Providing Grid Schedulers with Passive Network Measurements

Yehia El-khatib, Christopher Edwards, Michael Mackay, and Gareth Tyson
Computing Department
Lancaster University
Lancaster, LA1 4WA, UK
{yehia, ce, m.mackay, g.tyson} @ comp.lancs.ac.uk

Abstract—**Grids offer the potential to carry out difficult computing tasks and achieve superior aggregate performance through the utilization of remote resources and transparent collaboration between independent bodies (whether individuals or organizations). Grids are highly complex systems consisting of heterogeneous resources on disparate hosts interconnected across different Autonomous Systems and virtual organizations via a mixture of communication standards. Operating in such environments often proves erratic. Monitoring grid resources allows grid schedulers to adapt to changes in the status of remote resources and the network paths between them. Thus, monitoring the status of the different grid components is critical in order to ensure optimum performance. In this paper we introduce a distributed solution, GridMAP, to collect network and end-host resource measurements, analyze their performance and feed these statistics and predictions back to schedulers. At this stage, we present our implementation of a passive TCP-SYN technique to provide GridMAP with round trip time and throughput measurements and we evaluate our approach against ping and iperf.**

Index Terms—**Availability, Computer network performance, Distributed computing, Measurement, Monitoring.**

I. INTRODUCTION

GRIDS are distributed systems that aggregate a large pool of resources in order to run highly demanding applications and to provide seamless collaboration between virtual organizations. High efficiency is always expected and hence contention on resources is similarly high. However, efficient management of grid resources in such environments is only possible if access to correct and current information about the resources is available. In other words, scheduling decisions can only be as good as the resource information provided to the grid scheduler [1]. Such information is quite difficult to acquire in most grid systems for a number of reasons. In some instances, the middleware is only able to provide pre-defined status information about resources. Such dated information is of no real benefit in improving scheduling during operation. Other middleware solutions often include a Grid Information System (GIS) to gather resource performance information, but dealing with it in many cases is a cumbersome and ineffective process. Therefore, we identify a need to supply grid schedulers with accurate resource information in a simple and scalable manner.

The geographical distribution of resources is one of the fundamental properties of grid systems. The network typically used to connect these resources – i.e. the Internet - is not dedicated to this purpose and is known to be of unpredictable nature[1]. Therefore, there needs to be a means of accessing information about how network connections are performing and how they are expected to perform, otherwise the unpredictable nature of the Internet could seriously affect the performance grids can achieve.

IP networks do not offer the solution to this as they do not readily provide feedback about their practical behavior. It is for that reason that the last three decades have witnessed a continual growth in the number of network monitoring tools, developed for one or more of three core purposes: management, troubleshooting, and pre- and post-deployment probation. Regardless of their purpose, any information retrieved from these tools would typically be analyzed for its significance and reacted to manually by system administrators or users. This approach may be sufficient for traditional applications, but we find it stagnant and hence inappropriate for use in dynamic high performance systems such as grids.

We introduce GridMAP, a distributed grid service which collects network performance and resource availability information and uses it to provide, analyze and predict performance and availability. In this paper, we specifically focus on how GridMAP obtains its network performance information. We make use of a fully passive measurement technique in order to avoid the negative effects of injecting measurement probes into the network. To ensure that accuracy is not compromised, we evaluate the accuracy of our measurements against well known active measurement tools.

The remainder of this paper is arranged as follows. In section II we introduce the GridMAP service which stores and analyzes performance. Then, in section III, we describe how we obtain basic network metrics using passive monitoring techniques. Section IV presents the outcome of the tests we used to evaluate the accuracy of our measurement technique. In section V we review related work, and finally in section VI we present our conclusions and discuss future work.

[1] Network providers and domain administrators are increasingly negating such unpredictability by over-provisioning their networks or deploying QoS techniques, which is by no means pervasive and no guaranteed service can generally be expected [2]. Nevertheless, even grid applications running such environments would benefit from the knowledge of the available resources.

