Which Network Measurement Tool is Right for You?
A Multidimensional Comparison Study

Esma Yildirim, Ibrahim H. Suslu and Tevfik Kosar
Department of Computer Science & CCT
Louisiana State University
70803 Baton Rouge, Louisiana, USA
{esma, ihsuslu, kosar}@cct.lsu.edu

Abstract

Network performance measurement and prediction is one of the most prominent and indispensable components in distributed computing environments. The selection of the most advantageous network measurement tool or system for specific needs can be very time consuming and may require detailed experimental analysis. The multi-dimensional aspects and properties of such systems or tools should be considered in parallel. In this paper, we take two of the most widely used and accepted network measurement tools as a case study: Iperf and Network Weather service. We compare these two prediction tools by listing the pros and cons based on accuracy, overhead, intrusiveness, system requirements, capabilities, reliability, scalability and response time. We present different methodologies used to measure their performance in previous experiments and run experiments for comparing them to actual FTP, GridFTP and SCP transfers based on different parameters.

1. Introduction

The heterogeneity of the resources that make up the computational and data grid environment plays a major factor in the efficiency and effectiveness of the services they provide to the user community of such systems. The resources have different capabilities and are quite variable over time. Network is among the most variable resources, and measurement tools for predicting the performance of its availability have become an important part of Grid architectures long ago. Various tools have been designed not only to measure the performance of network but also cpu, memory, disk and file systems [5, 4, 3, 2, 6].

A commonality between the network performance measurement tools is the idea of imitating TCP transfers and sending dummy probe data. However the measurement tools differ in the application of the usage of the probe data. Because of that they may also differ greatly in the accuracy of results they gained and have different effects on the underlying resources with a characteristic of intrusiveness and overhead generated. While some of them may be very accurate in predicting the performance and have little system requirements, they may as well lay additional burdens over the resource in terms of intrusiveness, may not scale well for applications serving a wide number of resources or respond very poorly to failures. Also tools may be even chosen based on the quality of just being easy to use. The dimensions of differentiality are so many in number and not considered altogether in the previous studies.

Among the performance measurement tools, Iperf and Network Weather Service (NWS) are the most widely accepted and used. Especially Iperf has become a standard that it is used as a base case for the actual transfer rate to compare to other measurement tools [10]. There are various network measurement studies that use Iperf and NWS for measuring performance of network and compare them to other tools [10, 14, 11, 12, 9, 15]. The methodology that is used in those studies varies and it is important to present different experimental paths the authors followed to get a clear idea of which parameters to use in our own experiments. While Iperf shows similar characteristics to the other tools in terms of design, NWS differentiates itself as a large-scale performance prediction service which is capable of many qualities rather than just being a measurement tool [15].

There are studies that compare Iperf and NWS separately to actual GridFTP transfers [14, 13, 7] and show the measurement differences among Iperf-GridFTP and NWS-GridFTP. However because those studies represent the experiments done in different environments using different parameters, it is hard to get a clear idea of which gives more accurate results to specific protocol transfer rates.

In this study, we compare Iperf and NWS in terms of multi-dimensional aspects such as accuracy, overhead, intrusiveness, system requirements, capabilities, reliability, scalability and response time. We give categorization of experimental techniques used and define our own experimental analysis path. We also present experimental results regarding Iperf and NWS' success of predicting different types of protocol transfer rates such as FTP, GridFTP and SCP by running extensive tests in aspects of dimensions such as accuracy and overhead and compare our results.

2 Measurement Techniques

We believe that identifying a correct measurement technique is essential in the way that it affects the results gained and the inference we make about them. In this section we present the techniques and parameters used to test either performance of a network or measurement tools.

The different parameters used in the studies [10, 11, 12, 9] are TCP buffer size, probe size, file size, measurement duration, window size and probe interval. Another important aspect is the current load of the traffic during measurements. As that property is quite variable over time, it may be seen that inference made on the results gained during that traffic can be biased and may not represent rules to be applied all the time. For that reason, some of the studies used controlled cross traffic so that they could easily see the effect of their transfers. These are called synthetic traffic load [11]. However in that case the consequences of a real variable traffic is ignored and assumptions made can have a quite inverse effect. The real network load may not represent the patterns that have been looked for. Hence there are other studies that also use generally wide area network loads. In that case, the measurements are repeated over time in regular or random periods. Generally the results are represented as median values and the quartile ranges.

While the performance of different measurement tools are tested and also with actual transfers, it is important to run the tools back-to-back for the same period of time so that the tools can experience the similar network variation for a specified set of parameters. However for the same tool, each measurement has a regular or random time interval between the previous one. While a tool's accuracy is compared to an actual transfer, the measurements however may not be done with the same time interval between successive iterations. In Vazhkudai et al [14], 400 GridFTP transfers occurred comparing the 1500 NWS probes during the total 12 hour period with an NWS probe size of 64KB to a randomly selected file transfers of size ranging between 10MB and 1GB.

To be able to see the effect of window size and parallel streams another strategy has been used. The sequence of window size and stream number has been deliberately selected so that they do not monotonically increase or decrease preventing any trend for biasing the result [9, 7]. Also to be able to see the affect of transfer on the RTT, pings are issued within regular intervals both during and after the transfer [9].

The comparison to be made between Iperf and NWS thus may have the following difficulties based on the techniques described above. The NWS probe size and Iperf s transferred data has a great difference in amount. While NWS sends 64KB size probes Iperf makes 10sec duration transfers to be able to capture the accurate bandwidth. Under those circumstances, we could say that NWS measures available bandwidth while Iperf measures achievable bandwidth. Of course NWS probe size could be readjusted however that will have an incredible impact on the non-intrusive design goal of NWS.

