Fog in the Network Weather Service: A Case for Novel Approaches

Muhammad Murtaza Yousaf
Murtaza.Yousaf@uibk.ac.at

Michael Welzl
Michael.Welzl@uibk.ac.at

Malik Muhammad Junaid
Malik.Junaid@uibk.ac.at

University of Innsbruck, Austria

ABSTRACT

A very large amount of data must be used to reasonably measure the available network bandwidth in a Grid by simply checking the time that it takes to send it across the network with TCP. The Network Weather Service (NWS) is the most common tool for obtaining transfer delay predictions from network measurements in Grids. We show that, in simple tests in a real Grid, the results that it obtains are not good enough or require heavily loading the network. The point of this study is to illustrate the need for more sophisticated and appropriately designed network measurement tools.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Monitoring

General Terms

Measurement, Experimentation, Verification

Keywords

Bandwidth estimation

1. INTRODUCTION

Knowledge of existing resources and their capacities in a grid is of utmost importance for Grid schedulers and resource brokers. Grid resources include CPU power, disk storage, memory, and (very much ignored) network resources. Grid applications vary greatly [22] with respect to data transfers, flow dependencies, computational requirements and many other parameters. There is a long list of data intensive Grid applications which rely heavily on network resources among different Grid sites. One such example is Large Hadron Collider (LHC) Computing Grid project [1] at CERN, which is expected to produce and distribute around 15 Petabytes of data every year for analysis. Scheduling of large data flows for such data intensive applications is highly dependent on network path characteristics, mainly network bandwidth.

Therefore, for computationally intensive applications, resource broker or scheduler needs to have a comprehensive knowledge of network properties to fulfill service level agreements, ensure quality of service, and to make clean choices for advance reservation. The dependency of scheduler and resource broker-like components on network properties, calls for as accurate as possible estimation and prediction of network path properties.

Although many tools and approaches have been proposed for the estimation of network bandwidth, the Network Weather Service (NWS) [2] is still the most widely used tool in the Grid community. The main reasons behind this are:

1. Along with network bandwidth and latency, it also predicts CPU availability (for already running processes and newly-started processes) and free disk space.
2. It is possible to install it into the globus installation tree.
3. It is possible to package NWS using Grid Packaging Technology (GPT).

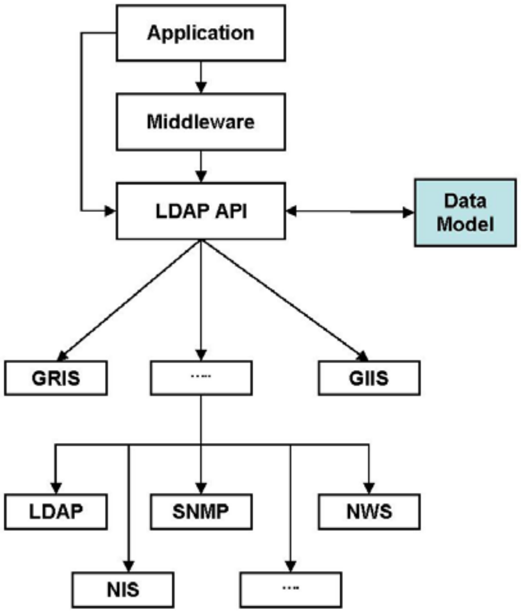

Figure 1. A generic MDS architecture

Grid Resource Discovery is generally performed through querying the Grid Index Information Service (GIIS), which is a part of a General Information Infrastructure known as Metacomputing Directory Service (MDS). MDS collects the information from different information providers, which include LDAP, SNMP, NIS and NWS. NWS is used as a source because it provides information about many properties related to machines as well as network properties like bandwidth and latency. A graphical representation of the MDS hierarchy and sources of information to GIIS and GRIS is presented in Figure 1.

The work described in this paper is partially supported by the Higher Education Commission (HEC) of Pakistan under the doctoral fellowship program for Austria, the European Union through the FP6-IST-045256 project EC-GIN, and the Tyrolean Science Fund

2. Bandwidth Estimation in NWS

2.1 Architecture of Network Weather Service

The Network Weather Service [4] is a distributed system designed to forecast the performance of computational and network resources and make them available for higher level applications.

