A.1 GODDI motivation/significance/vision/background

Data intensive sciences such as High Energy and Nuclear Physics, global weather prediction, astronomy, and bio-informatics have critical needs to share large volumes of data (already reaching the PetaByte scale). Without this ability the large global scientific collaborations such as the PPDG or BaBar [BaBar] or the LHC, will be unable to work efficiently. However, though high and higher links are becoming available, it is increasingly difficult to achieve high throughput between distantly separated sites.

There are several reason why it is difficult to achieve good performance on high performance long-distance links. These include: limitations of today's standard TCP (Reno based) stacks, the need for large windows AND parallel streams to overcome the TCP limitations; the "Wizard gap" in understanding how to configure tune applications and networks; the difficulty in simultaneously optimizing both TCP buffer/window sizes and numbers of parallel streams; the need for dynamic (during transfers) changes in configurations based on recent measured performance & predictions. These all indicate a need for new transport mechanisms and the need to to automate configurations.

Estimating the major optimum parameters (the TCP maximum buffer/window sizes, and numbers of parallel streams) today typically involves measuring the Round Trip Time (RTT) and the "bottleneck bandwidth" to yield the Bandwidth-Delay Product (BDP). The RTT is usually easy to estimate using the ping facility, or if that is blocked by using atool such as SynACK [SynACK]. There are both lightweight and heavier weight-tools to measure the "bandwidth", however both have their drawbacks. Today's lightweight bandwidth measurement tools such as ABwE [ABwE], Patchhirp [Pathchirp], and Pathload [Pathload] don't work on emerging high speed networks due to interrupt coalescence, off-loading of TCP functionality to NICs, and insufficient granularity of the system clocks. At the same time, methods such as iperf [Iperf] to measure TCP throughput are very network intensive requiring the transfer of hundreds of Mbits/s for many seconds (to minimize the effects of start up effects such as slow start) to measure Gbits/s throughput. It is increasingly important therefore to incorporate the measurements into applications, so that bandwidth is not wasted extraneously, and the measurements can track the transfers.

In addition to being able to achieve high network throughput, there are also bottleneck at higher layers such as the disk and file level and the servers that go with them. Overcoming these bottlenecks requires new innovative parallelism techniques to be developed..

Can we say anything about the statistics side of things. E.g. current techniques of analyzing the data use rudimentary statistical methods, can we say we plan to improve the application of modern statistical methods to our problem area? What would be especially useful would be techniques to detect temporal change anomalies in the data to raise alerts. The nuclear industry uses technologies such as Sequential Probability Ratio test to discover problems at an early stage, is that related?

User needs ability to choose transport layer to optimize performance (e.g. use HSTCP-LP {HSTCP-LP] to soak up unused bandwidth, use HS-TCP [HS-TCP] to get higher, but fair performance, use UDT [UDT] since do not have advanced TCP stack available in proprietary OS' such as Solaris).

Heavy use of non GridFTP [GridFTP] techniques for bulk data movement in data intensive fields such as HENP (who use bbcp [bbcp] and bbftp [bbftp]) points to the need for a data mover that is easy to use, easy to install (e.g. does not require certificates, and/or Globus environment).

A.2 Prior/related work/state of the art

A.2.1 Introduction

Achieving high performance bulk-data transport involves multiple layers, each with its own bottlenecks, each of which needs careful attention. Obviously at the lower layers a high speed network path is a pre-requisite. Several high-speed testbeds (e.g. TeraGrid, UltraScienceNet, UltraLight, UKLight, NetherLight, DataTAG, and the yearly SuperComputing show testbeds) are in use or proposed today and provide high performance paths of up to 10Gbits/s. At the same time many production networks (ESnet, Abilene, GEANT) now commonly support 1Gbits/s end-to-end paths. Despite this, frequently users are unable to achieve even close to the expected performance. The next bottleneck to performance typically occurs at the transport layer.

A.2.2 Reliable transport protocols

The performance limitations of standard (Reno based) TCP on Fast Long-Distance (FLD) networks are well understood (see for example [HS-TCP]). As a result several groups are developing new TCP stacks and UDP based reliable transports.

The start-up phase of a TCP flow uses an algorithm known as "slow-start" to estimate the available bandwidth. On FLD paths this can take many seconds and often provide a poor estimate. Efforts [Slow-start] to address this are underway.

Much more important for bulk-transfers running for minutes or hours is TCP's behavior for most of the rest of the transfer in the "congestion avoidance" phase. In this phase, standard TCP uses an Additive Increase Multiplicative Decrease (AIMD) algorithm. AIMD dramatically reduces, by a factor of two, the amount of data sent without acknowledgement (congestion window) when encountering congestion (as identified by lack of acknowledgements), and then recovers very slowly (the congestion window increases by one packet for each acknowledgement received) on FLD paths. The advanced TCP stacks modify the AIMD behavior to reduce the factor of two multiplicative decrease and increase the recovery rate. The Vegas, Westwood, HSTCP-LP and FAST TCP stacks also utilize increases in the Round Trip Time (RTT) to indicate congestion. Evaluation [Bullot] of several of the advanced TCP stacks indicates that they provide much improved single-stream performance on FLD paths and the best provide reasonable stability and fairness.