Since we want to deal with large file transfers that may take long time, the following issues may arise. The units of transfers are different as in the case of Iperf and GridFTP. While the former does the transfer in seconds, the latter does it in bytes. How can we make sure that Iperf transfers the sufficient amount of data as to predict the transfer protocol(e.g. GridFTP)? Though Iperf has an option that will take data to be transferred from an input file, if the file is too small Iperf may not reach to the steady-state and may report inaccurate bandwidth measurement. If it is too large we also know that Iperf has a limit on memory for transfers and after that threshold it reports also inaccurate results.The response time also will be affected depending on the probe/transfer size we select. In the end we may find ourselves making a sacrifice from accuracy to gain more from response time. When NWS enters at that point, it will have different parameter options and use a different underlying method for measurement other than Iperf . So we have to design a measurement strategy that will take into account all the possible problems that may occur and must decrease the effects on each other to be able to do healthy experiments.

3 Tool Evaluation

The evaluation criteria focuses on three purposes. First, we need to find the parameters that affect the accuracy and overhead of Iperf and NWS. After defining which parameters to tune, we should optimize those parameters to efficiently use both tools and observe behaviors of the tools in both local and wide area networks. Finally, the ability of Iperf and NWS to predict different data sizes by different types of protocols(e.g. GridFTP, FTP, SCP) must be measured since the insights gained from these results can be helpful for Grid applications trying to make intelligent data transfer choices and tuning (e.g. data-aware schedulers) based on network conditions. At last we define the

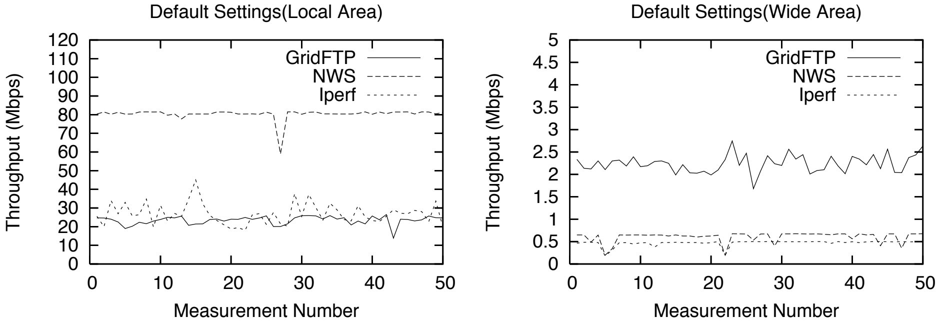


Figure 1. Measured throughput with default settings

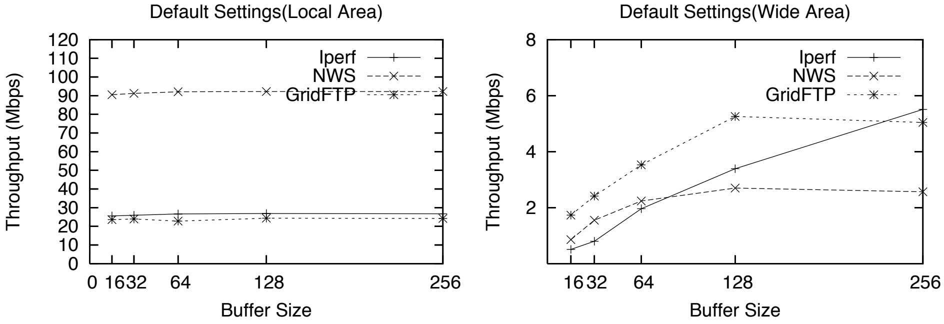


Figure 2. Effect of buffersize on throughput

scenarios about which tool is more beneficial to use, however this is very difficult since there exists a trade-off between most of the parameters. The comparison evaluation is based on the results of local and wide area transfers. Our test bed consists of three Linux machines, two of them are local area Redhat workstations in the Center for Computation and Technology and also DSL Laboratory at Louisiana State University, the last one belongs to a Suse cluster in University of Trento in Italy. All of the machines have installed Globus Toolkit and have trusted certificates for each other.

3.1 Default Settings

To be able to see the performance of tools without any tuning, both Iperf and NWS is used with their default settings in both local and wide area environments and GridFTP is chosen as the compared transfer protocol. In a default setting Iperf works with 16K buffer size and a 10 second transfer time. During that 10 seconds, it tries to consume the bandwidth. On the other hand, NWS uses lightweight 64K probe sizes with a 32K buffer size. We compared the ability of both tools to predict a 128MB file transfer with GridFTP. The transfers are started back to back with 10 seconds intervals in between and 50 iterations are performed. Figure 1 shows the results of the experiment. In local area, surprisingly NWS's light weight probes are far from predicting the actual transfer rate of GridFTP and it gives exaggeratedly high throughput measurements. The reason for this exaggerated result is because of the light weight probe size. Iperf on the other hand gives reasonable throughput measurements but saturates too much comparing to GridFTP. According to the wide area measurements, it is harder to make a comparison. Both of the tools are far from predicting the GridFTP transfer rates. For NWS, usage of light weight probes is the reason. However for Iperf, the results are related to the buffer size, since Iperf uses a smaller buffer size with respect to GridFTP.

