop Grid computing

Desktop grid Computing gives the user the ability to utilize idle CPU time and memory on regular workstations Through a local or global (the Internet) network workstations can be linked together and cooperate on heavy load applications.
A Desktop Grid can be seen as an alternative to clusters or super computers, it is also a cheeper alternative since you can use your existing workstations and hardware. The only thing needed is the software.

Below is a couple of questions that will aid me in my work to find how interesting Desktop Grid Computing is. Answer them as good as you can and remember that the questions is about what you think, not what you necessarily know.

*** 1. What is you organizations main area of business? (e.g. mobile applications, software design, business solutions etc.)**

Mobile applications	7
Software design	4
Software development	3
Tele communication	3
Research	2
Configuration	1
Data management	1
Game development	1
IT consulting	1
Web development	1
Total:	**24**

*** 2. What is your role within the organization?**

Provide the most applicable alternative for you.

Developer	7
Project manager	4
Researcher	4
Architect	3
Test engineer	3
Designer	1
Management	1
Team leader	1
Total:	**24**

*** 3. Is your organization using Desktop Grid Computing today?**

Yes	1
No	22
Do not know	1
Totalt:	**24**

*** 4. Disregard cost, time needed and other such criterions. How interesting would it be for your organization to use Desktop Grid Computing?**

We must have it!	0
We are interested	6
We have some interest	8
We have no interest	9
Total:	**23**

If your organization is using Desktop Grid Computing: which system do you use?

IncreadyBuild	1

5. If your organization were to start using Desktop Grid Computing, how important are these criterions? Prioritize them.

Number of answers	Easy to setup and install	Low purchase price	Low operating cost	Operating-system independent	Secure data transfers	Access to support	Open source	
Top priority	3	6	3	1	5	2	1	21
Priority 2	1	1	9	1	2	3	4	21
Priority 3	9	6	2	0	2	1	1	21
Priority 4	1	4	3	0	7	6	0	21
Priority 5	3	1	3	6	1	5	2	21
Priority 6	0	2	1	6	3	3	6	21
Priority 7	4	1	0	7	1	1	7	21
	21	21	21	21	21	21	21	**147**

Percentage within row/ column	Easy to setup and install	Low purchase price	Low operating cost	Operating-system independent	Secure data transfers	Access to support	Open source	
Top priority	14%	29%	14%	5%	24%	10%	5%	100%
Priority 2	5%	5%	43%	5%	10%	14%	19%	100%
Priority 3	43%	29%	10%	0%	10%	5%	5%	100%
Priority 4	5%	19%	14%	0%	33%	29%	0%	100%
Priority 5	14%	5%	14%	29%	5%	24%	10%	100%
Priority 6	0%	10%	5%	29%	14%	14%	29%	100%
Priority 7	19%	5%	0%	33%	5%	5%	33%	100%
	100%	100%	100%	100%	100%	100%	100%	

List other criterions you see as important.

High business use	1
Low total cost of ownership	1

6. Think freely. What in which areas do you think your organization could use Desktop Grid Computing?

Automated test	10
Compilation	6
Heavy calculations	5
Analytical tools	2
Building releases	2
Server applications	2
Search indexing	1
Simulations	1
Test data generation	1
Total:	**30**