II. THE GRIDMAP SERVICE

This section introduces the GridMAP grid service, one part of our solution. We describe how the GridMAP grid service operates, what it provides and how it is useful to grid schedulers.

Grid schedulers are designed to monitor and control the execution of jobs in grid systems. Such environments typically include a large number of heterogeneous resources residing in different administrative domains. Grid schedulers do not own these resources but are expected to use them efficiently to achieve high performance computing which is, as aforementioned, one of the main goals behind adopting grid systems. For this goal to be realized, grid schedulers need to be provided with accurate status information about system resources, including the underlying network. Foster and Kesselman [3] illustrate the characteristics of grids, adding: "*Fundamental to all of these issues is the need for mechanisms that allow applications to obtain real-time information about system structure and state, use that information to make configuration decisions, and be notified when information changes*". System state including information about both end-host resources (CPU, memory, storage, etc.) and network performance (latency, packet loss rate, etc.) are highly valuable to grid schedulers as they allow them to make more informed decisions on node selection and resource allocation. However, to date there have been few efforts to provide grid schedulers with such status information.

GridMAP (Grid Monitoring, Analysis and Prediction) is a grid service. It is an application that runs as a Web Service and conforms to the WSRF (Web Service Resource Framework) [4] and the OGSI (Open Grid Services Infrastructure) [5] specifications. The GridMAP service provides a set of standard grid service interfaces that allow convenient access for schedulers, enabling them to receive performance information about relevant nodes and connections. Schedulers can then incorporate this information into their job and data allocation processes to automatically adapt to the perceived and foreseeable resource and network performance.

The GridMAP service serves as a distributed repository of performance history. Daemons run on grid nodes to measure resource and network performance and send the metrics on a regular basis to the GridMAP service, which in turn indexes and stores them for analysis and further use. The interaction between the service and daemon is depicted in Fig. 1.

By deploying passive monitoring daemons pervasively on many endpoints, it is possible to exploit the behavior of grid applications to implement a monitoring service using real network traffic. This is discussed in more detail in the following section. Our aim here is to demonstrate the applicability of this form of measurement to provide grid schedulers with information about the resources which they do not control yet must use to efficiently execute jobs.

The measurements collected by GridMAP serve several purposes. First, they act as a "health record" for resources which provides a better insight for troubleshooting, QoS charging and accounting, and verifying SLAs with ISPs. They are also helpful for researchers wishing to evaluate grid applications. Second, the collective data archives are logically available from one source, i.e. the service interface. This allows advanced analysis to be performed on the metrics accumulated from different grid nodes. We plan to employ a pattern recognition scheme (similar to that used in [9]) to provide predictions of future performance.

Being a grid service, GridMAP is also intrinsically distributed. Stored data is automatically replicated across the grid. This decentralized property eliminates having a single point-of-failure, ensuring resilience and high availability. Additionally, this approach makes it possible to afford the demanding computational costs of storing, indexing and analyzing the large amounts of measurement data that is anticipated. Finally, pattern matching techniques can be used for other advanced purposes such as anomaly detection.

This framework is applicable to a wide range of distributed applications, but is particularly important to grids where the requirement for high performance can be hindered by the unpredictable nature of the Internet. GridMAP supplies a sending host with information about other end-hosts and the connection between them. This reduces the maintenance cost for applications and makes them more responsive to changes in the grid in terms of contention over end-to-end network and end-host computational resources. Moreover, this process is performed without the cooperation of intermediate network elements (e.g. routers).