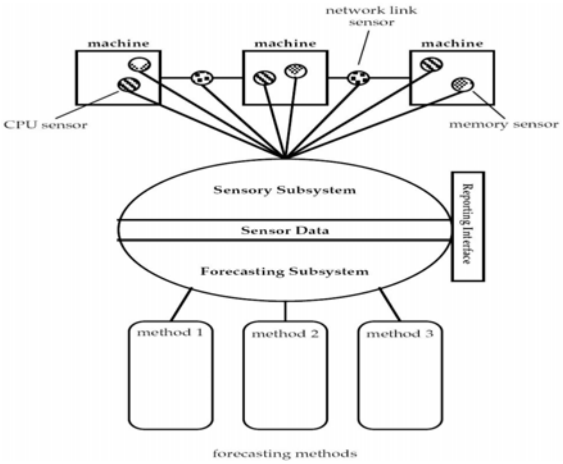

Figure 2. NWS Architecture (figure taken from[3]).

Among the basic components of NWS are a name server (to bind and manage all components), one or more memory servers (persistent storage which may be located at one or more machines), and sensors (for the measurements). The architecture of the system is depicted in Figure 2 as explained in [3].

2.2 Bandwidth and Latency Measurement by NWS

For bandwidth and latency measurements, sensors probe periodically. Suppose we are interested in network measurements from host 'A' to host 'B', then probing mechanism works in this way:

'A' sends a message of 4 bytes via TCP to 'B', and receives back the reply from 'B'. Now, latency is calculated by dividing the round trip time by 2.

$$latency = \frac{t_{round-trip}}{2}$$

Where

$t_{round-trip}$ is the round-trip time.

After latency calculation, 'A' sends a large message via TCP (the default value is 64K byte) to 'B' and the bandwidth is calculated in this fashion:

$$bandwidth = \frac{S}{\Delta t - latency}$$

Where

S is the message size, and

Δt is the message transfer time

Here, latency is subtracted from the message transfer time to exclude the overhead to initiate the TCP/IP communication stream [3].

2.3 Back-of-the-Envelop Calculation to Saturate a Link

NWS calculates throughput by dividing the amount of data that it managed to transfer by the time that it took. In order to saturate the network, which is indispensable for properly measuring the available bandwidth, the amount of data that is sent would have to be a function of the *bandwidth x delay* product of the end-to-end path. If the amount of data is much smaller, what is measured is a side effect of TCP behavior but not related to the network (that is, it would not matter if the capacity at the bottleneck is 10, 100 or 500 Mbit/s as explained in Figure 3).

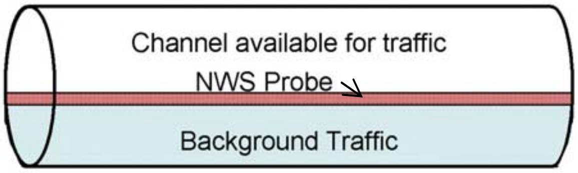

Figure 3. NWS probes in a large bandwidth x delay product end-to-end path

In fact, for very small amounts of data, it is quite likely that TCP will not even finish the slow start phase before the transmission is over. This is definitely the case for the default value of 64 Kbyte mentioned in [5], as this is also the usual default threshold in most operating systems for ending the slow start phase (*sshtresh*). Recently, it has become common to increase this value, which lets TCP operate better in high *bandwidth x delay* environments, but also means an even longer phase that would have be measured by NWS (instead of quickly probing for the available bandwidth which is the intention).

In slow start, a TCP sender begins by sending a single packet (or more, according to RFC 2581 [6] for simplicity, we assume only one packet at this point) and waiting for the corresponding ACK from the receiver. Then, it sends one more packets for each incoming ACK until *ssthresh* is reached. The amount of data sent in slow start can therefore easily be calculated; it is shown in Table 1 for an example Round Trip Time (RTT) of 500ms, packet size of 1500 bytes, and initial congestion window (*cwnd*) of 1.

