On Analytics of File Transfer Rates Over Dedicated Wide-Area Connections[†]

Satyabrata Sen, Nageswara S. V. Rao, Qiang Liu, and Neena Imam
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831
{*sens,raons,liuq1,imamn*}*@ornl.gov*

Rajkumar Kettimuthu and Ian Foster
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
{*kettimut,foster*}*@ornl.gov*

***Abstract*—File transfers between the decentralized storage sites over dedicated wide-area connections are becoming increasingly important in high-performance computing and big data scenarios. Designing such scientific workflows for large file transfers is extremely challenging as they depend on the file, I/O, host, and local- and wide-area network subsystems, and their interactions. To gain insights into file-transfer rate profiles, we develop polynomial, bagging, and boosting regression models for Lustre and XFS file transfer measurements, which are collected using XDD over a suite of 10 Gbps connections with 0-366 ms round trip times (RTTs). In addition to overall trends and analytics, these regressions also provide file-transfer rate estimates for RTTs and number of parallel flows at which measurements might not have been collected. They show that bagging and boosting techniques provide closer data fits than the polynomial regression. We develop probabilistic bounds on the generalization error of these methods, which combined with the cross-validation error establish that former two are more accurate estimators than the polynomial regression. In addition, we present a method to efficiently determine the number of parallel flows to achieve a peak file-transfer rate using fewer than full sweep measurements; in our measurements, the peak is achieved in 96% of cases with 15-25% of measurements of a full sweep.**

***Keywords*-Wide area transport; dedicated connections; TCP; throughput profiling; regression; cross-validation; fast probing.**

I. Introduction

Scientific workflows related to file transfers among geographically-dispersed data storages are critical for having successful collaborations among researchers in many science areas, including biology, chemistry, climate science, computing, materials science, physics, and others [1], [2]. These workflows originate due to a variety of purposes, including archiving (from compute/instrument systems to storage sites), post-processing and visualization (from storage sites to compute/visualization facilities). For example, in climate science, large collections of observational climate data and simulated data are stored in various distributed data repositories, and the Earth System Grid Federation (ESGF) grants remote access of these datasets to thousands of users for analysis and visualization purposes.

The support infrastructure is being enhanced in several ways to facilitate such data transfers. The wide-area networks, such as the Department of Energy's ESnet, provide on-demand, dedicated links [3], and high-performance filesystems, such as Lustre [4], deployed with large collections of disk drives provide site-wide access. Then, dedicated hosts, such as Data Transfer Nodes (DTNs) [5], employ tools, such as GridFTP [6] and XDD [7], to transfer the files. However, even after these developments, significant challenges remain in designing disk-to-disk file transfer workflows over wide-area connections, as they involve a complicated composition of filesystems, I/O, and local-area and wide-area network segments. Sustaining high file transfer rates requires *joint* optimization of these parameters to account for the impedance mismatches among them [8].

We measured file I/O and network throughput, and file transfer rates or throughput of Lustre and XFS file systems for a suite of seven emulated connections in the 0-366 ms RTT range. Significant statistical variations in measurements are observed due to the complex interactions of non-linear TCP dynamics with parallel file I/O streams. Consequently, it became necessary to repeat the measurements to ensure confidence in analytics based on them. To gain insights into the transfer rate profiles, we develop polynomial, bagging, and boosting regression models [9] for these measurements as functions of RTT and number of parallel flows. These regressions indicate the overall transfer rate trends as functions of various parameters, namely, monotonicity or unimodality with respect to the number of flows. In addition, they provide transfer rate estimates for configurations at which measurements have not been collected, for example, at new RTT and/or number of flows.

Our results show that both bagging and boosting techniques provide better data fits compared to the polynomial regression. Their regression functions, however, are quite different: the polynomials are smooth, namely, infinitely differentiable, whereas the other two are not differentiable at all. It makes it harder to compare them at a finer level based solely on the data fit error, particularly, in terms of their accuracy in regions where no measurements are

[†]This work was supported in part by using resources of the Computational Research and Development Programs, United States Department of Defense, and by the Net2013 and RAMSES Projects, Office of Advanced Computing Research, Department of Energy, under Contracts DE-AC05-00OR22725 and DE-AC02-06CH11357, at the Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

collected. We estimate probabilistic performance bounds on their generalization errors using the Vapnik-Chervonenkis theory [10]. Combined with the cross-validation errors, these results indicate that both bagging and boosting estimates are more accurate than the polynomial regression, and in addition they provide insights into optimal parameters needed for these regression methods.

The regression models are built using a complete full sweep measurements; however, collection of this many measurements required weeks to months of dedicated system time. The d-w method has been proposed in [11] for identifying near-optimal performance using significantly fewer measurements. By exploiting the overall unimodality of the profiles, this method implements a stochastic gradient approach using d repeated measurements over w-sized windows; in particular, this method utilizes jumps across the w-window, which does not necessarily follow the regression trends. We present in this work a regression-driven d-w method that utilizes all d measurements within w-window for gradient computation. The performance analyses of this method show that for both Lustre and XFS filesystems the peak throughput is achieved in more than 96% of cases while using only 15-25% of total measurements, and these performances in general are commensurate with those in [11], and are better in the specific case of default file I/O.

The rest of the paper is organized as follows. We summarize related work in Section II; describe network transport, file I/O subsystems, and XDD file transfers, and present an overview of the file transfer measurements with various filesystem configurations in Section III; present a detailed regression analysis to fit the measurements and derive the probabilistic bounds on the generalization error in Section IV; and describe our d-w method and its performance analysis in Section V. We conclude in Section VI.

II. Related Work

Regression techniques have been applied to analyze file transfer rates over shared [12] and dedicated connections [13], and recent advances in analytics [14] provide additional perspectives for these transfers. In particular, simple mean- and median-based models are used for GridFTP transfers in [12] and piece-wise linear regressions are used for file throughput over dedicated connections in [13]. In contrast, we apply more advanced bagging and boosting regression methods, and analytically derive the corresponding probabilistic bounds on the generalization error. There exist a number of tools for choosing parameter values to maximize file transfer performance. The GridFTP-APT project develops models that identify TCP buffer sizes and number of TCP flows for improved transfer performance, and builds tools for dynamically changing the number of connections during a file transfer [15]. Kissel, et al. [16], use similar optimizations as GridFTP-APT, but leverage the Phoebus network transport layer that includes its own

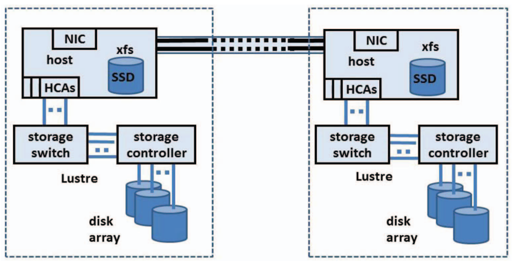


Figure 1: File transfers over long-haul connections.

dynamic network optimization scheme to improve GridFTP file transfer performance. There is also previous work on estimating optimal parallel storage system parameters [17], [18], [19]. Our work is similar but we attempt to optimize both network and storage system parameters, and use measurements rather than coarse-grain TCP behavior to estimate optimal parameters. The d-w scheme [11] is similar to the more computationally intensive techniques, such as stochastic gradient descent [20]; but our window-average implementation does not require significant amounts of training data for each source-destination pair. It can be viewed as an implementation of the stochastic approximation method [21] with derivative computation based on averaged measurements within w-window.