3.2 Effect of Buffersize

The effect of buffer size is measured with again default settings in local and wide area environment. The buffer size parameter ranges between [16K-256K]. The same methodology is used as in the first experiment. The results are shown in Figure 2 . As the buffer size increases the throughput gained increases slightly and becomes steady after some point for the local area environment. We can say that the best values for throughput are gained with 64K buffer size. More importantly the tools show the same characteristic to the buffer size increase factor. There is not a significant improvement of Iperf over NWS and vice versa. However on the wide area, the results are quite different. The buffer size parameter has a huge effect. The throughput achieved increases for both Iperf and NWS although they again give low results with respect to GridFTP. The measurements of Iperf and GridFTP intersects between 128K and 256K buffer size. However for lower values Iperf results are not comparable at all. From this observation we can say that Iperf could predict GridFTP when the buffer size is properly tuned for both Iperf and GridFTP. The curve of NWS is more similar to the curve of GridFTP, however because it uses a small probe size(512K) it can not give again comparable results with GridFTP.

3.3 Effect of Probe Size on Accuracy, Computation Overhead and Response Time

The most important parameter that affects the throughput accuracy with a trade-off of intrusiveness is the probe size. By increasing probe size(transfer duration) we also increase the response time of the tool. However response time may also depend on the extra computation overhead of the tools as well. The comparison based on the accuracy and the response time of tools are made with respect to SCP transfers carried out both in the local and wide area networks and by tuning probe size parameter, we observed how quickly the tools converged to the throughput of a 128MB SCP file transfer. The buffer size is set to 128KB for both Iperf and NWS and the results are of 20 iterations . The computation overhead of the tools is the actual calculation time apart from the transfer time and is calculated as follows:

$$Overhead = Tt - (Bt/Th) \qquad (1)$$

where Tt is the response time , Bt is the bits transferred and Th is throughput in bits per second.

Figure 3.a-f shows the accuracy, computation overhead and response time of Iperf and NWS over local area network. Iperf transfers are carried out for different transfer durations ranging from 1 to 15 seconds while NWS transfers are carried out for different probe sizes ranging from 1MB to 30MB. Iperf starts to give more steady results around 4-5 seconds. However overall its measurements vary much comparing to SCP. NWS on the other hand gradually gets more accurate results and after 12MB, it becomes steady. In this case one can arrive at the conclusion that NWS arrives to steady state with a less transfer size. However to better understand we should look at the the extra computation overheads of the tools before making a judgment. Figure 3.b,e shows the computation overheads of Iperf and NWS.

Clearly the computation overhead of NWS is much higher with respect to Iperf. As we indicated before NWS overhead changes maximum 1 second with probe size, however it starts with a higher overhead about 5 seconds. Iperf overhead varies upto 8 seconds transfer duration and then steadily increases as the transfer duration increases. The response times of both tools increase linearly as the probe size increases. The additional computational overhead of NWS causes it to lose against Iperf in terms of response time.

The characteristics of both Iperf and NWS totally differs when they are used in wide area networks. First observation is that they reach the accurate results very quickly and increasing probe size and transfer duration cause them to over-measure the throughput of SCP transfers. Figure 3.g,j shows the accuracy of both tools. Because the RTT of wide area path is quite large we changed the Iperf durations to [2-45] seconds. The throughput measured by Iperf is less than NWS hence closer to the transfer rate of SCP and it becomes steady around 5 seconds while NWS becomes steady around 10-12 MB. NWS gives accurate results with 1MB however increasing the probe size results in a much higher throughput than the actual SCP transfer. The computational overhead and response times of the tools are presented in Figure 3.h,i,k,l. The computation overhead of the tools are increased as the data transferred is increased. Iperf outperforms NWS in wide area transfers in terms of computation overhead again. The response times of the tools increases linearly.

3.4 Ability to Predict Different Data Transfer Protocol Rates for various data sizes

In this section, we compare the throughput achieved by various data transfer protocols for different data transfer sizes and make an evaluation of how well Iperf and NWS can predict different types of data transfers. The set of protocols used are FTP, GridFTP and SCP. Six different file size of 512K, 1M, 16M, 64M, 512M and 1G are used as data transfers. Figure 4.a shows the protocol transfer rates in the local area network. For small file sizes GridFTP and SCP can not gain high transfer rates because of the protocol overheads. Clearly FTP outperforms these two protocols. All protocols have an equal transfer rate in 16M but as the file size increases more, FTP outperforms the others significantly. SCP performs slightly better for file sizes greater than 16M than GridFTP but overall their performances are similar. The transfer duration of Iperf and the probe size parameter of NWS ping were changed during those experiments and how well these tools converged to the actual transfer rates were analyzed. Figure 5 shows