From Table 1, we can see that it takes 8 seconds till the TCP flow reaches 100MB/s. At this point, the congestion window reaches 64 Kbytes, which means that slow start would end; however the amount of data that was already sent is much more - clearly, a sender which sends only 64 Kbytes (as in the case of NWS) would not even reach this stage.

Table 1. TCP Slow Start

Time (ms)	cwnd	Used Bandwidth (KB/s)	Already Sent (KB)
0	1	3.0	0.0

500	2	6.0	1.5
1000	4	12.0	4.5
1500	8	24.0	10.5
2000	16	48.0	22.5
2500	32	96.0	46.5
3000	64	192.0	94.5
3500	128	384.0	190.5
4000	256	768.0	382.5
4500	512	1536.0	766.0
5000	1024	3072.0	1534.5
5500	2048	6144.0	3070.5
6000	4096	12288.0	6142.5
6500	8192	24576.0	12286.5
7000	16384	49152.0	24574.5
7500	32768	98304.0	49150.5
8000	65536	196608.0	98302.5

To make things worse, it is common for the receiver to acknowledge only every other packet (as is recommended in the RFC 1122 [7]), which means that reaching *ssthresh* takes twice as long. The specification of a mechanism that corrects this error, Appropriate Byte Counting [8], is still experimental, and hence it cannot be expected to be widely deployed.

Let us now consider the case where the amount of data is enough to leave slow start and enter the congestion avoidance phase. RFC 3649 [9] states:

"The congestion control mechanisms of the current Standard TCP constrain the congestion windows that can be achieved by TCP in realistic environments. For example, for a Standard TCP connection with 1500-byte packets and a 100 ms round-trip time, achieving a steady-state throughput of 10 Gbps would require an average congestion window of 83,333 segments, and a packet drop rate of at most one congestion event every 5,000,000,000 packets (or equivalently, at most one congestion event every 1 2/3 hours). This is widely acknowledged as an unrealistic constraint". From this discussion we can conclude that its underlying dynamics make TCP a poor vehicle for the kind of test that NWS carries out.

3. Results and Analysis

3.1 Experimental Setup

We have performed experiments on the Austrian Grid [10] sites listed in Table 2, which are located at geographically remote locations across Austria.

The first two sites of the infrastructure are co-located at the University of Innsbruck. There is a dedicated Gigabit network between them. The other three Grid sites are also part of the Austrian Grid and are connected through broadband Internet connections. The overall setup is shown in Figure 4.

Table 2. Grid sites used from the Austrian Grid

Site	Master
ALTIX-UIBK	altix1.uibk.ac.at
dps-prod	karwendel.dps.uibk.ac.at
JKU	Hydra.gup.uni-linz.ac.at
SBG	schafberg.sbg.coma.ac.at
FHV	grid.labs.fhv.at

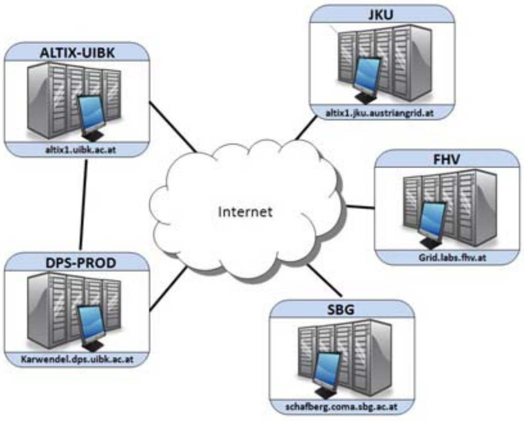

Figure 4. Experimental setup – Austrian Grid.

We started the NWS Name Server (*nws_nameserver*) and Memory Host (*nws_memory*) at dps-prod. Then we started Sensor Hosts (*nws_sensor*) and started the *tcpMessageMonitor* activity on all grid sites to measure the *bandwidth* among these sites.

3.2 Measurements (with default parameters)

In our first experiment, we measured the bandwidth using default values (64k, 32k, 16k) of the *tcpMessageMonitor* activity, which means that NWS used four 16kB messages to send a total of 64kB of data using a socket buffer size of 32kB. The results are shown in Table 3.