Most of the advanced TCP stacks are implemented in kernel space. This requires access to the kernel sources, root privileges to build, and today the stack upgrades are not synchronized with operating system (OS) patches upgrades etc. This makes deployment difficult, in fact at this time, none of the new TCP kernels exist for proprietary OSs. Thus, there is considerable interest in efficient user-space reliable transport implementations. UDT is such a promising reliable transport protocol, it is built on UDP rather than TCP and runs in user space. UDP implementations tend to be less efficient than TCP's and use more compute cycles per bit transferred. This is a serious concern since for the emerging 10Gbits/s paths compute power is a gating factor to high performance. However, the most recent versions of UDT are closing this gap.

{Constantinos we need something on SOBAS}

A.2.3 File transfer/copy applications

High performance file transfer/copy middleware applications such as bbcp, bbftp and GridFTP utilize the reliable transport services. To achieve high performance these applications provide options such as large TCP send and receive buffers/windows and parallel TCP streams. Today these performance options are set once at the start of the transfer and not modified to track path performance during the transfer. There are also UDP based file transfer applications such as RBUDP [RBUDP] and Tsunami [Tsunami], however, they have not achieved any notable production usage in the data-intensive science and Grid communities.

A.2.4 Data placement & replication

The data sets produced in fields such as HENP, climate, astronomy etc. are already large reaching into the PetaByte range today, and are expected to grow to ExaBytes in the next decade. Copying these data sets in an acceptable time requires high throughput performance and long transfer times (it takes roughly a day to transfer a TByte at 100Mbits/s). Determining the optimal options to use for a large data transfer (e.g. maximum TCP buffer/window sizes, number of parallel streams) requires that the hosts be properly configured, that the middleware can provide the required options, and that the user knows the appropriate settings for the selected network path(s). Setting up the transfer in an optimal fashion can result in orders of magnitude improvement in throughput performance compared to the default settings. As networks improve in performance this manual optimization is increasingly difficult [wizard gap]. It is increasingly apparent that the middleware needs to be made more network-aware so it can optimize itself. Further with transfers taking many hours or even days during which bottleneck bandwidth can vary by orders of magnitude or more when measured at 20 second intervals [DRS] (measuring achievable throughput via iperf, and file transfer by bbftp every 90 minutes for 10-20 second durations, factors of two or more variations are seen [IEPM-BW]), automating this process so the optimal adjustments can be made during a transfer is increasingly important.

Increasingly data intensive science disciplines such as HENP [PPDG] utilize multiple computer and data centers to process their data. As a result there are copies of the data at multiple sites. This allows another layer of optimization to be achieved by selecting to receive the data from the replica site with the highest available bandwidth. Even better performance may be achieved if disjoint parts of the data can be transferred from multiple sites in parallel {need to read up on SplitStream, Bullet, Divisible Load Theory etc. this is not my area of expertise} in such a way as to optimize the overall transfer. The Logistical Networking concepts [Loci] also enable the use of multiple sites by providing meta-data to indicate the locations where the data may be found. The initial selection of sites from which to transfer the data requires a knowledge of the available bandwidth during the potential expected transfer time. Armed with such predictions it is possible to estimate how much data to transfer from each of several sites (each with potentially different available bandwidth predictions over the period of interest) so all the transfers finish at roughly the same time [Divisble Load Theory].

The Network Weather Service [NWS] and other projects make repetitive measurements of a metric such as available bandwidth and then use various prediction techniques to extrapolate into the future. Typically this provides accuracies of 80-90% for one or two hours forward [cottrell]. One difficulty with this, even assuming they are available, is finding the measurements and predictions that are relevant to the source and destinations for the planned transfers. An alternative is to make the required measurements on demand. However this may require special facilties to be set up at the sources and destinations together with the privileges to use them. Another attractive alternative is to use measurements made during the transfer itself (e.g. from the application itself or if available from the underlying transport layer itself e.g. via [Web100], or even from network devices) to adjust the transfer method and to provide estimates of transfer time still required etc. This requires a new breed of middleware bulk-data transfer/copy utilites that are network aware. On top of these can be built the disjoint parallel transfers from multiple sites.