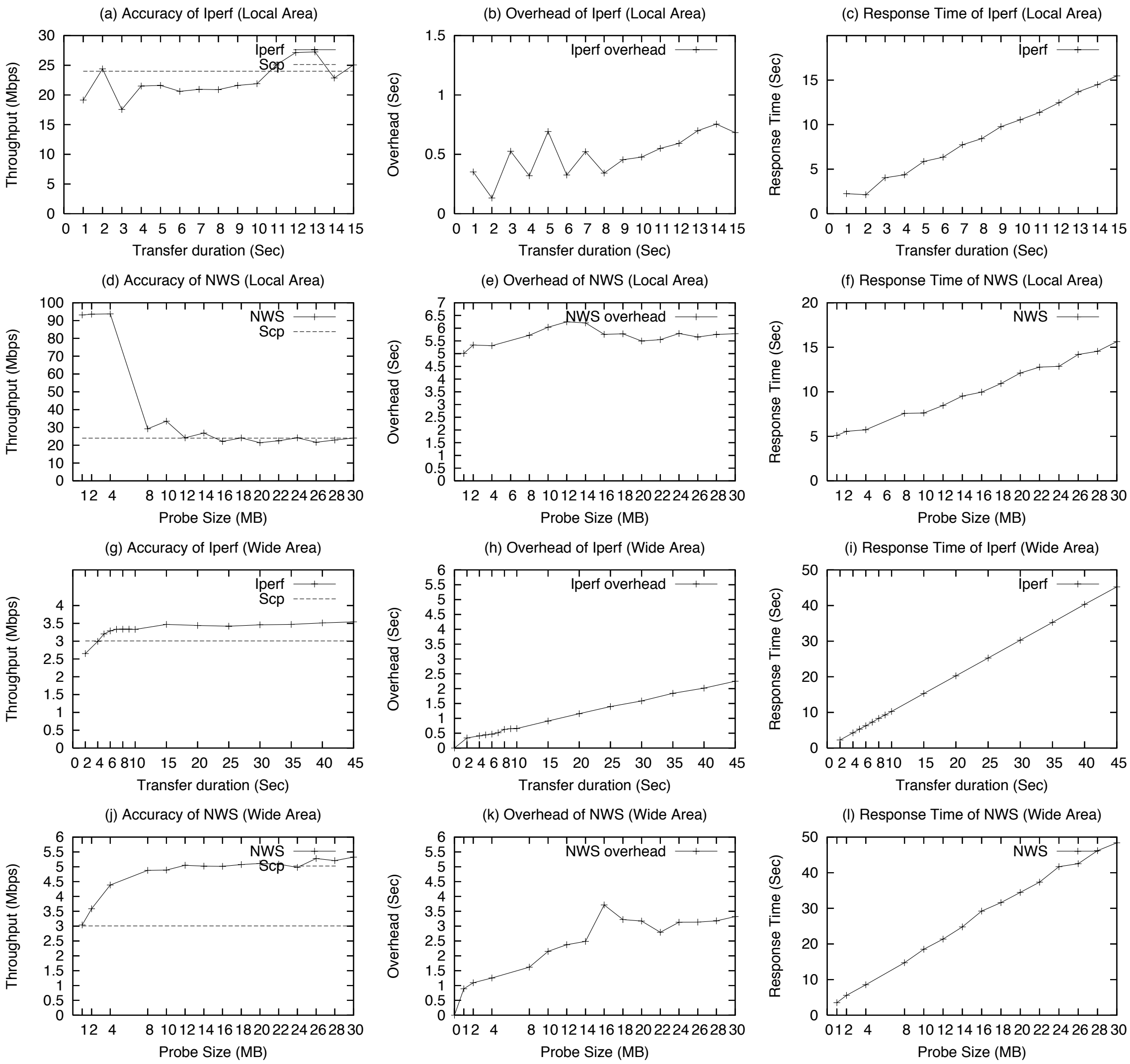


Figure 3. Accuracy and Overhead of Iperf and NWS

that both NWS and Iperf can not predict 512K transfers because of the protocol overheads but may only give similar results for FTP(a,g).For larger file transfers both of the tools can predict approximate transfer rates for GridFTP and SCP. However FTP gains more transfer rate than Iperf and NWS could predict. For Iperf, the transfer duration changes from 0.25 to 13 seconds. For small transfer durations such as 0.25 to 1 seconds, the throughput predicted is so high. After 2 seconds although the measurement varies a lot, it converges to the protocol transfer rates and becomes steady around 9-10 seconds. NWS probe size varies between 64KB to 256KB exponentially. The NWS predictions converges to the protocol transfer rates around 8-16MB probe size. The total transfer time varies between 7 to 10 seconds, between 8-16 MB probe sizes.

We conducted the same tests over the wide area network. Figure 4.b shows the protocol transfer rates achieved. A difference observed from the local area protocol transfer rates is that SCP performs the worst and GridFTP performs better. FTP of course has the ultimate performance again. While the transfer duration range is kept the same for Iperf in wide area, we selected smaller probe sizes for NWS ranging from 64KB to 20MB. Figure 6 shows the throughput measured by the tools comparing to the protocol transfer rates. 256KB buffer size is used. When the probe size or transfer duration are increased for NWS and Iperf, the bandwidth measured by the tools is increased. Then again the tools fail to predict file sizes under 16MB for all protocols and can predict SCP for smallest probe size and transfer duration. For large file sizes although both tools predict

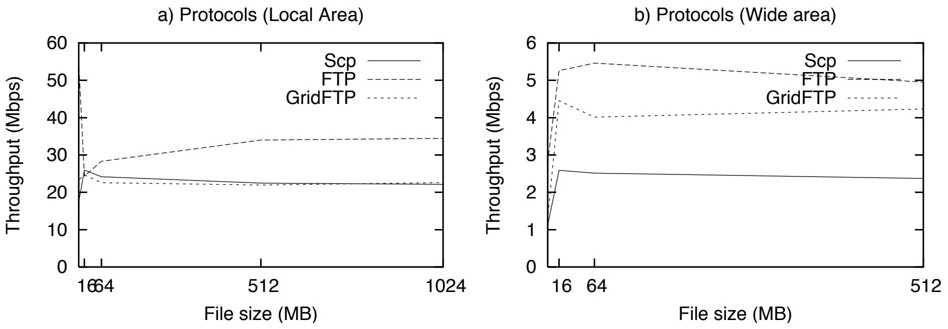


Figure 4. Protocol transfer rates

GridFTP and FTP with a good precision. We could say that both tools give higher throughput results than the protocols can achieve when the probe size and transfer duration is increased in the wide area except for FTP and also GridFTP and FTP transfer rates are better predicted.