Unfortunately we could not find some free ports at FHV, consequently the Table 3 does not have any measurements from other sites to FHV.

3.3 Parameter Values' Impact

To check the impact of parameter values on the measurements (which we believed to be significant) we gathered measurements from SBG to JKU and ALTIX-UIBK to dps-prod. SBG → JKU path was selected to investigate a path with internet as backbone and ALTIX-UIBK → dps-prod was selected to examine a path with a Gigabit link.

Table 3. Bandwidth (Mb/s) among all grid sites

Source Site	Destination Site			
	ALTIX-UIBK	dps-prod	JKU	SBG
ALTIX-UIBK		222.077	6.481	14.166
dps-prod	228.004		6.483	14.152
JKU	6.394	6.492		9.925
SBG	21.083	21.869	10.074	
FHV	5.986	5.986	3.710	4.819

We started with parameter values (100k, 100k, and 100k) close to default values of NWS and gradually increased the data size. We kept on increasing the probe size until the measurements became stable, which was the stage when we managed to actually saturate the link. The results for SBG → JKU are depicted in Figure 5, and the relationship of probe size with measurements for ALTIX-UIBK → dps-prod is shown in Figure 6. In both graphs, each value is actually representing the average of 10 measurements for a particular set of parameter values.

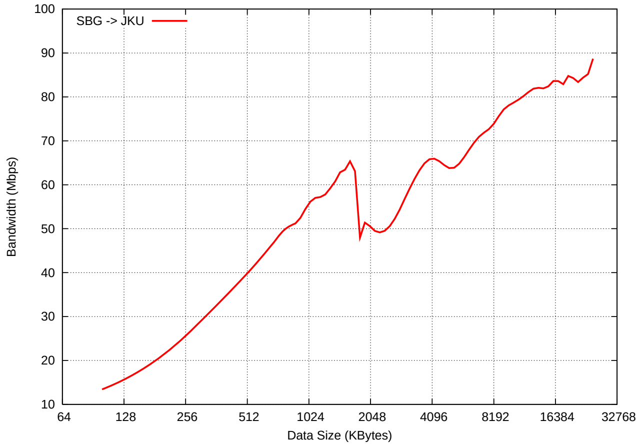

Figure 5. Probe size vs Measurement. SBG → JKU

.

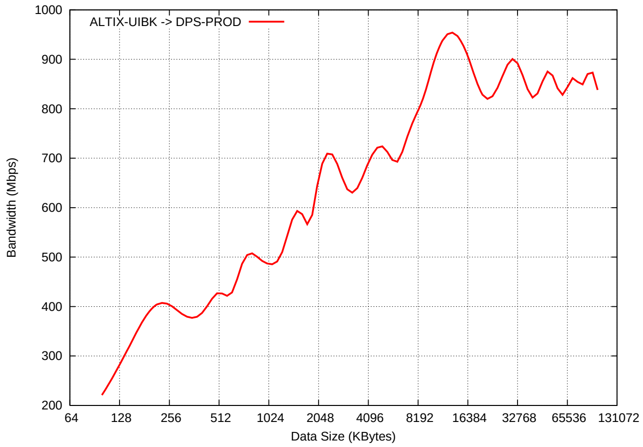

Figure 6. Probe size vs Measurement. ALTIX-UIBK → dps-prod

We can observe that the measured bandwidth increases with larger probe size. A small dip in the measurements in Figure 5 is most probably because of a congestion event at that particular time. So, at that time the available bandwidth was very small and even a smaller probe size was sufficient to saturate the link.

3.4 Analysis

If we compare the final measurements achieved in the previous section with the ones in table 3 with a default probe size, a significant difference is notable.

For SBG → JKU, we needed more than 20MB data to really saturate the link resulting in a bandwidth measurement of more than 80 Mb/s which was measured to be around 10 Mb/s in case of the default 64K message size. This difference is more interesting in case of an end-to-end path where *bandwidth x delay* product is large. We can see this in our attempt to saturate the link for ALTIX-UIBK → dps-prod. For this Gigabit link, the overall probe size to measure a relatively accurate value of bandwidth was around 100 MB, ultimately providing us with a bandwidth in the range of 900 Mb/s. We followed the approach used in section 3.3 to find the probe size for more accurate measurements by NWS methodology and found significant differences as compared to values shown in Table 3.