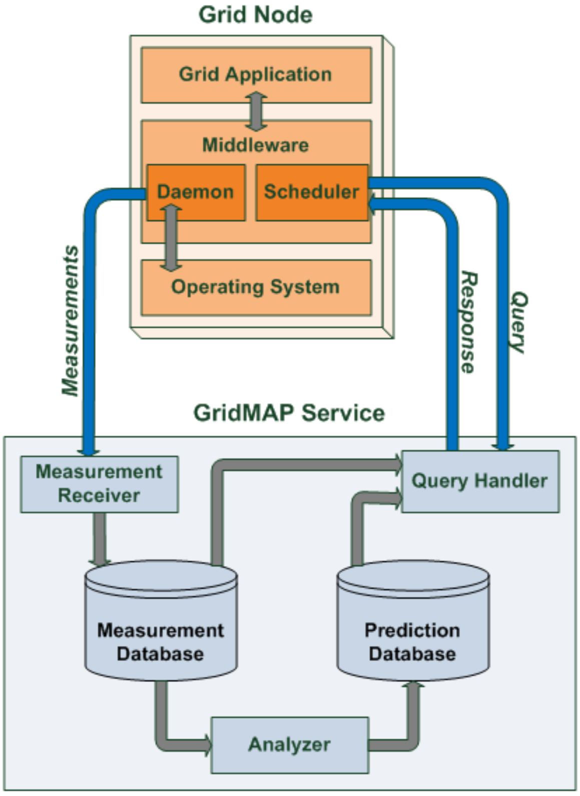

Fig. 1. The interaction between the GridMAP service and daemon.

III. NETWORK MONITORING DAEMON

In this section we explain the technique used to unobtrusively extract network performance information.

There exists different approaches when it comes to measuring network performance. One approach is active measurement where the network is probed to obtain accurate metrics. This obligates the network to accommodate artificial traffic, i.e. the probes, in addition to real traffic thus potentially decreasing the overall performance and affecting the result (see the example of NWS in subsection V.A). Other approaches employ ICMP messaging because it is uses light-weight probes and is relatively easy-to-use. Unfortunately, such approaches are futile in networks where ICMP is disabled or treated differently than TCP traffic; this is not uncommon practice. In contrast, passive measurement approaches attempt to measure network performance without injecting any artificial traffic.

Our motivation is to employ a completely passive technique in order to eliminate any negative effects on the network. We are also equally interested in not compensating accuracy for unobtrusiveness. To attain accurate network measurements using completely passive techniques, we exploit one of the intrinsic properties of grid applications.

Typically, grid nodes constantly exchange data sets, job state, result sets, and control signals during operation. This virtually continuous communication, whether a few kilobytes or hundreds of gigabytes large, is carried out using TCP [6]. Our technique exploits such frequent TCP interactions to extract basic network metrics. This is done by monitoring TCP three-way handshakes, which is not a new approach in itself. The novelty of our technique lies in the context in which it is applied. Grids provide us with an abundance of natural TCP connection that can be exploited using this technique to provide accurate measurements. This is not feasible in other systems which is the reason why some other TCP-based measurement techniques are supplemented with active probes. This makes these techniques problematic (see subsection V.B).

Furthermore, our technique is essentially quite trivial but because so it is easily decentralized on all nodes in the grid as part of the middleware. This provides a powerful viewpoint which results in realistic data instead of estimations, as in the case of tomographical measurements (see subsection V.A).

We have developed a daemon that uses the pcap library to capture packet headers from a network interface to calculate round trip time (RTT) and throughput. Connection setup is used to calculate RTT as the delay between sending a SYN packet and receiving its corresponding SYN-ACK packet. Obviously, this delay consists of the two-way propagation delay as well as any processing delay generated at the remote host but we assume the latter to be negligible compared to the former in most cases. As each TCP connection terminates, data throughput is calculated as the total amount of application-level data divided by the total duration of the connection. Both RTT and throughput measurements are time-stamped and locally cached to be periodically submitted to the service.

Monitoring the traffic generated by the running grid application thus becomes an automatic process that continues as traffic naturally passes through the node. By using real application data, no artificial traffic is injected into the network and hence no disruption is caused to traffic already traversing the network. The metrics calculated by our daemon therefore directly reflect the experience of TCP traffic in the network. In addition, this prevents measurements from being mistaken for threats such as TCP-SYN flooding or Denial-of-Service attacks. Furthermore, this overcomes the possibilities of measurement traffic following different routing paths than data, receiving different prioritization, or not going through at all (as can be the case with ICMP in many instances). Moreover, the daemon works independently with no need for peer coordination. Finally, our technique does not rely on IP or NetFlow accounting and hence does not depend on whether routers run accounting schemes or whether such information is available.