III. Experimental Setup and Measurements

A wide-area disk-to-disk file transfer involves reading from a source filesystem, transporting the file data across the local- and wide-area networks, and creating and writing the file contents on a destination filesystem. Therefore, the transfer process encompasses multiple subsystems, such as storage devices, data transfer hosts, and LAN and WAN connections, as illustrated in Fig. 1.

A. Experimental Setup

Measurements of file transfer rates for Lustre filesystems using XDD (described in next subsection) are collected over our testbed that includes two dedicated 48-core Linux servers and an emulated 10 Gbps connection for RTT $\tau = 0.4$, 11.6, 22.6. 45.6, 93.6, 183, and 366 ms. The network connections are emulated in hardware using ANUE-ixia devices to which host 10GigE interfaces are directly connected; see Fig. 2. The Lustre filesystem is mounted over a local InfiniBand network. The individual *throughput profiles* of network transport and disk file I/O are obtained by sweeping over the values for various system parameters. We collected TCP throughput measurements using *iperf* and the file I/O throughput using *xddprof*.

B. XDD File Transfers

A single XDD process spawns a set of threads to open a file and perform data transfers between either storage and memory or memory and the network. A set of source-XDD and destination-XDD processes are need to be paired to accomplish the file transfer. A source-XDD process generates

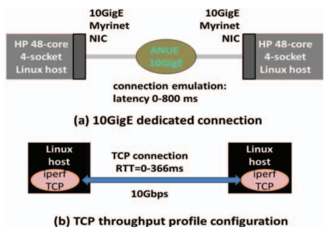

10GigE
Myrinet
NIC
10GigE
Myrinet
NIC
HP 48-core 4-socket Linux host
ANUE 10GigE
HP 48-core 4-socket Linux host
connection emulation: latency 0-800 ms
(a) 10GigE dedicated connection
Linux host
iperf TCP
TCP connection RTT=0-366ms
10Gbps
Linux host
iperf TCP
(b) TCP throughput profile configuration

Figure 2: Test configurations of emulated long-haul connections.

a *TargetThread* that opens the file, and creates a number of *QThreads* that issue read commands to fill a thread-local buffer, and subsequently transfer those buffered data over the network to a destination-XDD process. A destination-XDD process performs similar tasks in reverse order to receive data from the network and write them into the storage system. The number of source and destination QThread pairs is equal to the number of TCP parallel streams, and hence we refer to each source-destination QThread connection as a *flow*. XDD reports *read* transfer rate at the sender and *write* transfer rate at the receiver for each file transfer by aggregating across all flows.

C. File Transfer Measurements

We collected three sets of XDD disk-to-disk write file transfer measurements, for Lustre, one with the buffered I/O (the Linux default), another with the direct I/O option (avoids the local copy of the file on the host), and a third one for XFS. Each configuration measurement was repeated 10 times, and can be regarded as independent.

1) Lustre-to-Lustre Default I/O

In the default I/O Lustre setup, the number of flows varies from 1 to 8, and the number of stripes 2 and 8. A few representative throughput profiles of write transfer rates are plotted in Figs. 3(a) and 3(b). We observe that the overall throughput profiles are unimodal with respect to the number of flows. When 2 stripes are used, at lower RTTs, mean throughput peaks at 4 flows, and takes a nosedive at 6 flows. The sharp drop is progressively delayed at higher RTTs. Comparing the profiles between 2 stripes vs. 8 stripes, we notice somewhat higher transfer rates at lower RTTs with 2 flows and 4 flows when 2 stripes are used, whereas use of 8 stripes yields slightly higher rates at higher RTTs with 8 flows. However, the sharp drop in throughput, if any, occurs earlier, at 5 flows, when 8 stripes are used.

2) Lustre-to-Lustre Direct I/O

We use similar configurations for direct I/O Lustre experiments: the number of flows from 1 to 10, and the number of stripes 2 and 8. The corresponding plots are shown in Figs. 3(c) and 3(d). The throughput profiles show monotonically increasing trends with respect to the number of flows. Comparing the performances of 2 vs. 8 stripes, we notice that the use of 2 stripes yields somewhat higher transfer rates for lower flow counts. With more flows, overall throughput is higher, and 8 stripes is the better option. Also, the peak transfer rates with 10 flows and 8 stripes demonstrate a significant improvement over the default I/O Lustre counterparts.

3) XFS

For XFS, the number of flows varies from 1 to 10, and the throughput profile plot is shown in Fig. 3(e) for various RTTs. Similar to the Lustre direct I/O configuration, the overall throughput exhibits predominantly increasing trends with respect to the number of flows; although compared to the former, the throughput increases much faster with increasing flow counts in lower RTT cases.

IV. Regression Analysis

In this section, we first describe three regression techniques, namely, polynomial, bagging, and boosting, applied to file transfer rate measurements of previous section, and derive probabilistic bounds for their generalization error.

A. Polynomial Regression

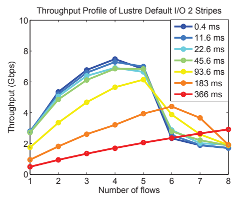
To model the nonlinear relationship between the predictors and response, the polynomial regression method is the most simple approach, which includes polynomial functions of the predictors in the regression model [9]. In general, considering the number of flows (n) and RTT (τ) as two predictors, a polynomial regression model $f_{\text{poly}}(\cdot)$ of file transfer rate or throughput that includes the *interaction terms* of the predictors is given by

$$\begin{aligned} f_{\text{poly}}(n,\tau) &= \beta_0+\beta_1 n+\beta_2\tau+\beta_3 n^2+\beta_4 n\tau \\ &\quad +\beta_5\tau^2+\cdots+\beta_{M(M+1)/2+M-1}n\tau^{M-1} \\ &\quad +\beta_{M(M+1)/2+M}\tau^M, \end{aligned}$$

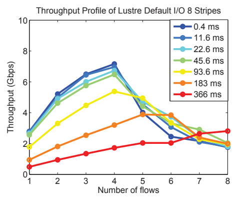
where M denotes the degree of the polynomial and βs are the regression coefficients or parameters. Now, since this model is still a *linear model* with predictors n, τ, $n\tau$, n^2, τ^2, etc., we can use standard least squares linear regression to estimate the regression coefficients βs and produce an overall nonlinear fit $\hat{f}_{\text{poly}}(\cdot)$.

To select the polynomial degree M, we apply the k-fold cross-validation method. In this approach, the whole throughput measurement set is randomly divided into k subsets, or folds, of approximately equal size. Among these k folds, the first fold is considered as a validation set, and the regression training is applied on the remaining $k-1$ folds altogether. The cross-validation mean-squared error (CVMSE_1) is then computed with respect to the held-out validation fold. This procedure is repeated k times with a different fold of observations as a validation set in each time. Finally, the overall cross-validation error is computed by averaging $\{\text{CVMSE}_i, i=1,2,\ldots,k\}$.

