

Projecting Disk Usage Based on Historical Trends in a Cloud Environment

Murray Stokely
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
mstokely@google.com

Amaan Mehrabian
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
amaan@google.com

Christoph Albrecht
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
calbrecht@google.com

François Labelle
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
flab@google.com

Arif Merchant
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA
aamerchant@google.com

ABSTRACT

Provisioning scarce resources among competing users and jobs remains one of the primary challenges of operating large-scale, distributed computing environments. Distributed storage systems, in particular, typically rely on hard operator-set quotas to control disk allocation and enforce isolation for space and I/O bandwidth among disparate users. However, users and operators are very poor at predicting future requirements and, as a result, tend to over-provision grossly.

For three years, we collected detailed usage information for data stored in distributed filesystems in a large private cloud spanning dozens of clusters on multiple continents. Specifically, we measured the disk space usage, I/O rate, and age of stored data for thousands of different engineering users and teams. We find that although the individual time series often have non-stable usage trends, regional aggregations, user classification, and ensemble forecasting methods can be combined to provide a more accurate prediction of future use for the majority of users.

We applied this methodology for the storage users in one geographic region and back-tested these techniques over the past three years to compare our forecasts against actual usage. We find that by classifying a small subset of users with unforecastable trend changes due to known product launches, we can generate three-month out forecasts with mean absolute errors of less than 12%. This compares favorably to the amount of allocated but unused quota that is generally wasted with manual operator-set quotas.

Categories and Subject Descriptors

C.4 [Performance of Systems]: Modeling techniques; K.6 [Management of Computing and Information Sys-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ScienceCloud'12, June 18, 2012, Delft, The Netherlands.

Copyright 2012 ACM 978-1-4503-1340-7/12/06 ...\$10.00.

tems]: Installation Management—*Computing equipment management, performance and usage measurement*

General Terms

Management, Measurement

Keywords

capacity planning, resource usage, ensemble forecasting

1. INTRODUCTION

Storage providers in a cloud environment may serve thousands of users and groups of users with widely varying requirements generally using a distributed storage system such as GFS [8] and Bigtable [5]. Each user's requirements are a complex combination of capacity, I/O bandwidth, seeks, and caching capacity for hot data, and the storage provider must plan in advance so that the users receive adequate levels of capacity and isolation. The time horizon for making such provisioning decisions can range from weeks, when the capacity can be provided by rearranging existing loads, to months if new hardware must be ordered and integrated, to years if entirely new datacenters are to be set up.

Conventionally, provisioning is done based on per-user quotas with the aggregated demand of these quotas used to inform purchasing decisions for additional capacity. While alternative mechanisms for provisioning and allocating storage quota have been proposed [14], they are not in widespread use. However, users and operators are very poor at predicting future requirements. Generally, operators collect estimates from at least the large users, and aggregate them. For safety, both the users and the operators may add headroom over these estimates, leading to gross over-provisioning.

To see if the storage requirements could be predicted automatically, we studied usage information collected over three years on data stored in distributed filesystems in a large private cloud spanning dozens of clusters on multiple continents. We recorded storage quota allocated and actual usage of storage, as well as I/O rates and the age distribution of data stored by thousands of users and engineering groups. Analysis of the trends in the data shows that the usage of individual users and groups is hard to predict, since

they often have time-varying trends. The aggregate usage is easier to predict. We applied a combination of user classification, aggregation, and ensemble forecasting methods, and found that the resulting predictions were considerably more accurate than the operator-assigned quotas.

The rest of the paper is organized as follows. Section 2 describes related work on cloud capacity planning. Section 3 characterizes the storage usage patterns across a large private cloud. Specifically, storage usage, I/O usage, and the age and hotness of data are quantified. Section 4 describes techniques to aggregate users across clusters and classify those with abrupt trend changes. Section 5 describes our ensemble forecasting methodology applied to the aggregations created in the previous section. Section 6 applies these aggregation and forecasting methods to a collection of clusters in a single geographic region across three storage usage dimensions. Section 7 summarizes our results and considers the meanings of the findings.

2. RELATED WORK

Agrawal, et al. [1] collected snapshots of filesystem metadata from a large fleet of corporate desktops and studied the temporal changes in file size, file age, and other characteristics. The characteristics of the cloud storage environment studied here are significantly different from the corporate desktops studied in that work. Allspaw [2] characterizes some of the practical operational challenges of a cloud environment, but does not attempt more advanced methodologies beyond regressions of an observed trend to generate more accurate predictions. There has been a lot of work on resource provisioning, right-sizing, and performance prediction of individual MapReduce jobs [15], but techniques for dynamic scheduling are orthogonal to longer-term capacity planning. Menascé and Ngo [10] explore some of the implications of capacity planning from the point of view of the cloud provider, but their work focuses more on provisioning user needs with existing resources as opposed to forecasting future trends. Loboz [9] finds the distribution of storage needs in a large cloud environment to be highly imbalanced and suggests that traditional approaches to forecasting and capacity planning need to be reconsidered. Mishra, et al. [11] describe the importance of splitting up tasks in a cloud computing cluster into a small number of groups that can be forecast separately, but they do not provide any forecasting methodology. Our work is most similar to the work of Loboz and Mishra, however, we are working with a much larger number of clusters and focus on aggregation and forecasting techniques to provide accurate predictions of future storage needs.