IV. EVALUATION

To evaluate the accuracy of the measurements supplied to the GridMAP service using our passive measurement technique, we conducted a series of five tests over different distances. In the first test, the source and destination hosts are connected locally by Ethernet. In the second test, the destination is connected via DSL and is 4 hops away from the source. The third test is carried out on a 12-hop connection from Lancaster to Oxford, the fourth on a 15-hop connection from Lancaster to Munich, Germany, and the final test on a 17-hop connection from Innsbruck, Austria to Lancaster.

The setup in each test is identical: we generate TCP traffic using iperf [7] for 34 different transmission durations (ranging from 1 to 500 seconds). In every test, our daemon sits on the sending node while the destination node acknowledges received packets. We compare our RTT measurements to those of ping and our throughput measurements to those of iperf.

A. *Round Trip Time*

We set up these experiments such that 5 ping repetitions are triggered with each iperf probe. At the same time, we used the TCP handshake of the iperf probe to measure RTT. We then compared our results to the minimum and mean of the ping repetitions but left out the maximum values since they were much larger than the mean values[2]. Fig. 2 depicts the ratios of our measurements to the minimum and mean ping values against the test duration. Note that during the test with the DSL connection, ping packets did not get through (due to disabled ICMP messaging) and hence we could not calculate the ratios.

We find our technique to be consistently accurate with the RTT results obtained from ping. Fig. 2 shows that the vast majority of our RTT measurements are almost identical to the

---

[2] In some instances, the maximum RTT was up to 200% more than the mean RTT. This is because ICMP packets are often treated as low priority traffic.

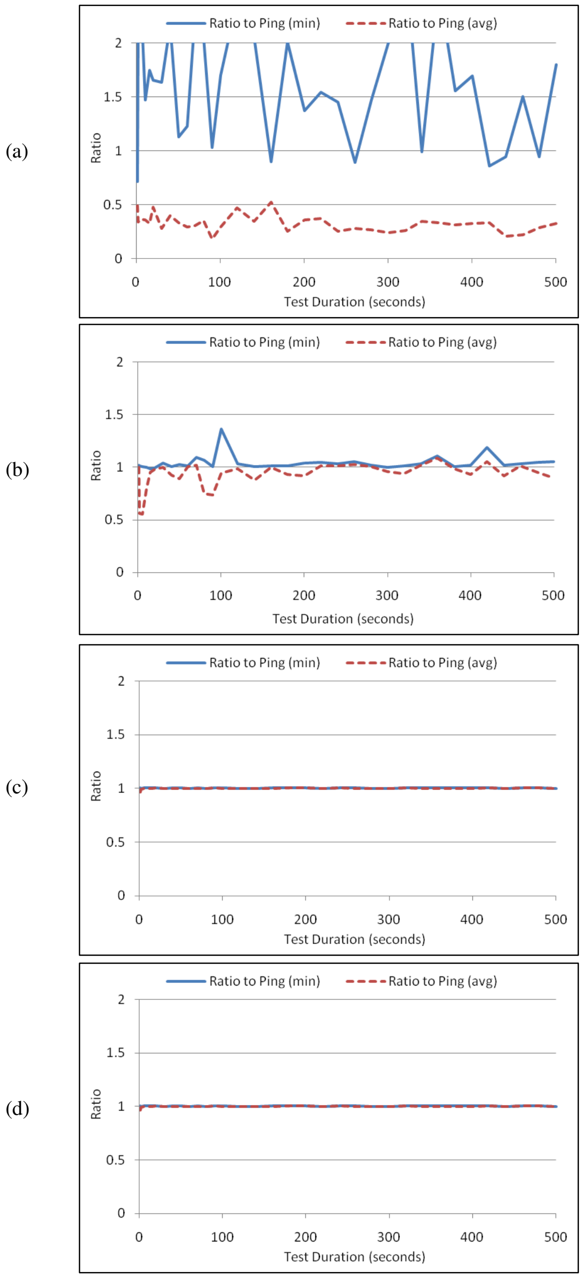

Fig. 2. The ratio of our RTT measurements to the minimum and mean ping results for: (a) Ethernet, (b) Oxford, (c) Munich and (d) Innsbruck.