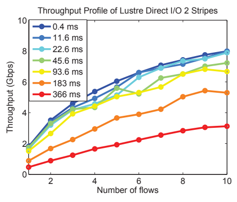
Fig. 4 summarizes the results of cross-validation approach when applied to both the Lustre default I/O and Lustre direct I/O datasets. In case of Lustre default I/O with 2 stripes the cross-validation root mean-squared error (CVRMSE) monotonically decreases with the increase in model order,



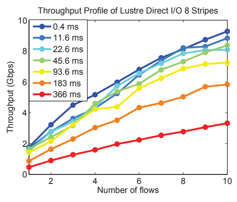
(a) default I/O, 2 stripes



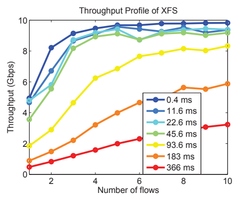
(b) default I/O, 8 stripes



(c) direct I/O, 2 stripes



(d) direct I/O, 8 stripes



(e) XFS

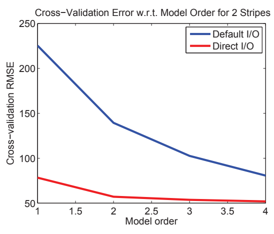
Figure 3: Mean throughput profiles of Lustre default I/O, Lustre direct I/O, and XFS file write transfer rates.

but the amount of such decrease becomes progressively smaller at higher model order; see Fig. 4(a). Therefore, it seems that a polynomial with degree $M = 3$ or 4 is required to fit the 2-striped Lustre default I/O data. On the other hand, the CVRMSE of Lustre direct I/O with 2 stripes remains almost constant to its value at $M = 2$ as model order is increased. Hence, a polynomial model with degree $M = 2$ would be sufficient for the 2-striped Lustre direct I/O dataset. From Fig. 4(b) we notice similar performances of Luster default I/O and direct I/O with 8 stripes.

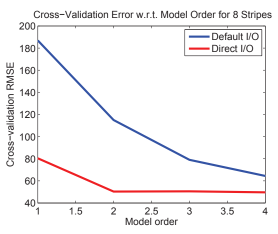
In addition, from the training data portions of the cross-validation method, we compute the adjusted-R^2 statistic, defined as adjusted-$R^2 = 1 - [\text{RSS}/(P - L - 1)]/[\text{TSS}/(P - 1)]$, where RSS is the residual sum of squares of the fit, TSS is the total sum of squares of the data, P is the size of data, and L is the number of predictors in the model. Unlike the usual R^2 statistic that always increases as more predictors are added to the regression model, the adjusted-R^2 metric pays a price for the inclusion of unnecessary predictors in the regression model. In Figs. 4(c) and 4(d), we respectively show the variations of the adjusted-R^2 value with respect to M for both 2 and 8 stripes Luster default I/O and direct I/O datasets. These results also corroborate that, to fit the Lustre default I/O and direct I/O datasets, we require $M = 3$ (or 4) and $M = 2$, respectively.

In Fig. 4, we also depict the variations in the number of statistically significant predictors, whose p-value < 0.00001, with respect to the model order. A small p-value indicates that there is an association between the predictor and the response. From Figs. 4(e) and 4(f), we observe that for the Lustre default I/O the number of significant predictors peaks at $M = 3$ for both the 2-striped and 8-striped datasets; at $M = 4$ the number remain the same for the 2-striped dataset only. For Lustre direct I/O, at $M = 2$ the number of significant predictors becomes the maximum and equals to the total number of predictors. Therefore, we reiterate that the polynomial models with $M = 3$ and $M = 2$ respectively fit the Lustre default I/O and direct I/O datasets.

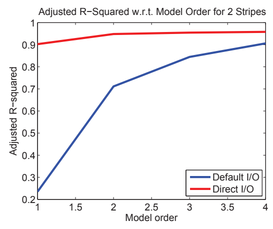
The polynomial regression fits with $M = 3$ and $M = 2$ respectively to the Lustre default I/O and direct I/O are shown in Fig. 5, along with the measured datasets. For the Lustre default I/O, we show the throughputs with respect to RTT for 1 flow and 4 parallel flows; whereas for the Lustre direct I/O, we depict the same for 1 flow and 10 flows. It is



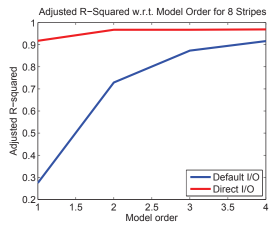
(a) CV RMSE, 2 stripes



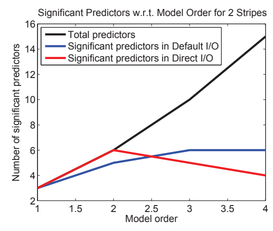
(b) CV RMSE, 8 stripes



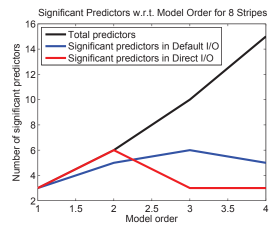
(c) adjusted R^2, 2 stripes



(d) adjusted R^2, 8 stripes



(e) significant predictors, 2 stripes



(f) significant predictors, 8 stripes

Figure 4: Results of cross-validation analysis with respect to polynomial regression model order for the Lustre default and direct I/O file write transfer rates.

evident that the polynomial regression model fits the Lustre direct I/O relative better than the Lustre default I/O.

B. Bagging

To capture the nonlinear interactions among the predictors, an alternative approach to the polynomial regression (with interaction terms) is the decision tree based methods. Bagging is one of such powerful regression models that uses decision trees as the basic building blocks [9]. In this approach, B different trees $f_b(\cdot)$ are trained on B bootstrapped datasets $\{n_b, \tau_b\}$, which are obtained by taking repeated samples from the original training set $\{n, \tau\}$. Each of these trees are grown very deep, and are not pruned. Hence, each individual tree overfits each bootstrapped data and results into high variance but low bias. As averaging

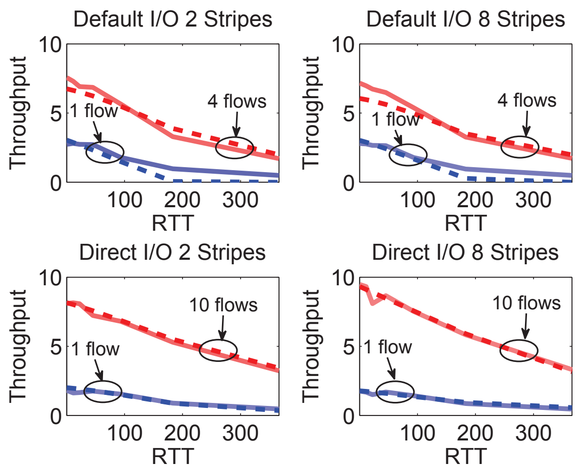


Figure 5: Results of polynomial regression fits on the Lustre default and direct I/O file write transfer rates: solid lines – measured data, broken lines – fitted data.