It is possible to simply configure NWS to carry out very long lasting measurements, sending hundreds of megabytes from one side of the network to the other. Clearly, stressing the network with such huge amounts of otherwise useless measurement data is also not desirable. Moreover, a measurement that is obtained in this way will only be useful for predictions for files exceeding a certain minimum size (or otherwise TCP's slow start behavior will again predominate).

Clearly, given the many efforts that were made to carry out highly sophisticated network measurements which would efficiently yield more useful results, it is a poor choice to predict file transfer delays with a simplistic method that does not take the behavior of TCP into account. Since the Grid has the additional advantage of enabling distributed measurements from a set of end systems which can be expected to remain available for a sustained duration, we believe that there is a clear need for novel Grid-specific methods that would exploit this fact.

4. Related Work

In [21] authors studied the forecasting mechanism of NWS in terms of the stability of forecasts and the confidence level of the forecasts. The authors found that the confidence level of the forecast on well provisioned links was not very high (41%) with a prediction error of almost 20%. On the other hand, on heavily loaded links, the forecasting error was much smaller, being nearly equal to 5%, whereas the confidence level approached to 84%. The authors also proposed a simple model to eliminate the effect of *slow-start*.

Bandwidth estimation and throughput prediction is of interest for many reasons, including optimization of end-to-end transport performance, QoS assurance, and optimal selection of grid sites for a grid scheduler. A detailed survey of bandwidth estimation tools is presented in [11] with a discussion of the underlying techniques and methodologies used in those tools. A more recent survey can be found in [12], conducted on the similar basis as it was done in [11].

Many end-to-end bandwidth estimation tools have been designed, for example Nettimer [13], Pathrate [14], Pathload [15] and many more. A detailed list can be found at CAIDA's tool page [16] as well as at ICIR's page [17] of tools for bandwidth estimation. A more detailed list is available at [18].

Among the many methodologies that have been proposed and used in the past for bandwidth estimation, packet pair [19] and its variations (including packet triplets [20] and packet trains), which use the packet dispersion for characteristics extraction, have performed well. Nevertheless, these methods are still not included in practical online measurement tools such as NWS.

5. Conclusion

We have shown that, under realistic conditions, a very large amount of data must be used to reasonably measure the available network bandwidth in a Grid by simply checking the time that it takes to send it across the network with TCP. This is due to the way that the TCP protocol operates. If the data size is too small, as it will normally be the case with the default parameters of the Network Weather Service (NWS), TCP does not have enough time to saturate the link, and may in fact not even reach the end of its initial slow start phase. What is measured is then an artifact of TCP behavior and has very little to do with the environment conditions. This means that a transfer delay prediction from NWS may be correct if the file to be transmitted is exactly as large as the measurement probe, but if the file is 10 times as large, for example, it is completely wrong to assume that its transfer will take 10 times as long.

Accuracy is not the only problem with NWS – it also does not provide enough information about the network. We illustrate this with a simple example. Due to its distributed nature, a Grid application can adversely influence the network performance of itself if some hosts send too much. Similarly, Grid application 1 can disturb Grid application 2 if they share the same hosts. This problem is shown in Figure 7, where the two hosts A and B are connected to host C via a single link (the broken line in the figure). If both A and B send at a high rate, they can reduce the throughput of each other, thereby degrading the performance of the application. If this fact was known, a Grid scheduler could try to circumvent this problem by relocating parts appropriately – but in practice, these instances remain uninformed as there is no means available to detect such a problem.