minimum ping values. Our measurements are also close to the mean ping values. The mean ping values for the Oxford connection (Fig. 2(b)) displayed more irregularity than the others: the standard deviations for the Oxford, Munich and Innsbruck connections were 1.923, 0.297, and 0.276 respectively. We believe this to be because some routers flag ICMP ping packets as low-priority. By avoiding ICMP, our measurement approach thus provides more reliable data. On average, our measurements were 1.55% away from the minimum ping values and 2.33% away from the mean ping values.

Fig. 2(a) depicts the RTT measurements of the Ethernet connection which are almost double the minimum ping values. This is because the propagation delay is so small (around 0.57 ms) that the processing delay can no longer be neglected. However, this is not a significant limitation as the daemon's

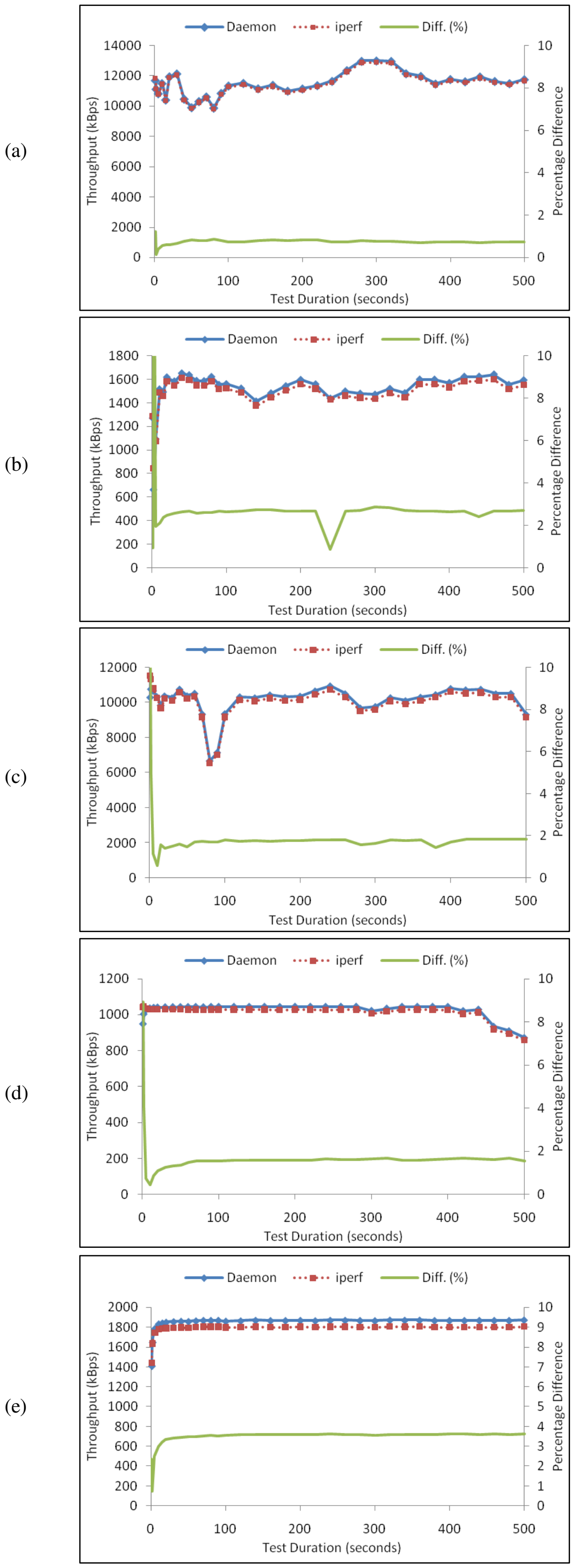

Fig. 3. The throughput values in obtained by our daemon and iperf, and the percentage difference for the connections: (a) Ethernet, (b) DSL, (c) Oxford, (d) Munich, and (e) Innsbruck.

utility lies in its ability to monitor Internet scale interactions. Connections within the same domain rarely require such real-time measurements and are not the main focus of our research.