From the above insights we could make the following conclusions:

- It is not wise to use measurement tools to predict the transfer rates of small sized data transfers because of the protocol overheads and the time to fill the network pipe is not enough for a small transfer.
- Iperf and NWS can better predict protocols whose parameters can be set equal as themselves. For example, GridFTP transfers are better predicted because its buffer size parameter is set the same as Iperf and NWS. However every protocol may give differentiated transfer rates because of their different computational overheads
- NWS measurements are more stable and Iperf measurements are more varied. For local area transfers Iperf reaches the stable protocol transfer rates more quickly than NWS. For wide area both tools seem to have similar behavior.
- Both local and wide area measurements need a certain amount of probe size to give accurate results. However the inaccurate measurements fall down in local area while in wide area the measurement results rise up to reach the stable transfer rates.

3.5 Accuracy vs Total Overhead

In order to get accurate results we have seen that we may need a certain amount of probe size. However that comes with the overhead in getting response from the tools. We made a comparison of the total transfer times (response time) of Iperf and NWS measurements when they converge to the protocol transfer rates and the results are given in Figure 7 and 8 with 90% accuracy for local and wide area networks. It is observed that Iperf has less overhead comparing to NWS ping in most of the cases.

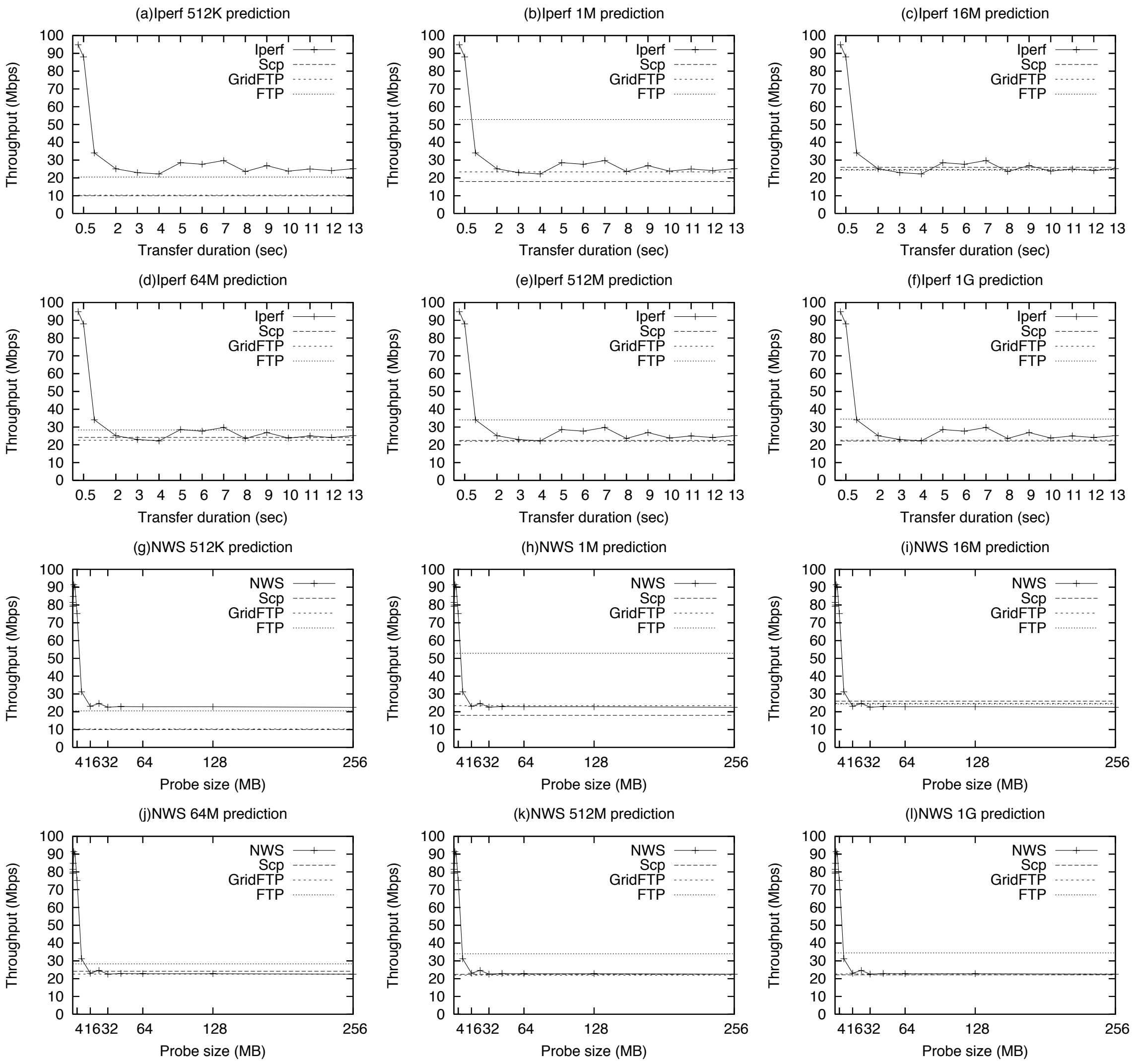
4 How to Choose?

The evaluation of a tool depends on many aspects such as accuracy, overhead, intrusiveness, system requirements, capabilities, reliability, scalability and response time. For an effective performance gain, all should be considered in parallel.

Tool vs Service System. The majority of Grid users prefer to use stand-alone individual performance measurement tools which are considered to be easy to install and use. However performance monitoring service systems are overlooked since they are only used for optimization rather than enabling application execution [15]. Individual tools are designed for single user execution. Tools like Iperf measure the end-to-end network throughput by sending data from a source to a sink host. If the number of users is high, this can place a heavy burden on the network and additional synchronization is needed. This also brings many other issues such as weakness against failures, scalability, and security concerns.