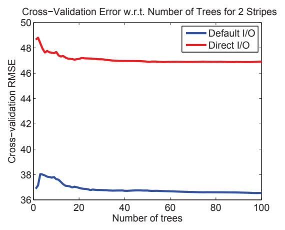
is a natural way to reduce variance, the bagging regression model is formulated by taking the average of all trees as

$$f_{\text{bag}}(n, \tau) = \frac{1}{B} \sum_{b=1}^{B} f_b(n_b, \tau_b).$$

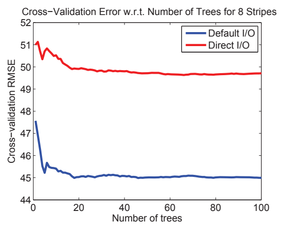
In Fig. 6, we show various performance measures of the bagging regression method with respect to the number of trees and the relative importance of the predictors. Figs. 6(a) and 6(b) depict the cross validation root mean-squared error as a function of B respectively for the 2-striped and 8-striped Luster default and direct I/O datasets. In the bagging context, this error is also known as the out-of-bag (OOB) error, which is calculated using the OOB observations, i.e., the remaining observations not used to train a particular bagged tree. We observe that the OOB error initially decreases and then becomes almost constant as we increase B. In both the 2-striped and 8-striped scenarios, the Lustre default I/O has smaller OOB error than that of the Lustre direct I/O.

The variations of the R^2 statistic with respect to the number of trees are shown in Figs. 6(c) and 6(d) respectively for the 2-striped and 8-striped Luster default and direct I/O datasets. In both cases, the R^2 values remain almost constant, except for slight increments at small values of B. When 2 stripes are used, the R^2 statistic of the Lustre default I/O is larger than that of the Lustre direct I/O; whereas that trend gets reversed when 8 stripes are used.

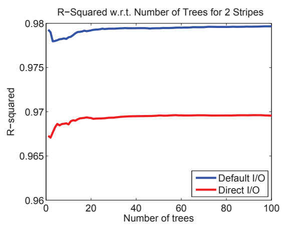
The relative importance of the predictors (number of flows and RTT) are depicted in Figs. 6(e) and 6(f). To calculate this measure, we first record the total amount of decrease in RSS due to splits in decision trees over a specific predictor, and then take average over all B trees. Therefore, a large value associated with a predictor signifies a large decrease in RSS due to splits with respect to that predictor, and



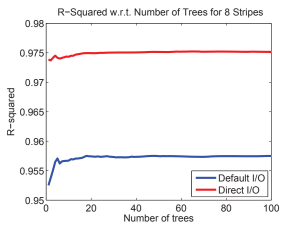
(a) CV RMSE, 2 stripes



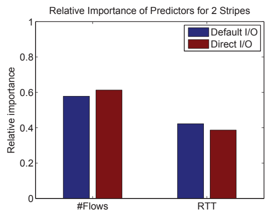
(b) CV RMSE, 8 stripes



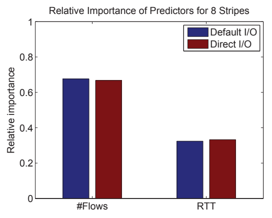
(c) R^2 statistic, 2 stripes



(d) R^2 statistic, 8 stripes



(e) relative importance, 2 stripes



(f) relative importance, 8 stripes

Figure 6: Results of cross-validation analysis of bagging regression method for the Lustre default and direct I/O file write transfer rates.

hence indicates that particular predictor is important. From this perspective, the number of flows appears to be more important predictor than RTT for both the 2-striped and 8-striped Lustre default I/O and direct I/O datasets.

In addition, in Fig. 7, we demonstrate the bagging regression fits with $B = 50$ for the Lustre default I/O and direct I/O alongside the measured throughput variations. As before, we show the throughput profiles for 1 flow and 4 flows of the Lustre default I/O; and those for 1 flow and 10 flows for the Lustre direct I/O. These bagged regressions show improved fits than the polynomial regressions.

C. Boosting

Another regression approach that involves decision trees as the core technique is the boosting method [9]. Unlike bagging, where each tree is grown independently on a bootstrapped data, in boosting, the trees operate on the original training data and they are built sequentially using the residual information from the previously grown trees. The motivation behind such an approach is that each tree, typically having a small depth, when fitted to the residuals can improve the overall predictions in areas where the previously grown trees did not perform well. Hence, the boosting method follows the philosophy of learning slowly to improve performance. Mathematically, the boosting regression model

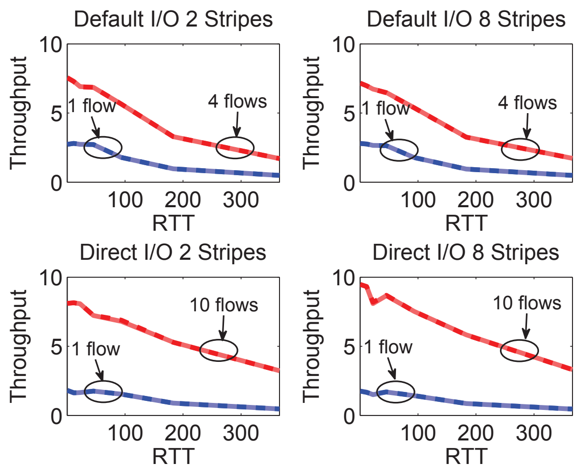


Figure 7: Results of bagging regression fits on the Lustre default and direct I/O file write transfer rates: solid lines – measured data, broken lines – fitted data.

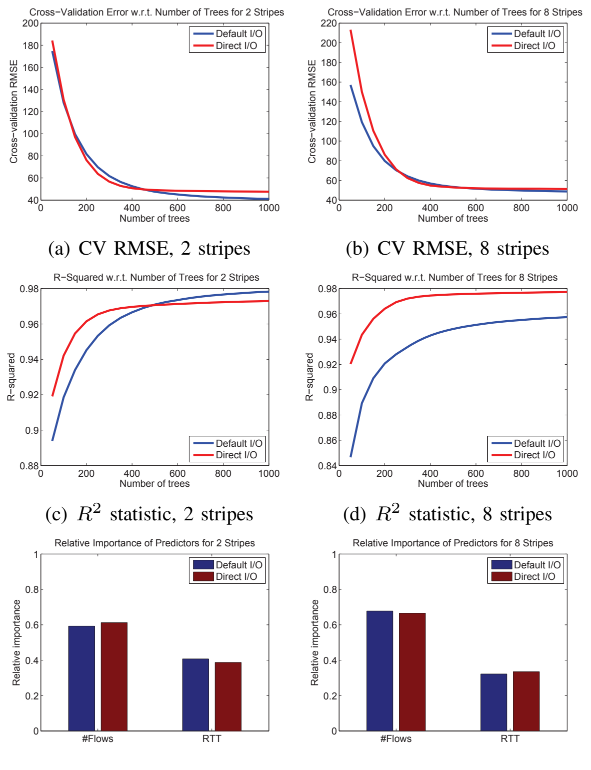
with B decision trees can be written as

$$f_{\mathrm{boost}}(n,\tau)=\sum_{b=1}^{B}\lambda f_b(n,\tau;R_b),$$

where λ is known as the shrinkage parameter that controls the rate at which the boosting method learns; and R_b is the residual values applied to the bth tree. At the beginning, R_b is initialized with the measured throughput values, i.e., $R_1 = Y$, and then it is sequentially evaluated as $R_b = R_{b-1} - \lambda\hat{f}_{b-1}(n,\tau;R_{b-1})$.