B. *Throughput*

For the throughput evaluation, we created 34 TCP connections lasting between 1 and 500 seconds in each case as described above. In Fig. 3 we compare our calculated throughput against that returned by the iperf client. Because the different connections achieve significantly different throughputs, Fig. 3 also graphs the percentage difference between the measurements obtained by the two methods.

We establish that our throughput measurements are consistently accurate compared to iperf for all TCP transfers. Overall, our throughput measurements are within 2.20% of the measurements obtained by iperf. We did notice however that for connections that last 2 seconds or less, our throughput measurements are around 10% away from those of iperf. We believe this is due to inaccuracy of the estimations made by iperf for short duration flows.

V. RELATED WORK

In this section, we highlight a selection of the body of work that is relevant to ours. We discuss measurement frameworks that allow distributed systems (such as grids) to be informed about changes in their networking environment. We then discuss network measurement techniques that work in a similar fashion to our daemon.

A. *Measurement Frameworks for Grids*

With the birth of the metacomputing paradigm, recent years have seen an increased interest in monitoring grids. While the focus is slightly different in each case, the common aim is to gauge the performance delivered to the application. Here we focus on a number of efforts that are closely related to the framework we present.

The Network Weather Service (NWS) [8] is a dynamic system that provides schedulers with regular network and system performance measurements and forecasts. Although the NWS forecasts for the availability of computational resources are fairly accurate, the predictions provided for network performance are not [9] [10] [11]. This inaccuracy stems from two main characteristics of NWS. First, the default network probe used is not sufficient to force TCP beyond its slow-start phase. To correct this, probe sizes could be increased but this would also serve to amplify the incurred network overheads. Second, NWS measurement is tomography-based, relying on information collected by *sensors* that report back on the metrics observed in their neighboring vicinity. The data collected by such a technique is at best a close estimate of the performance of the surrounding nodes, but is still merely an estimate.

REM [12] is a framework that aims to make sense of different performance indicators to identify unexpected network behavior and react by triggering automated network analysis. Although the motivations are different, both REM and GridMAP regard measurement as an automated process rather than an isolated and manual activity.

Beyond this, several hybrid measurement frameworks, like [13] and [14], have also been defined. These integrate both active and passive techniques where switching between the two occurs in response to high network utilization. This is a good compromise to reach a middle-ground of relatively low intrusiveness and high accuracy. However, active probing of over-utilized paths still increases the possibility of these links becoming bottlenecks.

Other related efforts in the literature include:

- Flexmon [15] - a framework that uses periodic probes to measure and record network performance metrics;
- eTOP [16] - an infrastructure that triggers active probes to inform users about the health of end-to-end paths in the system;
- perfSONAR [17] - a service-oriented, tomography-based framework that employs domain-specific *Measurement Points*;
- [18] - a tomography-based technique that estimates network distances using ICMP measurements;
- [19] - a passive technique that uses TCP traces to determine available bandwidth, but requires kernel modifications;
- [20] - a system that logs all application, OS, device, and network events and then compares logs to identify bottlenecks.

More general work in this area includes the standardization efforts within the IETF, such as the Real-time Traffic Flow Measurement Working Group which resulted in the NeTraMet architecture [21].

B. *Network Measurement Using TCP*

Using TCP handshakes to extract network metrics is a technique that has received extensive attention in network measurement literature (see for instance [22], [23], [24], and [25]) and has proven to be reasonably accurate for measuring the properties of TCP connections. This technique avoids the disadvantages of ICMP-based probes and obtains a true reflection of the treatment TCP packets receive in the network. However, this has traditionally been implemented using synthetic SYN packets; an approach that has its own disadvantages. The main drawback is that artificial TCP handshakes can be mistaken for threatening attacks such as TCP-SYN floods [23]. Furthermore, such techniques require a list of servers to which measurement messages are sent *a priori* which is not suitable for dynamic distributed environments such as grids. Such overheads thus hinder the use of such technique on any large scale basis.