On the contrary, a performance service system monitors resources constantly without perturbing them and this requires the system to be ubiquitous and invisible to users and administrators. Although some overhead is included, it is negligible comparing to aggressive measurement tools. NWS declares itself to be a measurement service system which runs continuously so that it can gather a history of available performance from each monitored resource with lightweight probes, so that it could respond at the time of request. Iperf on the other hand, runs on demand, sends dummy data to gather information about performance.

Accuracy. Iperf has become a de-facto standard in network performance measurement and there has been so many


(a)Iperf 512K prediction

(b)Iperf 1M prediction

(c)Iperf 16M prediction

(d)Iperf 64M prediction

(e)Iperf 512M prediction

(f)Iperf 1G prediction

(g)NWS 512K prediction

(h)NWS 1M prediction

(i)NWS 16M prediction

(j)NWS 64M prediction

(k)NWS 512M prediction

(l)NWS 1G prediction

Figure 5. Throughput predictions of Iperf and NWS against FTP,GridFTP and SCP in Local Area

studies which run experiments to test networks and impact of different parameters using Iperf [11, 12, 9]. There have been some studies comparing Iperf to other measurement tools [10], but the studies that compare NWS and Iperf are limited and not in detail.

Shriram et al. [11] compares the accuracy of Iperf to a number of other tools. The authors ran their experiments under both synthetically generated cross traffic and also the replay of TCP traces from a source. The results however turned out to be quite different from each other. According to Shriram et al results, with a maximum buffer size of 227 KB, Iperf is accurate within 15% or better. Also it is noted that using smaller buffer sizes did not make a significant difference surprisingly. However Iperf's performance was found quite disappointing when run through real tcp tracebacks. This observation is explained with the loss occurred when the tool and cross-traffic streams merged. Even though the amount of loss was insignificant, it caused Iperf to halve its congestion window and triggered a significant number of retransmissions. The retransmission timeout was so large that this time consumed 75% of Iperf running time.

The preference of a measurement tool instead of NWS is generally explained to be that the tool provided more accurate results and that NWS probes had the time-out mechanism which other tools did not and which generated an extra overhead and had an effect on the accuracy of measurement. However Wolski et al. [15] showed that the range of measurement values of NWS intersected with other tools like Iperf, nettcp and netperf. The authors ran each tool back to back every 60 seconds and over 72 hour period. All of