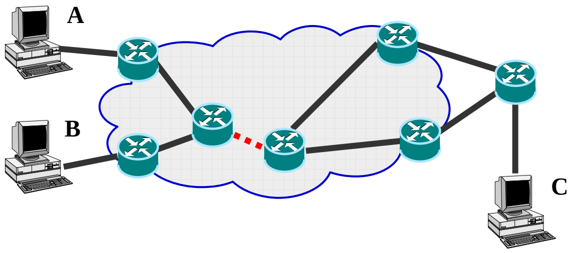

Figure 7. Traffic from A and B to C shares a bottleneck

In closing, we would like to remind the reader that the intention behind our efforts was not to bash NWS, but to show that the simple network measurement that this tool carries out is not good enough in a realistic Grid setting with high capacity links. Despite its name, the network weather service is much more than just a simple network measurement tool. It encompasses an architecture comprising a multitude of elements for functions such as monitoring (e.g. of the available CPU power) and storage, and contains a number of prediction algorithms among which the most suitable one is automatically chosen. Its extensible design as well as the fact that it is already widely used actually makes NWS an ideal system for integrating a new network measurement method, which would simply be another type of sensor in the system from the perspective of NWS.

6. ACKNOWLEDGMENTS

We would like to thank Stephen Hirschmann for the slow start calculations.

7. REFERENCES

[1] I. Bird et al. LHC computing grid technical design report. Technical Report CERN-LHCC-2005-024, June 2005.

[2] R. Wolski, L. Miller, G. Obertelli, M. Swany. Performance Information Services for Computational Grids. In Resource Management for Grid Computing, Nabrzyski, J., Schopf, J., and Weglarz, J., editors, Kluwer Publishers, Fall, 2003.

[3] R. Wolski. Dynamically Forecasting Network Performance Using the Network Weather Service. Journal of Cluster Computing, Volume 1, pp. 119-132, January, 1998.

[4] B. Gaidioz, R. Wolski, and B. Tourancheau. Synchronizing Network Probes to avoid Measurement Intrusiveness with the Network Weather Service. Proceedings of 9th IEEE High-performance Distributed Computing Conference, August, 2000, pp. 147-154.

[5] R. Wolski, N. Spring, and J. Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. Journal of Future Generation Computing Systems,Volume 15, Numbers 5-6, pp. 757-768, October, 1999.

[6] M. Allman, V. Paxson, W. Stevens. TCP Congestion Control. RFC 2581. April 1999.

[7] R. Braden. Requirements for Internet Hosts – Communication Layers. Oct 1989. IETF RFC 1122.

[8] M. Allman. TCP Congestion Control with Appropriate Byte Counting (ABC). February 2003.

[9] S. Floyd. HighSpeed TCP for Large Congestion Windows. December 2003.

[10] The Austrian Grid Consortium. http://www.austriangrid.at.

[11] R. S. Prasad, M. Murray, C. Dovrolis, K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. Published in IEEE Network, November – December 2003.

[12] A. Botta, A. Pescape, g.. Ventre. On the performance of bandwidth estimation tools. Systems Communications, 2005. Proceedings Volume , Issue , 14-17 Aug. 2005. Page(s): 287 – 292

[13] K. Lai, M. Baker. Nettimer: A Tool for Measureing Bottleneck Link Bandwidth. In Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems, San Francisco, California. March 2001.

[14] C. Dovrolis, P. Ramanathan, D. Moore. Packet-dispersion techniques and a capacity-estimation methodology. IEEE/ACM Transactions on Networking (TON). Volume 12 , Issue 6, December 2004.

[15] M. Jain and C. Dovrolis, “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” in Proceedings of ACM SIGCOMM, Aug. 2002, pp. 295–308.

[16] http://www.caida.org/tools/

[17] http://www.icir.org/models/tools.html

[18] http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html

[19] J. C. Bolot. End-to-End Packet Delay and Loss Behavior in the Internet. In Proceedings of ACM SIGCOM, 1993.

[20] C. Barz, M. Frank, P. Martini, M. Pilz. Receiver-Based Path Capacity Estimation for TCP. In Proceedings of KIVS'05, Kaiserslautern, Germany. February/March 2005.

[21] P. Primet, R. Harakaly, R. Bonnassieux. Experiments of Network Throughput Measurement and Forecasting Using the Network Weather Service. IEEE Conference on Cluster Computing and Grid 2002 (CCGrid2002), Berlin, Germany. June 2002.

[22] Y. El khatib, C. Edwards. A Survey-based Study of Grid Traffic". To appear in Proceedings of the ACM International Conference on Networks for Grid Applications (GridNets 2007). Lyon, October 17-19 2007.