Our technique, however, monitors grid applications that naturally provide a sufficient number of TCP connections to use for measurement. This negates the need to create artificial TCP connections and to compose list of destination nodes.

VI. CONCLUSIONS & FUTURE WORK

In this paper we introduce GridMAP, a decentralized solution to provide grid schedulers with accurate performance information about the resources in the grid, including the network. The solution is made up of two parts. The first is a grid service that collects and stores measurements of network performance and end-host resource availability. The service allows schedulers to access this information to automatically adapt to perceived and foreseeable resource and network performance. The second part of the solution is a daemon that captures packet headers from the network interface card and then calculates different network metrics. These are sent on a regular basis to the GridMAP grid service. The performance measurements are taken in an entirely passive fashion by exploiting the persistent TCP transfers common in grids. We tested the network measurements provided by our daemon against those from ping and iperf and concluded that our measurements are reasonably accurate.

Beside our ongoing work to implement the GridMAP service, there are a number of ways in which we plan to extend the work presented here. First, we will expand the number of metrics produced by the measurement scheme. Although RTT and throughput are certainly relevant to a large number of applications, they may not be sufficient for some. For instance, one-way delay variation is central to the performance of virtualization applications. Second, we have only tested the technique presented here against active tools, i.e. ping and iperf. We plan to test it against more active and passive techniques. Finally, we will extend the measuring daemon to take snapshots of the availability of local resources.

ACKNOWLEDGMENT

The authors are grateful to the European Commission for funding the EC-GIN project [26] under which this work was completed. The authors appreciate the help of Thomas Bocek, Tiejun Ma and Michael Welzl in setting up remote tests.

REFERENCES

[1] Jennifer M. Schopf. "Ten Actions When Grid Scheduling: The User as a Grid Scheduler". In J. Nabrzyski, J. Schopf, J. Weglarz. "Grid Resource Management: State of the Art and Future Trends", chapter 1, pp. 15-23, Kluwer Academic Publishers, 2004. ISBN:1-4020-7575-8

[2] Christopher Edwards, Michael Mackay (Ed), "JANET QoS Technical Guide", JANET (UK), June 2008. http://www.ja.net/documents/publications/technical-guides/qos-tg.pdf.

[3] Ian Foster, Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit". In Proc. Workshop on Environments and Tools for Parallel Scientific Computing, SIAM, Lyon, France, August 1996.

[4] OASIS, Web Service Resource Framework (WSRF) 1.2. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

[5] Steven Tuecke, Karl Czajkowski, Ian Foster, Jeffrey Frey, Steve Graham, Carl Kesselman, Tom Maquire, Thomas Sandholm, David Snelling, Peter Vanderbilt (editors), "GFD-R-P.15 - Open Grid Services Infrastructure (OGSI) version 1.0". http://www.ggf.org/documents/GFD.15.pdf

[6] Yehia El-khatib, Christopher Edwards, "A Survey-based Study of Grid Traffic". In Proc. ICST/ACM International Conference on Networks for Grid Applications (GridNets 2007). Lyon, October 2007.

[7] NLANR/DAST: Iperf - TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/Projects/iperf

[8] Rich Wolski, "Dynamically Forecasting Network Performance Using the Network Weather Service". In Proc. 6th International Symposium on High Performance Distributed Computing (HPDC '97), Portland, OR, USA, August 1997.

[9] Sudharshan Vazhkudai, Jennifer M. Schopf, Ian Foster. "Predicting the Performance of Wide-Area Data Transfers". In Proc. 16th International Parallel and Distributed Processing Symposium (IPDPS 2002), Fort Lauderdale, FL, USA, April 2002.

[10] Pascale Primet, Robert Harakaly, Franck Bonnassieux. "Experiment of the NWS Network Forecasting for Grid Networking". In Proc. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2002), Berlin, Germany, June 2002.