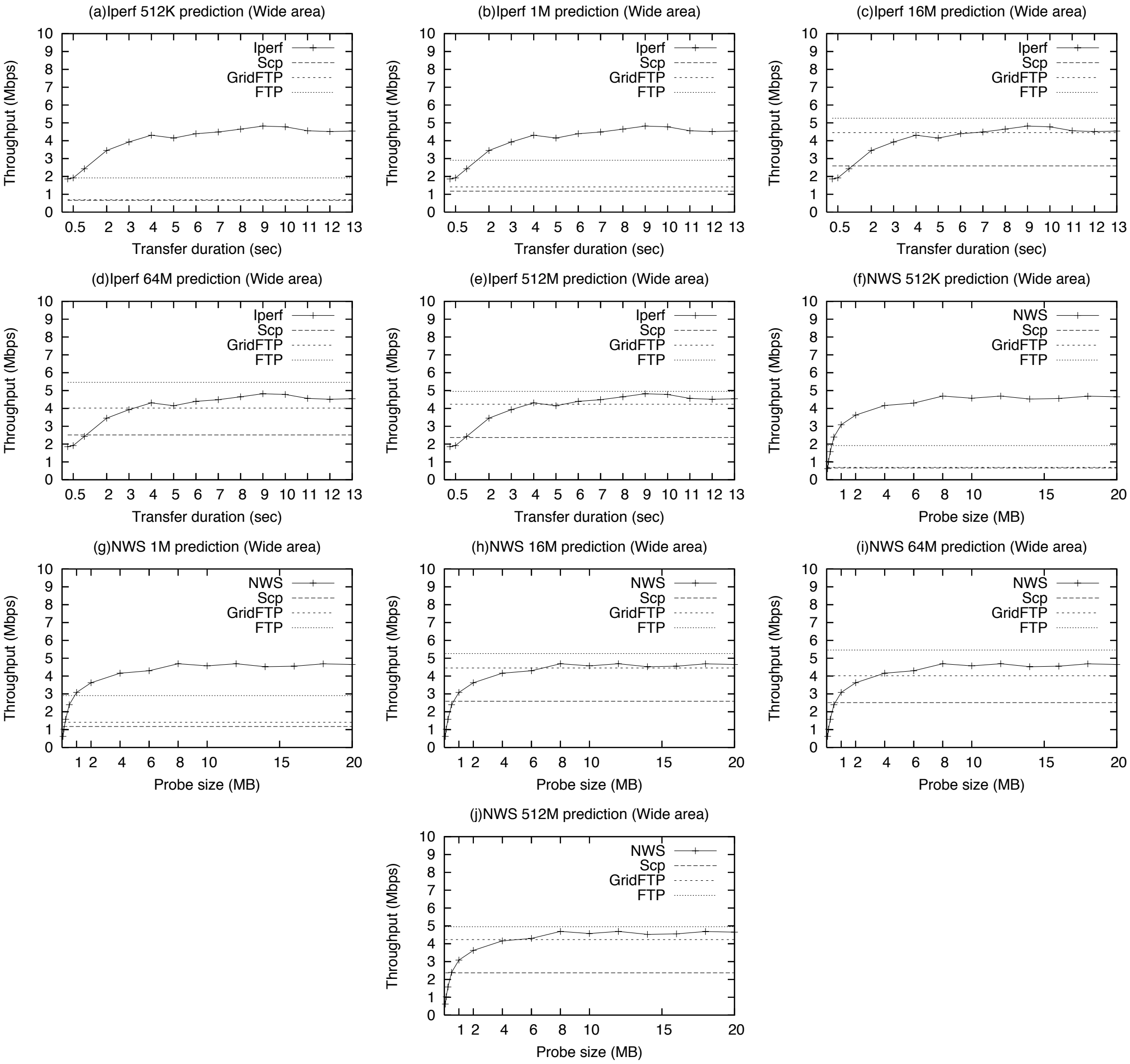


Figure 6. Throughput predictions of Iperf and NWS against FTP,GridFTP and SCP in Wide Area

the tools were configured to use the same amount of end system buffering. The results show overlapping ranges and yet NWS had the overhead of implementing reliable socket time outs. However those results do not show the accuracy of NWS and Iperf comparing to the actual bandwidth but only the closeness of values to each other yet NWS is used to transfer the same amount of data as Iperf making it as intrusive as Iperf.

According to the tests we have conducted with properly tuned buffer and probe sizes, NWS gives more accurate results than Iperf in local area networks and their results are almost similar in wide area networks.

Ease-of-Use. The NWS installation comes with different modules and commands and may require additional configuration. For the system to run properly, at least three modules should be running: nameserver, memory and sensor. On the other hand Iperf is a simple single command tool with a number of command options which should be run in server mode to give services to the requesting hosts. That feature makes Iperf more preferable. If the tool is well-engineered, documented, simple to understand, install and use, it is more likely to be preferred by users. On the other hand NWS is a long-running Grid service designed to support many users and resources simultaneously hence necessarily more complicated and take time to thoroughly master it. In this case Iperf is preferable than NWS in terms of ease-of-use.

Capability. Iperf is a simple tool with the capability of estimation of the achievable throughput but not the available bandwidth. On the other hand NWS not only monitors net-

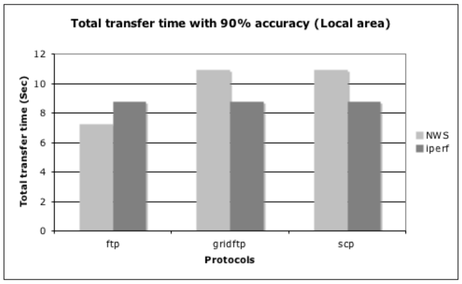


Figure 7. Total transfer time with 90% accuracy over Local Area

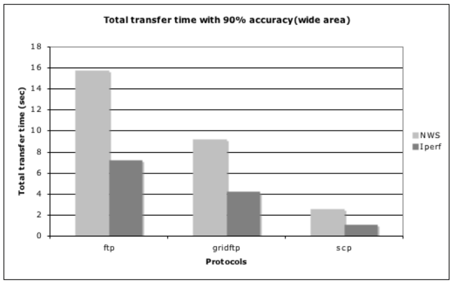


Figure 8. Total transfer time with 90% accuracy over Wide Area

work but also CPU, disk, memory and file system. It has a number of monitors for availability of a machine, availability of CPU cycles, availability of disk capacity, performance of a file system, TCP bandwidth and latency, TCP connection time and memory speed [15]. Those properties makes NWS more advantageous in terms of capabilities.

Scalability. The NWS components are designed with the ability to scale to a large number of hosts. The burden on the most centralized module which is NWS nameserver has been kept to the minimum. Only one nameserver has the capability to serve hundreds of hosts. However for more frequent accesses nameserver can be replicated. The case for memory component is a little bit different. Because large data is stored on memory modules it is more likely that there has to be many of memory hosts as the system scales to large number of hosts. However they are very flexible about where they can be placed. On the other side Iperf does not collect information constantly and any host which has the utility, at the same time has the ability to question another host. However this situation leads to a big problem when the system scales to large numbers. Each measurement done by Iperf causes transfer of data continuously for tens of seconds. If it is used rarely then the load generated by them is negligible. However when they are used by many hosts without coordination, the effect can be huge especially on the bottleneck link that dominates the end-to-end performance. NWS solves this by using the clique technique with which only one probe sequence measures that specified path which makes NWS a more scalable tool than Iperf.

Reliability. The problem with the use of a tool like Iperf rather than a service is that the abortion of the tool had to be done manually from the command line in case of failures. As an example, the project in [1] suffers from the reliability of Iperf servers. Typically Iperf servers die on a host once a day. A lot of effort has gone into detecting and automatically restarting them. Multiple strategies are necessary due to the non-uniform configuration of the remote hosts. The preferred is to use ssh to automatically start and stop the servers, if this is not possible then the servers are left running, and email sent that they need a manual restart. On the other hand NWS solves this problem with its portable timeout mechanism and adaptive time out discovery protocol [8] .However those mechanisms are said to bring additional overhead to the quality of measurements. But if the reliability is the most important concern then NWS is definitely the best choice.

Intrusiveness. The NWS is designed to be a light-weight monitoring system which uses 64KB probes to estimate the available network bandwidth [14]. The sensor modules also designed to be as non intrusive as possible on the host resource and uses only 2-4 MB system memory. Iperf, on the other hand generates large overhead(more than 70%) as it tries to fill the tight link to get as much bandwidth as available. Iperf is known to be the most expensive tool in terms of intrusiveness and overhead comparing to other measurement tools in [10]. Its traffic can have a huge impact on the existing cross traffic [11]. However this aggressiveness gives Iperf , the accuracy property since it tries to fill the network pipe just like a normal data transfer would.