A.3 Evaluation

Before any new developments are deployed in production, they need careful evaluation to determine their applicability in multiple environments including both ultra high speed testbeds and high-speed production networks. They also need to be compared with other alternatives so recommendations can be made on their relative benefits. For successful developments we will move to pilot deployment in production applications such as a large HENP experiment like BaBar. This will further help evaluate their applicability to the real world, wring out deployment problems, get feedback and support from the scientists who matter, and assist in getting traction for wider scale deployment.

We will work with and provide feedback to the developers of the various TCP stacks and UDP reliable transports, to evaluate and report on their performance, fairness, stability, and ease of integration deployment etc., and to appropriately encourage their development. The evaluation will utilize the ~ 40 high speed production paths set up as part of the IEPM-BW infrastructure. This has a wide range of bottleneck speeds from 10 Mbits/s to 900Mbits/s and sites in 9 countries. In addition we will use our access to various 10Gbit/s testbeds to evaluate at higher speeds.

As the modified version of bbcp/SOBAS becomes available we will plug it into the IEPM-BW framework, evaluate its performance on multiple production links and iterate with the developers shake it down and improve it. The testing will include memory to memory and disk to disk transfers, various TCP stacks and reliable UDP transports, and comparison of single and multiple parallel streams.

Following this we will develop a methodology for evaluating the effectiveness of selecting the best places to transfer data from and use this to evaluate and compare the data location selection techniques we and others have developed.

A.3.1 Deployment

We will work with BaBar physicists to deploy the preferred TCP stacks and/or reliable UDP transports at SLAC and collaborator sites. Currently the large SLAC production data servers utilize a proprietary OS (Solaris), so at SLAC we will start out evaluating bbcp running over user space transports such as UDT. Other BaBar collaborators (e.g. IN2P3 at Lyon France) utilize Linux for their data servers so we will work with them to evaluate bbcp with different TCP stacks. Caltech is also currently working with Sun to port FAST to Solaris. If this is successful, we have agreed with Sun and Caltech to assist in the testing and evaluation of FAST/Solaris and will include the bbcp

A.3.2 SLAC Facilities

SLAC is the home of the BaBar HENP experiment. BaBar was recently recognized as having the largest database in the world. In addition to the large amounts of data, the SLAC site has farms of compute servers with over 3000 cpus. The main (tier A) BaBar computer site is at SLAC. In addition, BaBar has major tier B computer/data centers in Lyon, France, near Oxford, England, Padova, Italy and Karlsruhe, Germany which share TBytes of data daily with SLAC. Further BaBar has 600 scientist and engineer collaborators at about 75 institutions in 10 countries. This is a very fertile ground for deployment and testing of new bulk-data transfer utilizing improved TCP stacks and Grid replication middleware. There are close ties between the SLAC investigators, the BaBar scientists and the SLAC production network engineers.

The IEPM-BW infrastructure, developed at SLAC, hasin 5 countries 10 monitoring sites and about 50 monitored sites with contacts, accounts, keys, software installed etc. This provides a valuable testbed for evaluating new TCP stacks etc. The SLAC site has high speed connections (OC12 and GE) to the CENIC/Abilene and ESnet backbones. The SLAC IEPM group has a small farm of 6 high-performance network test hosts with 2.5 to 3.4GHz cpus and 10 GE Network Interface Cards (NICs). SLAC hosts network measurement hosts from the following projects: AMP, NIMI, PingER, RIPE, SCNM, and Surveyor. SLAC has two GPS aerials and connections to provide accurate time synchronization.

SLAC has an OC12 Internet connection to ESnet, and a 1 Gigabit Ethernet connection to Stanford University and thus to CalREN/Internet 2. We have also set up experimental OC192 connections to CalRENII and Level(3). The experimental connections are currently not in service, but have been successfully used at SC2000-2003 to demonstrate bulk-throughput rates from SuperComputing to SLAC and other sites at rates increasing over the years from 990 Mbits/s through 13 Gbps to 23.6 Gbps. SLAC is also part of the ESnet QoS pilot with a 3.5 Mbps ATM PVC to LBNL, and SLAC is connected to the IPv6 testbed with three hosts making measurements for the IPv6 community[1]. SLAC has dark fibers to Stanford University and PAIX, SLAC plans to connect at 10Gbits/s to the DoE UltraScienceNet and UltraLight testbeds later this year. The SLAC IEPM group has access to hosts with 10Gbits/s connectivity at UvA on the NetherLight network in Amsterdam, at StarLight in Chicago and CERN in Geneva. We also have also close relations with Steven Low's group at Caltech and plan to get access to their WAN-in-Lab setup for testing applications with dedicated long distance fiber loops. SLAC has been part of the SuperComputing bandwidth challenge for the last 3 years, part of the team that won the bandwidth challenge last year for the maximum data transferred. Two time winner of the Internet2 Land Speed Record.

As part of our previous and continuing evaluations of TCP stacks, the SLAC IEPM team has close relations with many TCP stack developers (in particular the developers of FAST, H-TCP, HSTCP-LP, LTCP) and with the UDT developers.