[11] Muhammad Murtaza Yousaf, Michael Welzl, Malik Junaid. "Fog in the Network Weather Service: A Case for Novel Approaches". In Proc. MetroGrid Workshop: 'Metrology for Grid Networks' within the framework of the ICST/ACM International Conference on Networks for Grid Applications (GridNets 2007), Lyon, France, October 2007.

[12] Mark Allman, Vern Paxson, "A Reactive Measurement Framework". In Proc. 9th Passive and Active Measurement Conference (PAM 2008), Cleveland, OH, USA, April 2008.

[13] Bruce Lowekamp. "Combining Active and Passive Network Measurements to Build Scalable Monitoring Systems on the Grid". ACM SIGMETRICS Performance Evaluation Review, volume 30, issue 4 (March 2003), Special issue: Special section on Grid Computing: pp 19-26. ISSN: 0163-5999

[14] Muhammad Murtaza Yousaf, Michael Welzl, "A Reliable Network Measurement and Prediction Architecture for Grid Scheduling". In Proc. IEEE/IFIP International Workshop on Autonomic Grid Networking and Management (AGNM'05), Barcelona, Spain, October 2005.

[15] David Johnson, Daniel Gebhardt, Jay Lepreau, "Towards a High Quality Path-oriented Network Measurement and Storage System". In Proc. 9th Passive and Active Measurement Conference (PAM 2008), Cleveland, OH, April 2008.

[16] eTOP: End-TO-end Performance measurement infrastructure on KOREN and APAN links. http://etop.apan.net, http://master.apan.net/meetings/busan03/measurement/koren.ppt

[17] Andreas Hanemann, Jeff W. Boote, Eric L. Boyd, Jérome Durand, Loukik Kudarimoti, Roman Łapacz, D. Martin Swany, Szymon Trocha, Jason Zurawski, "PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring". In Proc. 3rd International Conference on Service Oriented Computing (ICSOC 2005), Amsterdam, The Netherlands, December 2005.

[18] Yuval Shavitt, Xiaodong Sun, Avishai Wool, Bulent Yener, "Computing the Unmeasured: An Algebraic Approach to Internet Mapping". In IEEE Journal on Selected Areas in Communications, volume 22, issue 1: pp.67-78, January 2004.

[19] Marcia Zangrilli, Bruce B. Lowekamp, "Transparent Optimization of Grid Server Selection With Real-Time Passive Network Measurements". In Proc. 3rd International Workshop on Networks for Grid Applications (GridNets 2006), San Jose, CA, USA, October 2006.

[20] Dan Gunter, Brian Tierney, Keith Jackson, Jason Lee, Martin Stoufer, "Dynamic Monitoring of High-Performance Distributed Applications". In Proc. 11th IEEE International Symposium on High Performance Distributed Computing (HPDC11), pp: 163-170, Edinburgh, United Kingdom, July 2002.

[21] CAIDA– NeTraMet. http://www.caida.org/tools/measurement/netramet/

[22] Stefan Savage, "Sting: a TCP-based Network Measurement Tool". In Proc. 2nd USENIX Symposium on Internet Technologies & Systems, Boulder, CO, USA, October 1999.

[23] Martin Horneffer, "Assessing Internet Performance Metrics Using Large-Scale TCP-SYN based Measurements". In Proc. First Passive and Active Measurement Workshop (PAM-2000), Hamilton, New Zealand, April 2000. http://www.pamconf.org/2000/pdf_papers/P010.pdf

[24] Synack (IEPM). http://www-iepm.slac.stanford.edu/tools/synack

[25] Hao Jiang, Constantinos Dovrolis, "Passive Estimation of TCP Round-trip Times". In ACM SIGCOMM Computer Communication Review, volume 32, issue 3 (July 2002): pp. 75-88, 2002.

[26] Europe-China Grid InterNetworking (EC-GIN), a Framework 6 STREP project, Action Line: IST-2005-2.6.5.1.c Grids China, Project Reference (Contract Number): 045256. http://www.ec-gin.eu/