Fig. 8 demonstrates the performance of the boosting method for both the Lustre default I/O and Lustre direct I/O datasets. The cross-validation root mean squared errors show monotonic decrease as the number of trees is increased for both the 2-striped and 8-striped Lustre datasets; see respectively Figs. 8(a) and 8(b). We notice that, for the Lustre direct I/O, the cross-validation errors become almost constant when the number of trees is increased beyond, say, $B = 500$; whereas the cross-validation errors for the Lustre default I/O continue to decrease even when $B = 1000$. Furthermore, at higher B values, the cross-validation error of Lustre default I/O is smaller than that of direct I/O.

The variations of R^2 statistic with respect to B are plotted in Figs. 8(c) and 8(d) respectively for the 2-striped and 8-striped Luster default and direct I/O measurements. As observed in the cross-validation performance, the values of R^2 statistic for the Lustre direct I/O become almost constant at $B = 500$ and above; whereas R^2 statistic for the Lustre default I/O always increases with B. Similar to the bagging method, at higher B values the R^2 statistic of the Lustre default I/O is found to be larger than that of the Lustre direct I/O when 2 stripes are used, and the relationship is flipped when 8 stripes are used.



(a) CV RMSE, 2 stripes (b) CV RMSE, 8 stripes

(c) R^2 statistic, 2 stripes (d) R^2 statistic, 8 stripes

(e) relative importance, 2 stripes (f) relative importance, 8 stripes

Figure 8: Results of cross-validation analysis of boosting regression method for the Lustre default and direct I/O file write transfer rates.

The barplots in Figs. 8(e) and 8(f) showing the relative importance of the predictors demonstrate very similar performances as in bagging. These plots again suggest that, for both the 2-striped and 8-striped Lustre datasets, the number of parallel flows is a more important predictor than RTT in terms of the amount of RSS reduction in the regression fit.

The boosting regression fits with $B = 500$ to the Lustre default I/O and direct I/O are shown in Fig. 9, along with the measured throughput values. As before, we show the boosting fits to throughput profiles for 1 flow and 4 flows of the Lustre default I/O; and those for 1 flow and 10 flows for the Lustre direct I/O. Overall, the performance of these boosting fits seems to be in between those of the polynomial and bagging regression fits.

In addition, we repeat the above regression analysis for the XFS configuration, and the results for bagging and boosting techniques are shown in Fig. 10. The observations we gather from these plots largely conform with those that we have seen in both Lustre configurations as well; namely, bagging in general leads to overall better fits than boosting, and with large number of trees their performances are almost similar.

D. Generalization Bounds

The regression models $f_a, a = \mathrm{poly}, \mathrm{bag}, \mathrm{boost}$, are useful in identifying the overall throughput trends, that is, the

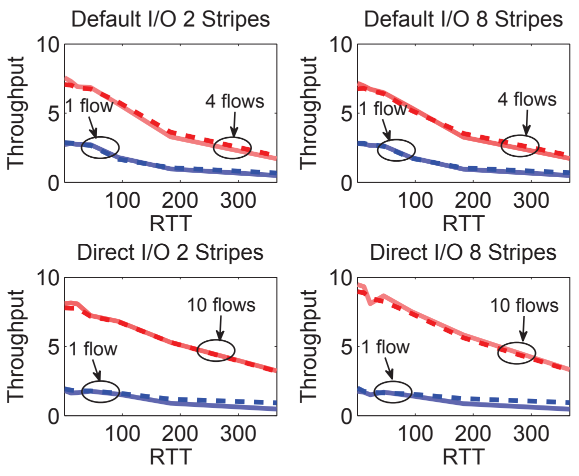


Figure 9: Results of boosting regression fits on the Lustre default and direct I/O file write transfer rates: solid lines – measured data, broken lines – fitted data.

monotonic decrease with respect to RTT τ and unimodality with respect to the number of parallel flows n. From a practical perspective, collecting measurements at a new RTT might incur a significant cost of establishing a new long-haul connection. If such a connection is available, however, the measurements at a new n value involves only executing the file transfer code. Collecting measurements in either case is time-consuming (even with emulators) because files need to be physically transferred over the connection, and furthermore the transfer times increase with RTT. As such, the regression models can be used to provide point throughput estimates for given n and τ, particularly, for values at which measurements have not been collected. In addition, the regression function f_a can be used to identify the optimal n^* to achieve the maximum throughput for a given τ, for example, by using estimated gradients $\frac{df_a(n,\tau)}{dn}$. Once regression is estimated based on prior measurements, n^* can be estimated computationally; typically, measurements at a large number of parameter setups are needed to ensure such estimation accuracy. At the other extreme, a connection with a given RTT τ may be set up and measurements at several n values collected, and in the next section we present a method that avoids a complete sweep of n values.

In either case, the effectiveness of this method critically depends on the accuracy of the throughput estimates, which in turn depends on the regression function and its empirical error RSS. We analyze these methods using the Vapnik-Chervonenkis theory [10], which provides a basis for their performance comparison, and indeed confirms the empirical observations from the previous subsection.

The file transfer throughput $\theta_{n,\tau}$, using n flows over a connection with RTT τ, is a random variable. Its distribution $\mathbf{P}_{\theta_{n,\tau}}$ is quite complex since it depends on file, host and network subsystems and their interactions. We define the *throughput regression* as

$$\bar{\theta}(n,\tau) = E\left[\theta(n,\tau)\right] = \int \theta(n,\tau) d\mathbf{P}_{\theta_{n,\tau}},$$

which depends on generally unknown $\mathbf{P}_{\theta_{n,\tau}}$. It can be estimated using l_k measurements $\theta(n,\tau_k,j)$, $j=1,2,\ldots,l_k$ collected over connections with RTT τ_k, $k=1,2,\ldots,m$. Consider an estimator f for the regression $\bar{\theta}(n,\tau)$ chosen from a function class $\mathcal{F}$, for example, polynomials. The *generalization error* $I(f)$ of such an estimator f is

$$I(f) = \int \left[f(n,\tau) - \theta(n,\tau)\right]^2 d\mathbf{P}_{\theta_{n,\tau}},$$

and the *best estimator* f^* is given by $I(f^*) = \min_{f\in\mathcal{F}} I(f)$. The *best empirical estimator* $\hat{f} \in \mathcal{F}$ minimizes the empirical error RSS

$$\hat{I}(f) = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{l_k}\sum_{j=1}^{l_k}\left[f(n,\tau_k) - \theta(n,\tau_k,j)\right]^2,$$