3. CHARACTERIZING DISK USAGE IN A LARGE PRIVATE CLOUD

We studied the storage usage patterns across a large private cloud (we call it *the fleet*) comprised of tens of storage clusters and thousands of users (internal engineers or engineering teams). We define a *storage installation* as a unique tuple of user and cluster. For example, if user A stores data on clusters X , Y , and Z , then A_X , A_Y , and A_Z are three unique storage installations for that user. We group clusters together at multiple aggregation levels for provisioning purposes. We define a *region* as a grouping of two or more clusters in the same geographic area.

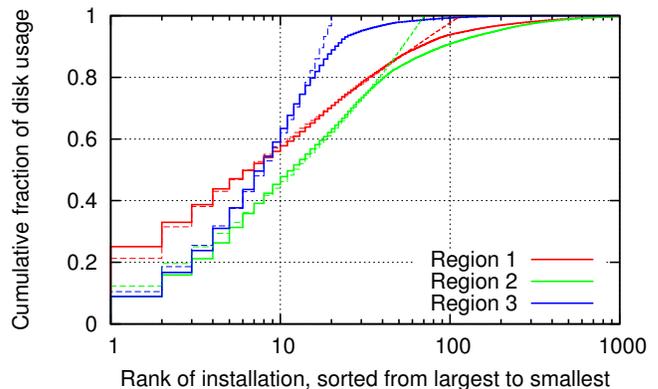


Figure 1: Cumulative distribution functions of the disk space used by storage installations in 3 regions, and Zipf’s law approximations drawn with dotted lines, fitted to the first 80% of disk usage.

The data presented in this section was collected by a fleetwide storage monitoring system. Large numbers of time series are collected from each server in the fleet, but only a few relevant storage usage time series are considered here. These time series were loaded into R [12] for aggregation, cleaning, statistical analysis and final graph generation.

3.1 Storage Capacity

In the fleet, the large majority of the storage belongs to a small fraction of the installations. Figure 1 shows the CDFs of space used by installations in three regions. The CDFs appear to have two components: a smooth portion (on a semi-log scale), followed by a bend. The smooth portion approximately follows Zipf’s law “usage(rank) = $\frac{c}{\text{rank}^s}$ ”, with exponent s between 0.37 and 1.06 depending on the region, and 0.68 fleetwide. The bend means that small users use even less space than what the law would predict given their rank. The bend occurs above 80% of disk usage, so most of the space is used by installations that follow Zipf’s law. Depending on the region, the top 20 to 90 installations account for 90% of the space used, thus the majority of users have little impact on the provisioning process.

In addition to differences in total magnitude of data stored, individual users exhibit significant variation in temporal trends, which creates challenges for forecasting future usage. Figure 2 shows four representative patterns we have observed in the storage usage growth rates. The first row (a) of three time series represents individual users with a linear growth rate. Specifically, these time series fit a linear model with an R^2 value greater than 0.99. The second row (b) of time series represents individual users that fit an exponential growth rate with an R^2 value of greater than 0.99. The third row (c) represents user time series with a significant autocorrelation at the 7-day lag. The fourth row (d) shows some of the regime changes and step functions that can make forecasting particularly challenging.

There are tens of thousands of user storage time series which have similar characteristics to the exemplars shown. For example, many new products initially experience exponential growth. Also, many products or services have a built-in periodicity or day-of-week effect that may cause visible changes in the storage space used on lower level distributed