B References

[ABWE] J. Navratil, R. L. Cottrell, "A Practical Approach to Available Bandwidth Estimation", published at PAM 2003, April 2003, San Diego. http://moat.nlanr.net/PAM2003/PAM2003papers/3781.pdf.

[bbcp] A. Hanushevsky, A. Truov, R. L. Cottrell, Peer-to-peer computing for secure high performance data copying. In Computing in High Energy Physics, Beijing, 2001. Available for download etc. at http://www.slac.stanford.edu/~abh/bbcp/

[bbftp] IN2P3, Large Files Transfer Protocol, available http://doc.in2p3.fr/bbftp/

[BaBar] see http://www.slac.stanford.edu/BFROOT/

[Beck] M. Beck, T. Moore, J Plank, An End-to-end Approach to Globally Scalable Programmable Networking, Workshop on Future Directions in Network Architectures, Karlsruhe, Germany, August, 2003, available http://loci.cs.utk.edu/modules.php?name=Publications&d_op=downloadPub&lid=175

[Bharadwaj and Robertazzi 96] B. Veeravalli, D. Ghose, V. Mani, and T. Robertazzi, Scheduling Divisible Loads in Parallel and Distributed Systems. Los Alamitos, CA: IEEE Computer Society Press, Sept. 1996.

[1] See for example *SLAC IPv6 deployment* presented by by Paola Grosso at the Internet2 Member meeting, Indianapolis Oct.13-16

[Bharadwaj and Robertazzi 03] B. Veeravalli, D. Ghose, and T. G. Robertazzi, Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems , in special issue of Cluster Computing on Divisible Load Scheduling (T. G. Robertazzi and D. Ghose, eds.), vol. 6 of 1, pp. 7 18, Kluwer Academic Publishers, Jan. 2003.

[Bullot] H. Bullot, R. Les Cottrell, R. Hughes-Jones, Evaluation of Advanced TCP Stacks on Fast Long-Distance Production Networks, submitted to Journal of Grid Computing, 2004, also SLAC-PUB-10402

[DRS] Wu-chun Feng, M. Fisk, M. Gardner, E. Weigle, Dynamic Right-Sizing: A Automated, Lightweight, and Scalable Techniques for Enhancing Grid Performance, proceedings of the 7th International Workshop on Protocols for High-Speed Networks, Berlin. April 2002.

[GridFTP] W. Allcock, The GridFTP Protocol Protocol and Software. Available http://www.globus.org/datagrid/gridftp.html

[HS-TCP] Sally Floyd, HighSpeed TCP for Large Congestion Windows, RFC 3649, Experimental, Dec 2003

[HSTCP-LP] A. Kuzmanovic, E. Knightly, R. L. Cottrell, A Protocol for Low-Priority Bulk Data Transfer in High-Speed High-RTT Networks, presented at PFLDnet 2004, ANL, Feb 2004 available http://dsd.lbl.gov/DIDC/PFLDnet2004/papers/Kuzmanovic.pdf

[IBP] J.S. Plank, S. Atchley, Y. Ding and M. Beck, Algorithms for High Performance, Wide-Area Distributed File Downloads, Parallel Processing Letters, Vol. 13, No. 2 (2003), pages 20.

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

[LHC] LHC: Large Hadron Collider Home Page, available http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/

[Pathchirp] V. Ribeiro and R. Riedi and R. Baraniuk and J. Navratil, R. L. Cottrell, pathChirp: Efficient Available Bandwidth Estimation for Network Paths, Passive and Active Measurements 2003,

[Pathload] M. Jain, C. Dovrolis, Pathload: A Measurement Tool for End-to-End Available Bandwidth, Passive and Active Measurements 2002.

[PPDG] Particle Physicse DataGrid, available http://www.ppdg.net/

[RBUDP] E. He, J Leigh, O. Yu, T. DeFanti, *Reliable Blast UDP: Predictable High Performance Bulk Data Transfer*, IEEE International Conference on Cluster Computing (CLUSTER'02), Sep 22-26, 2002

[Robertazzi 03] T. Robertazzi,   *Ten Reasons to Use Divisible Load Theory* , Computer, 2003.

[Slow-start] Sally Floyd, Limited Slow-Start for TCP with Large Congestion Windows, RFC 3742, Experimental, March 2004

[SynACK] SynACK, available at http://www-iepm.slac.stanford.edu/tools/synack/

[Tsunami] Tsunami Home page, available http://www.indiana.edu/~anml/anmlresearch.html

[UDT] Y. Gu, R. Grossman, UDT: An Application Level Transport Protocol for Grid Computing, PFLDNet 2004, Feb 2004, available at http://dsd.lbl.gov/DIDC/PFLDnet2004/papers/Grossman.pdf