that is, $\hat{I}(\hat{f}) = \min_{f\in\mathcal{F}} \hat{I}(f)$. Here, $\hat{f}$ is best sample-based approximation of ideal f^* for which the Vapnik-Chervonenkis theory [10] provides the following performance guarantees. Let $l = \sum_{k=1}^{m} l_k$ denote the total number of measurements. Then, we have

$$\begin{aligned}&\mathbf{P}\left\{I\left(\hat{f}\right) > I(f^*) + \epsilon\right\}\\ &\leq \mathbf{P}\left\{\max_{h\in\mathcal{F}} \left|I\left(h\right) - \hat{I}(h)\right| > \epsilon/2\right\}\\ &\leq 16\mathcal{N}_\infty\left(\frac{\epsilon}{C},\mathcal{F}\right) l e^{-\epsilon^2 l/(4C)^2}\end{aligned}$$

where $\theta_{n,\tau} \leq C$, and $\mathcal{N}_\infty(\epsilon,\mathcal{F})$ is the ϵ-cover of $\mathcal{F}$ under L_∞ norm. The regression f_a is chosen from class $\mathcal{F}_a$, and f_a^* and $\hat{f}_a$ are the expected and empirical best estimates.

For polynomial regression, functions of $\mathcal{F}_a$ have a bounded Lipschitz constant $\mathcal{L}$, given by the maximum gradient, and their ϵ-cover is bounded as [22]:

$$\mathcal{N}_\infty(\epsilon,\mathcal{F}_{\text{poly}}) < \frac{2\mathcal{L}}{\epsilon} 2^{\left[\frac{\mathcal{L}^2}{\epsilon^2}\right]},$$

when τ and n are scaled to the range $[0,1]$. Thus, the expected error $I(\hat{f}_{\text{poly}})$ of the profile mean is within ϵ of the optimal error $I(f^*_{\text{poly}})$ with a probability that increases with l, and is independent of the underlying distributions.

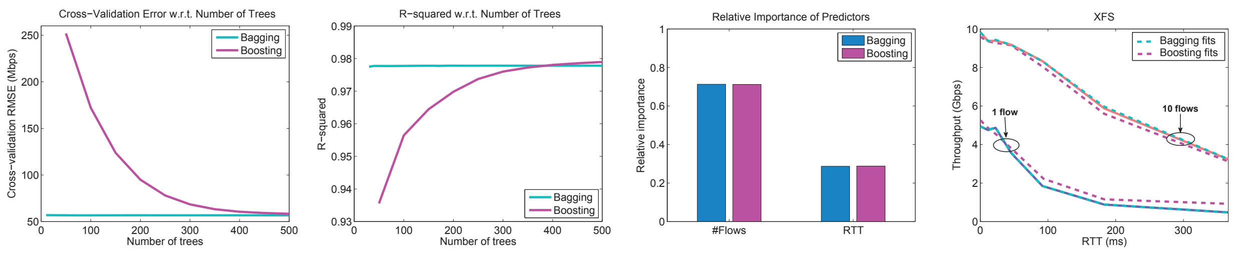
The functions in $\mathcal{F}_a$ used for bagging and boosting regression estimates have the total variation upper bounded by $2C$, which provides us the upper bound ([23], p. 175):

$$\mathcal{N}_\infty\left(\frac{\epsilon}{C},\mathcal{F}_a\right) < 2\left(\frac{l}{\epsilon^2}\right)^{(1+C/\epsilon)\log_2(2\epsilon/C)}$$

By using this bound, we obtain

$$\begin{aligned}&\mathbf{P}\left\{I\left(f_a\right) > I(f_a^*) + \epsilon\right\}\\ &< 32\left(\frac{l}{\epsilon^2}\right)^{(1+C/\epsilon)\log_2(4\epsilon/C)} l e^{-\epsilon^2 l/(2C)^2}\end{aligned}$$

As before, the expected error $I(\hat{f}_a)$ is within ϵ of the optimal error $I(f_a^*)$ with a probability that increases with l.



(a) CV RMSE (b) R^2 statistic (c) relative importance (d) regression fits

Figure 10: XFS regression performance.

The regression estimator f_a is not in general guaranteed to minimize the expected error $I(.)$ or the empirical error $\hat{I}(.)$. In particular, consider that $\hat{I}(f_a) = \hat{I}(\hat{f}_a) + \hat{\epsilon}_a$ for $\hat{\epsilon}_a \geq 0$, and that with probability at most δ_a, we have the condition $\mathbf{P}\left\{\max_{h\in\mathcal{F}_a} |I(h) - \hat{I}(h)| > \epsilon/2\right\}$ as described above. Then, with probability at least $1-\delta_a$, we have

$$\begin{aligned} I(f_a) &\leq \hat{I}(f_a) + \epsilon/2 = \hat{I}(\hat{f}_a) + \hat{\epsilon}_a + \epsilon/2 \\ &\leq \hat{I}(f_a^*) + \hat{\epsilon}_a + \epsilon/2 \leq \hat{\epsilon}_a + I(f_a^*) + \epsilon \end{aligned}$$

Now, since $\hat{I}(\hat{f}_a) \geq 0$, we have $\hat{\epsilon}_a \leq \hat{I}(f_a)$. Thus, with probability at least $1-\delta_a$, we have the condition $I(f_a) \leq \hat{I}(f_a) + I(f_a^*) + \epsilon$, which can be equivalently stated as

$$\mathbf{P}\left\{I(f_a) > \hat{I}(f_a) + I(f_a^*) + \epsilon\right\} \leq \delta_a.$$

This condition indicates that the generalization error of f_a depends on two main factors: (a) its empirical error $\hat{I}(\hat{f}_a)$ which is reflected in CVRMSE of f_a, and (b) the best possible error $I(f_a^*)$ achievable by the class $\mathcal{F}_a$ itself. We note that the above three estimator classes have strong approximation properties which make the second term small. Consequently, the generalization error is mainly determined by the data fit, as confirmed by the lower cross-validation errors described in the previous subsections. Furthermore, increasing the number of parameters of f_a, such as the degree of polynomials and number of regression trees, beyond a certain limit does not improve its generalization, as indicted by a small or no reduction in CVRMSE.

V. Fast Probing Method for Peak Transfer Rate

In this section, we describe a fast probing depth-width (d-w) method and analyze its performance characterization.

