

Limits of the HTCondor Transfer System

Edgar Fajardo^{1,*}, *Frank Würthwein*^{1,**}, *Richard Jones*^{2,***}, *Sandy Philpott*^{3,****}, and *Kurt Strosahl*^{3,†}

¹University of California San Diego, 9500 Gilman Drive, La Jolla, CA, 92093. USA.

²University of Connecticut, 2152 Hillside Road, Storrs, CT 06269-3046 Unit 3046. USA

³Thomas Jefferson National Accelerator Facility, 12000 Jefferson Avenue, Newport News, VA 23606. USA

Abstract. In the past, several scaling tests have been performed on the HTCondor batch system regarding its job scheduling capabilities. In this talk we report on a first set of scalability measurements of the file transfer capabilities of the HTCondor batch system. Motivated by the GlueX experiment needs we evaluate the limits and possible use of HTCondor as a solution to transport the output of jobs back to a submitter as a function of the number of concurrent jobs, the size of the output and the distance between a submitting node and computing node. We conclude that HTCondor transfer system satisfies the current GlueX computing demands.

1 Introduction

GlueX[1] is a high energy photoproduction experiment at Jefferson Lab. It employs a 9 GeV linearly polarized photon beam interacting in a liquid hydrogen target to produce a broad spectrum of hadronic final states. Charged and neutral particles produced as decay products from these interactions are detected in a high-acceptance magnetic spectrometer designed to identify and reconstruct exclusive final states from the lightest mesons up to the charm threshold. The foremost scientific goal of GlueX is to find definitive evidence of hybrid mesons, i.e. quark-antiquark states with excited internal gluonic degrees of freedom. Lattice QCD calculations estimate the masses of the lightest hybrid to be in the mass range 1.5-2.5 GeV/c. Past searches for these states over the past 30 years, mostly with pion beams, have found evidence for resonance behavior in a few isolated channels, but the number and pattern of mass splittings for this exotic new class of hadrons remains unclear. Resolution of this important question demands new experimental data with a complementary probe to independently verify the existence of the reported states, and search for the other members of the expected flavor multiplets. GlueX began physics data collection in February 2017, and is now nearing the completion of its initial exploratory run. Analysis of these data is now underway.

*e-mail: emfajard@ucsd.edu

**e-mail: fkw@ucsd.edu

***e-mail: richard.t.jones@uconn.edu

****e-mail: philpott@jlab.org

†e-mail: strosahl@jlab.org

The GlueX collaboration stated that in order to fulfill their physics program their Monte Carlo simulation production would require three million of CPU hours for the 2016/2017 campaign. Given the embarrassingly parallel characteristic of those Monte Carlo simulations the Grid was picked as one of the possible sources of computing resources for these workflows. The GlueX collaboration had already been using the Open Science Grid [2] to run some of the analysis workflows. However one of the challenges of the Monte Carlo workflows is to retrieve the middle products (output of the jobs) back from the compute nodes for storage and further analyzing which can be as ten times bigger in size than their analysis counterparts.

Since GlueX already had experience using GlideInWMS [3] the question to be answered is whether the HTCondor transfer system [4] can handle this load. In the HEP community tools like `gfal` and `globus-copy` have been the preferred choice in order to send large output files from grid jobs back to a server running GridFTP [5], Xrootd [6] or some other web facing interface. These tools have several disadvantages: first is that the jobs (or the experiment frameworks) are the ones in charge of sending their output back so if the stage out fails the whole job fails and it has to be manually resubmitted and secondly another external service has to be operated by the collaboration. The HTCondor transfer system takes care of re-transmits of the output (if there are network problems) and the submit host (Schedd) is the same service ran to submit the jobs so there is no overhead for the experiment. We organize this paper as follows in section 2 we quantify the computing infrastructure needs for the Monte Carlo Production. Then on Section 2.1 we translate the quantified MonteCarlo needs into a computing large scale test with their parameters. Finally we present the results of the test in section 3.

2 GlueX Monte Carlo requirements

The first step to test the capability of the HTCondor transfer to handle the load produced by GlueX MonteCarlo simulation workload is to parameterize them in terms of: number of cores, output size per job (simulations are single threaded) and average running time of a job. The reason being is that these three parameters single handily define the average write ratio (W) of output to the submit host as captured in equation 1. During the GlueX data challenge of April 2014 the appropriate parameters for the GlueX needs are summarized in Table 1. Combining the two and assuming a worst case scenario for the number of events per job and at most 20k cores available for the simulations (between GlueX owned and OSG opportunistic resources), we obtain the worst case result captured in equation 1. The size of the output of a job is calculated as 30k events per job, times 3kB per event resulting on 90MB of output per job, the length of a job is calculate as 9h times the number of seconds on an hour.

$$W(n, l, s) = \frac{nJobs * size}{length} = \frac{20000 * 90}{9 * 3600} = 55MB/s \quad (1)$$

2.1 Scaling Problem

In order to scale the HTCondor transfer system at different levels (concurrency, job output size or job length) we realize first that from equation 1 we can have shorter length jobs that are writing at higher rate or jobs writing larger files with less number of parallel jobs. For this testing we setup a submit host connected to a GlideInWMS system that used the UCSD CMS T2 sleeper pool. A sleeper pool is made out of a batch system pool in which nodes

Table 1. Computing needs of Gluex MonteCarlo simulation based on 2014 Data Challenger

Parameter	Value
Number of Events per job	[10-30]k
Input File Size (per job)	< 10MB
Output Size per Event	3kB/event
Average processing time per job	9h

Table 2. Test submit host hardware specifications

Hardware	Spec
Memory	128GB
Network Card	10Gbps Full Duplex
Core Count	40
Disk Setup	SSD + raid 6

Table 3. Parameter specification on test to saturate the NIC

Parameter	Value
Parallel Jobs	4000
Output Size per job	250MB
Achieved throughput	800 MB/s
Average job time	30 min
Expected throughput	555 MB/s

are oversubscribed with more jobs than the number of physical cores, under the premise that those extra slots will consume a negligible amount of resources(CPU, memory and disk). This configuration allows us to reach high levels of concurrency at the network level reducing the chances of many jobs landing on the same node and being bottle necked by the NIC (Network Input Controller) on a single compute node while not disturbing the Tier 2 production duties. The specifications of the testing submit node are included in Table 2.