On the contrary, the experiments we have made showed that NWS can not predict the network with its light weight probe size. However with properly tuned probe size, it is able to give accurate results which makes it as intrusive as Iperf.

System Requirements. For NWS to be fully functional, at least one nameserver, one memory and as many sensors as the number of hosts it is measuring should be running. As nameserver keeps only small registration information, requirements are extremely low. However same thing cannot be said about the memory module as it stores time series data. Each sensor node also needs small system requirements from the host it is running. Iperf however does not keep continuously collected data and it serves on demand. But, at the time of request it consumes the bandwidth by performing an actual transfer. There is a physical limitation in the amount of memory that iperf can hold to calculate the throughput. It was found that this value is 4gbytes. This is equivalent to a duration of 32 seconds if we can transfer at GigE rates (1000mbit/sec).

Response Time. Tools like Iperf need long duration data transfers to be able to catch the performance of network during steady-state of data transfers. This time is 10sec by default. The time required to calculate an NWS forecast is 161 microseconds [15] which is very negligible. However the time for sending periodic probes are much larger but has not been mentioned in the previous studies. According to the experiments we have made with properly tuned parameters, nws ping's response time is larger than Iperf. So if time is critical in measurements, it is better to use Iperf.

5. Conclusions

Iperf and Network Weather Service: two widely used tools are compared to each other considering various acpects. The experiments conducted using both local and wide area networks showed that their behaviors change according to the network environment they are used. We have seen that there are many parameters like buffer size and probe size to tune for both tools to be able to predict the transfer rates of certain protocols. The most important parameter for them is probe size since neither of them could give accurate results with their default settings. In an environment where the response time of tools is important, it is better to use Iperf instead of NWS since its overhead is higher. However for a more robust system in which resources scale to large numbers as well as requests to the tools, NWS is a more beneficial service. Moreover the measurement results of NWS is more stable than Iperf and it is easier to determine the smallest probe size since once NWS reaches to the protocol transfer rate, it does not saturate for larger probe sizes.

Another observation is that both Iperf and NWS when tuned with proper probe and buffer sizes may over or underestimate the protocol transfer rates due to the protocol overheads. For example they underestimate FTP transfer rate while they overestimate SCP transfer rate. Also the protocol transfer rates changes a lot according to the data size. The throughput of the protocols enlarges as the data sizes increase so it is wise to consider the data size while using results of the tools predictions. For small data sizes there is no use in using a measurement tool for predicting the transfer rate since protocol overheads are large. Overall while Iperf has faster response times NWS is more robust and produces more stable measurement results.

6 Acknowledgment

This project is in part sponsored by the National Science Foundation under award numbers CNS-0619843 (PetaShare) and EPS-0701491 (CyberTools), and by the Board of Regents, State of Louisiana, under Contract Numbers DOE/LEQSF (2004-07), NSF/LEQSF (2007-10)-CyberRII-01, and LEQSF(2007-12)-ENH-PKSFI-PRS-03.

References

[1] Bandwidth to the world project. http://www-iepm.slac.stanford.edu/monitoring/bulk/sc2001/.

[2] Iperf. http://dast.nlanr.net/Projects/Iperf.

[3] The network weather service. http://nws.cs.ucsb.edu/.

[4] Pathchirp. http://www.spin.rice.edu/Software/pathChirp/.

[5] Pathload, a measurement tool for the available bandwidth of network paths. http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathload.html.

[6] Spruce. http://project-iris.net/spruce/.

[7] B. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Secure, efficient data transport and replica management for high-performance data-intensive computing. In *Proc. IEEE Mass Storage Systems and Technologies Conference(MSS01)*, page 13, Apr. 2001.

[8] M. S. Allen, R. Wolski, and J. S. Plank. Adaptive timeout discovery using the network weather service. In *Proc. IEEE International Symposium on High Performance Distributed Computing (HPDC02)*, pages 35–41, 2002.

[9] L. Cottrell, C. Logg, and I. Mei. Experiences and results from a new high performance network and application monitoring toolkit. In *Proc. Passive and Active Measurement Workshop (PAM03)*, 2003.

[10] M. Murray, S. Smallen, O. Khalili, and M. Swany. Comparison of end-to-end bandwidth measurement tools on the 10gige teragrid backbone. In *Proc. IEEE/ACM International Workshop on Grid Computing (GRID05)*, pages 300–303, Nov. 2005.

[11] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and K. Claffy. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In *Proc. Passive and Active Measurement Workshop (PAM05)*, Mar. 2005.

[12] A. Tirumala, L. Cottrell, and T. Dunnigan. Measuring end-to-end bandwidth with iperf using web100. In *Proc. Passive and Active Measurement Workshop (PAM03)*, 2003.

[13] S. Vazhkudai and J. Schopf. Predicting sporadic grid data transfers. In *Proc. IEEE International Symposium on High Performance Distributed Computing (HPDC02)*, pages 188–196, 2002.

[14] S. Vazhkudai and J. Schopf. Using regression techniques to predict large data transfers. *International Journal of High Performance Computing Applications*, 17(3):249–268, Aug. 2003.

[15] R. Wolski, G. Obertelli, M. Allen, D. Nurmi, and J. Brevik. Predicting resource performance online. *Handbook of Nature-Inspired and Innovative Computing*, pages 575–611, Mar. 2006.