A. D-W Method

The basic idea of our d-w method is to exploit the observed unimodality of transfer rate measurements with respect to the number of flows. The monotonicity property of Lustre direct I/O is a special case of unimodality. We utilize a stochastic gradient search method that starts with the largest number of flows, and continues to strategically probe multiple configurations (e.g., number of flows) with the objective of identifying the peak throughput with the fewest probes. According to Algorithm 1, at each configuration of width w, we collect d repeated measurements and compute their averages. Using those w averaged throughput values,

Algorithm 1 D-W Probing Algorithm

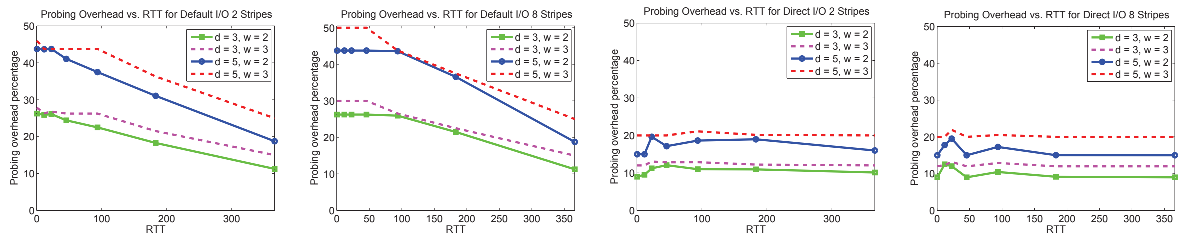
```
1:  initialize: $i = max\,flow$
2:  while $i > w$ do
3:      $w$-width configuration: $P_i = \{i-w+1, \ldots, i-1, i\}$
4:      $d$ measurements: $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,d}\}$ $\forall i \in P_i$
5:      $\bar{Y}_i = \{\bar{y}_{i-w+1}, \ldots, \bar{y}_i\}$, with $\bar{y}_i = (1/d)\sum_{j=1}^{d} y_{i,j}$
6:      $\bar{S}_i = (1/\binom{w}{2})\sum_{k=1}^{\binom{w}{2}} s_k$, where $s_k = slope(\bar{y}_{i1}, \bar{y}_{i2})$,
            $\bar{y}_{i1}, \bar{y}_{i2} \in \bar{Y}_i, i1 \neq i2$
7:      if $(\bar{S}_i < 0)$ and $(\bar{S}_{i-1} < 0)$ then
8:          $P_{\text{opt}} = P_{i-1}$
9:          $i_{\text{opt}} = \arg\max \bar{Y}_{i-1}$
10:         $y_{\text{peak}} = \max y_{i_{\text{opt}}}$
11:         break
12:     $i \leftarrow i - 1$
```

we evaluate the mean-slope of that configuration by averaging $\binom{w}{2}$ pairwise slopes of the averaged throughput values. We keep on shifting to a new probing configuration by decreasing the number of flows by one until two consecutive configurations result in negative mean-slopes. This stopping criterion reduces the risk of premature termination due to variability of insufficient observations. Finally, from the selected configuration, the setting with the highest average-throughput is chosen as the optimal probed setting.

Potentially there are different choices of d, w values. The larger the d value, the greater is the chance of correctly identifying the peak transfer rate, but at the cost of extra probing overhead. On the other hand, a large w value helps to smooth out any local variations, but also increases probing overhead by including additional configurations.

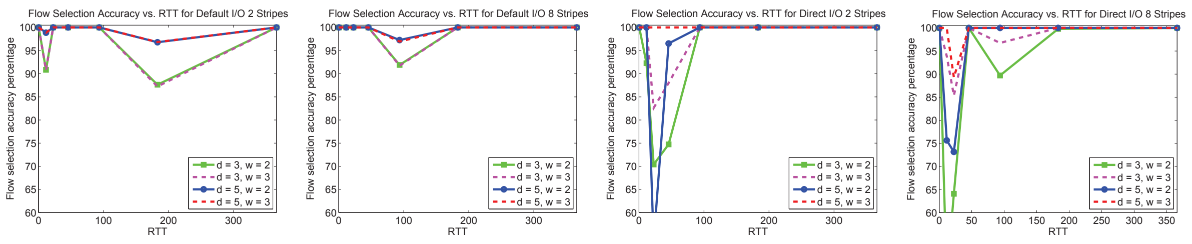
B. Performance Characterization

We characterize the overall performance of the d-w method in terms of *probing overhead* and *probing accuracy*. We express probing overhead (in percentage) as the ratio between the number of probes actually performed by the d-w method to the complete configuration sweep. To characterize probing accuracy, we utilize: (i) flow selection accuracy and (ii) peak throughout accuracy. We evaluate the flow selection accuracy as the fraction of times the d-w method correctly identifies the flow at which we actually observed the peak



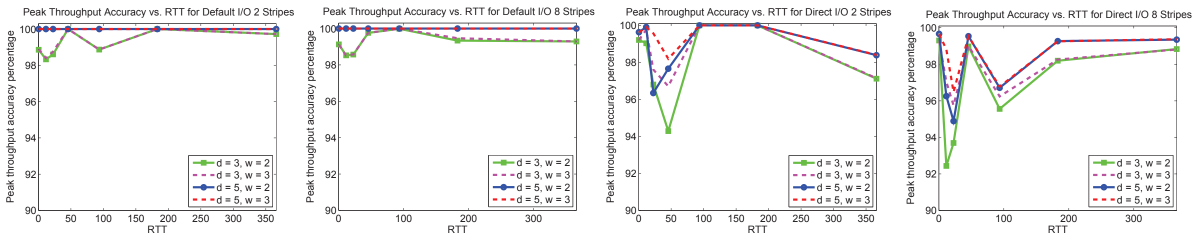
(a) default I/O, 2 stripes (b) default I/O, 8 stripes (c) direct I/O, 2 stripes (d) direct I/O, 8 stripes

Figure 11: Results of d-w method: Variations of probing overhead with respect to RTT.



(a) default I/O, 2 stripes (b) default I/O, 8 stripes (c) direct I/O, 2 stripes (d) direct I/O, 8 stripes

Figure 12: Results of d-w method: Variations of flow selection accuracy with respect to RTT.



(a) default I/O, 2 stripes (b) default I/O, 8 stripes (c) direct I/O, 2 stripes (d) direct I/O, 8 stripes

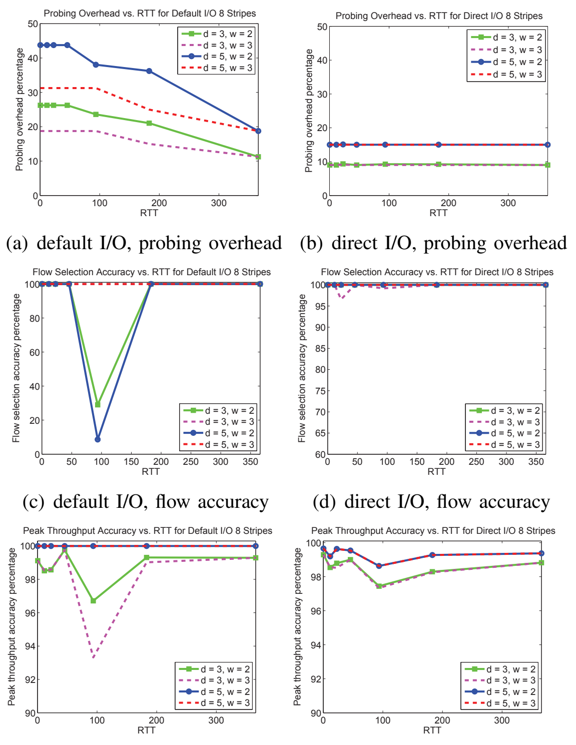
Figure 13: Results of d-w method: Variations of peak throughput accuracy with respect to RTT.

transfer rate. The peak throughput accuracy is computed as the ratio between the peak throughput returned by the d-w method to the measured peak throughput.