3 Scaling Results

The first question we answered was can HTCondor take advantage of the network, i.e given enough concurrency and perfect network between compute nodes and submit nodes can HTCondor utilize the capabilities of the NIC (i.e parameters of Table 3) to transfer data from the compute nodes into the submit nodes. The positive results are seen in Figure 1. Notice that the expected write ratio W is ten times the expected worst case scenario derived in section 2 and the achieved rate is also higher than the expected worst case rate. This was due to HTCondor doing re-transmissions, however they happen fast enough there is no visible loss in performance i.e cpu efficiency is close to 99. See next Section.

3.1 Latency Measurements

Once we knew that HTCondor Transfer system achieved the required performance given ideal network conditions the next step is to see what happens in the more heterogeneous Grid environment where compute nodes are accessed via the WAN with latency's up to 80ms from the submit point, for example a submit host at the Jefferson Laboratory and a running workflow at UCSD. The metric that we study is cpu efficiency lost, defined as the total wall



Figure 1. HTCondor transfers using over $6Gbps$ at peak usage with usual ramp up and ramp down on job submission

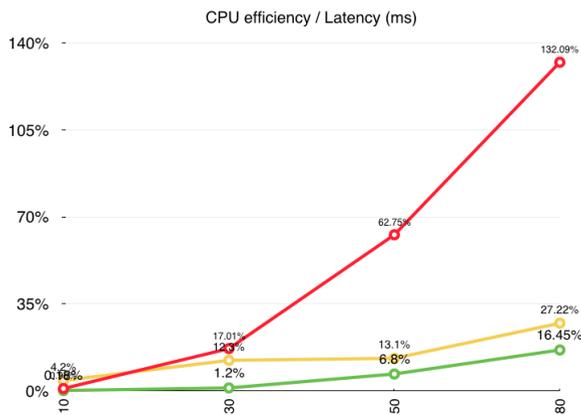


Figure 2. Ratio of transfer time to compute time against latency between compute and submit host. Cpu efficiency becomes lower when longer time is consumed in HTCondor transfers due to the latency. Red line is at twice the expected write rate W , the yellow line is at the expected rate and the green line is at the expected rate after the network kernel tuning

time a job occupied a core for transfers (input and output) over the total wall time the job actually used the core for computing. For this test we fixed parameters so we could test at: the worst case scenario for GlueX write rate ($55MB/sec$) and twice the rate at different latency levels between the compute nodes and the submit host. On our test the submit hosts and the compute nodes were located in the same computing room, so we add an artificial latency on the submit host interface. We found there is an increasing CPU efficiency loss with an increase of latency (i.e distance matters). But it can be mitigated through some kernel parameter exchanges include in table 4. The results on the HTCondor transfer system at several latency points are summarized in Figure 2.

4 Conclusions

Once proper kernel tuning is done it is easy to see that the HTCondor transfer system fits the GlueX experiment computing requirements to transfer the output back from their Monte Carlo simulations running in the Grid. However there is work to be done with the HTCondor team to understand the correlation between the latency and the loss of cpu efficiency. Such work is outside the scope of the present paper.

References

- [1] *GlueX collaboration web page*, <http://www.gluex.org/GluexX/Home.html>, accessed: 2019-02-19
- [2] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein et al., *Journal of Physics: Conference Series* **78**, 012057 (2007)
- [3] P. Mhashilkar, M. Mambelli, I. Sfiligoi, Holzman, K. Larson, J. Dost, ddbox, M. Mascheroni, J. Weigand, L. Lobato et al., *glideinwms/glideinwms: v3.4* (2018), <https://doi.org/10.5281/zenodo.1309679>
- [4] HTCondor-Team, *Htcondor 8.6.12* (2018), <https://doi.org/10.5281/zenodo.1324567>
- [5] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, I. Foster, *The Globus Striped GridFTP Framework and Server*, in *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing* (IEEE Computer Society, Washington, DC, USA, 2005), SC '05, pp. 54–, ISBN 1-59593-061-2, <https://doi.org/10.1109/SC.2005.72>
- [6] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, *XROOTD/TXNetFile: A Highly Scalable Architecture for Data Access in the ROOT Environment*, in *Proceedings of the 4th WSEAS International Conference on Telecommunications and Informatics* (World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2005), TELE-INFO'05, pp. 46:1–46:6, ISBN 960-8457-11-4, <http://dl.acm.org/citation.cfm?id=1391157.1391203>

Table 4. Kernel tuning to improve cpu efficiency lost on HTCondor transfer over the WAN

Linux Kernel Parameter	Default Value	Updated Value
<code>net.core.wmem_max</code>	212992	12582912
<code>net.core.rmem_max</code>	10485760	12582912
<code>net.ipv4.tcp_rmem</code>	4096 87380 6291456	10240 87380 12582912
<code>net.ipv4.tcp_wmem</code>	4096 16384 4194304	10240 87380 12582912
<code>net.core.netdev_max_backlog</code>	1000	250000
<code>net.ipv4.tcp_mtu_probing</code>	0	1