The performance analyses of the d-w method are shown in Figs. 11-13 for both 2-striped and 8-striped Lustre default I/O and direct I/O file transfer rate measurements. For these analyses, we use (d,w) = (3,2), (3,3), (5,2), (5,3). For the Lustre default I/O, the probing overhead decreases as RTT is increased, because the configuration having peak transfer rate changes from 4 flows at lower RTTs to 8 flows at the maximum RTT. Also, the probing overhead increases with increasing d and w values. From the analyses of flow selection and peak throughput accuracies, we notice that a change in w value does not affect the performance if d value is kept fixed, but a change in d value improves the accuracy performance. Overall, we get more than 90% flow selection accuracy (except at one RTT for 2 stripes) and more than 98% peak selection accuracy. Therefore, a method with (d,w) = (3,2), which corresponds to the lowest probing overhead, appears reasonable for the Lustre default I/O.

For the Lustre direct I/O, the probing overhead remains almost constant at different RTTs; this is because the peak throughput in general occurs at the maximum 10 flows for all the RTTs. Also, with the increase in the d and w values, the probing overhead increases expectedly. The analyses on probing accuracy show more variations than those observed with the Lustre default I/O. Particularly, an increase in w value considerably improves the flow selection and peak throughput accuracies at the lower RTTs. Overall, keeping $w = 3$ ensures more than 85% flow selection accuracy and more than 96% peak selection accuracy. In addition, an increase in d value also improves the accuracy performances. Therefore, to keep the probing overhead low, a method with (d,w) = (3,3) seems appropriate for the Lustre direct I/O.

In addition, we also compare the performance of our d-w method with the one proposed in [11]. Whereas d has the very same meaning in the other method, w does not denote the width of the sliding window there; rather, it describes the interval between probed configurations (e.g., $w = 2$ means every other configuration is jumped over). Select probing results of this alternate d-w implementation for Lustre 8 stripes are plotted in Fig. 14. These results are largely commensurate with those shown in Figs. 11-13, although it appears the d-w method of this work yields somewhat



(a) default I/O, probing overhead (b) direct I/O, probing overhead

(c) default I/O, flow accuracy (d) direct I/O, flow accuracy

(e) default I/O, throughput accuracy (f) direct I/O, throughput accuracy

Figure 14: Results of d-w method of [11] for 8-striped Lustre default and direct I/O file write transfer rates.

better throughput accuracy performance for default I/O, and the other method leads to higher accuracy for direct I/O.

VI. Conclusions

We presented disk-to-disk file transfer measurements for Lustre and XFS filesystems over emulated dedicated connections for a wide range of RTTs, and provided regression analysis to quantify the relationship between these file transfer rates and the corresponding system parameters, such as RTT and number of parallel flows. These results along with the probabilistic bounds on the generalization error of the regression methods provide valuable insights needed to predict the performance of file transfers. In addition, large variations in measured transfer rates indicate that repeated measurements are often necessary to ensure estimation confidence. Our gradient-descent based d-w method avoids the need for time-consuming full sweeps of all parameter combinations by probing a small number of measurements to identify configurations that achieve peak rates. Future directions include detailed analytical development of the d-w method and its variants in terms of performance guarantees. It will also be of interest to develop methods that dynamically adapt d and w values based on measurements.

References

[1] "Software defined networking for extreme-scale science: Data, compute, and instrument facilities," Bethesda, MD, DOE ASCR Workshop Rep., Aug. 5–6, 2014.

[2] "ASCR network requirements review," Germantown, MD, ESnet Rep., Apr. 22–23, 2015.

[3] "On-demand secure circuits and advance reservation system," http://www.es.net/oscars.

[4] Lustre Basics, https://www.olcf.ornl.gov/kb_articles/lustre-basics.

[5] Science DMZ: Data Transfer Nodes, https://fasterdata.es.net/science-dmz/DTN.

[6] "Gt 4.0 gridftp," http://www.globus.org.

[7] XDD - The eXtreme dd toolset, https://github.com/bws/xdd.

[8] B. W. Settlemyer, J. D. Dobson, S. W. Hodson, J. A. Kuehn, S. W. Poole, and T. M. Ruwart, "A technique for moving large data sets over high-performance long distance networks," in *Mass Storage Systems and Technologies (MSST), IEEE 27th Symposium on*, May 2011, pp. 1–6.

[9] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

[10] V. N. Vapnik, *Statistical Learning Theory*. New York: John-Wiley and Sons, 1998.

[11] N. S. V. Rao, Q. Liu, S. Sen, G. Hinkel, N. Imam, I. Foster, R. Kettimuthu, B. Settlemyer, C. Q. Wu, and D. Yun, "Experimental analysis of file transfer rates over wide-area dedicated connections," in *18th IEEE Intl Conf. High Performance Comput. Comm. (HPCC)*, Sydney, Australia, Dec. 2016.

[12] S. Vazhkudai and J. M. Schopf, "Using regression techniques to predict large data transfers," *Int. J. High Perform. Comput. Appl.*, vol. 17, no. 3, pp. 249–268, Aug. 2003.

[13] B. W. Settlemyer, N. S. V. Rao, S. W. Poole, S. W. Hodson, S. E. Hicks, and P. M. Newman, "Experimental analysis of 10Gbps transfers over physical and emulated dedicated connections," in *International Conference on Computing, Networking and Communications*, Jan. 2012, pp. 845–850.

[14] P. Raj, A. Raman, D. Nagaraj, and S. Duggirala, *High-Performance Big-Data Analytics: Computing Systems and Approaches*. Springer-Verlagl, 2015.

[15] T. Ito, H. Ohsaki, and M. Imase, "Automatic parameter configuration mechanism for data transfer protocol GridFTP," in *Intl Symp. Applications and the Internet*, 2006, pp. 32–38.

[16] E. Kissel, M. Swany, and A. Brown, "Improving GridFTP performance using the Phoebus session layer," in *Proc. Supercomputing Conf.*, 2009.

[17] B. Behzad, S. Byna, M. Snir *et al.*, "Pattern-driven parallel I/O tuning," in *Proc. 10th Parallel Data Storage Work.*, 2015, pp. 43–48.

[18] N. Liu, C. Carothers, J. Cope, P. Carns, R. Ross, A. Crume, and C. Maltzahn, "Modeling a leadership-scale storage system," in *Parallel Processing and Applied Mathematics*, 2011, pp. 10–19.

[19] P. H. Carns, B. W. Settlemyer, and W. B. Ligon III, "Using server-to-server communication in parallel file systems to simplify consistency and improve performance," in *Proc. ACM/IEEE Conf. Supercomputing*, 2008, pp. 1–8.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

[21] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley Pub, 2003.

[22] N. S. V. Rao and V. Protopopescu, "Function estimation by feedforward sigmoidal networks with bounded weights," *Neural Processing Letters*, vol. 7, pp. 125–131, 1998.

[23] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.