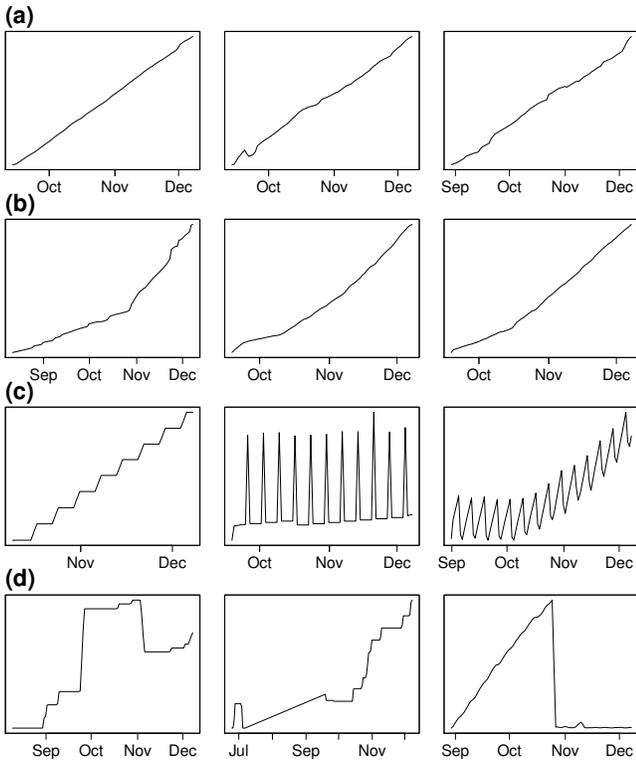


Figure 2: Example storage usage time series from the fleet showing (a) linear growth (b) exponential growth (c) periodicity and (d) regime changes.

filesystems. Other periodic variations may be caused by user behaviors, such as more uploads of YouTube videos or Picasa pictures on Sundays and major holidays. Yet other user storage time series show irregular behavior, as in Figure 2(d), from a combination of usage changes, automated clean-up mechanisms, and manual intervention when individual quota limits are about to be reached.

3.2 I/O Bandwidth

The I/O bandwidth across storage installations follows a similar trend as storage capacity. Figure 3 shows the CDFs of 1-day-average read rates of individual internal users for the same three regions as in Figure 1. Depending on the region, the top 50-90 installations account for 90% of the total read rate.

We cannot assume that I/O rates are simply proportional to the disk space used because for any given installation those two variables are not very correlated. In fact, many teams often read and write data owned by a different account. In the fleet, we observe that 25% of disk space is used by installations that don't read anything. For the remaining installations, Figure 4 shows the relationship between disk space used and 1-day-average read rates. Data points toward the top-left correspond to "hot users" with comparatively large I/O, while points at the bottom-right correspond to "cold users" with comparatively low I/O.

3.3 Age and Hotness of Data

Separately from the total amount of data stored, it is also important to characterize the amount of hot data, since data

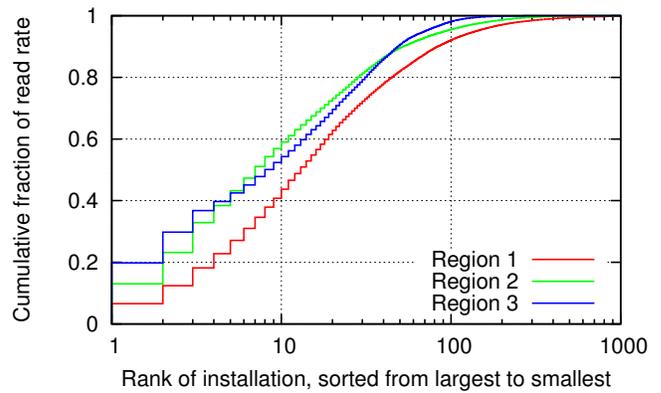


Figure 3: Cumulative distribution function of the average read rate of storage installations in 3 regions.

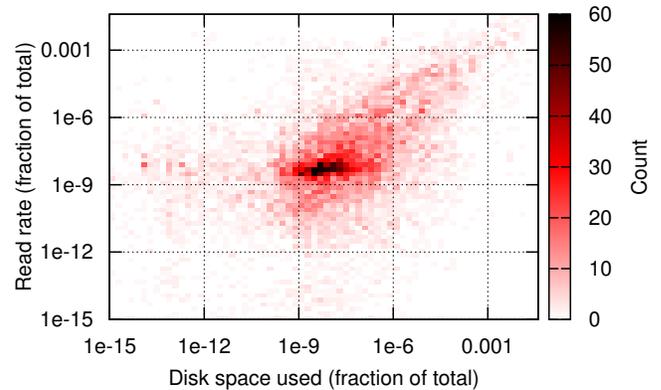


Figure 4: Heatmap of the number of storage installations as a function of disk space used and average read rate. The color intensity of each small rectangle corresponds to the count of installations that fall in that particular range.

that is actively accessed places different demands on the storage system. For example, the amount of hot data may influence the amount of memory required for caching, or the amount of flash in a hybrid flash-disk storage system.

However, hot data is hard to track directly; we instead track the amount of data that was modified recently, or *young* data. The hypothesis is that younger data is accessed more frequently. To test this hypothesis, we look at the fraction of reads directed at the youngest data.

For a representative subset of the storage installations, Figure 5 shows how old the data is, and how file age affects the hotness of the data. It gives read operations by file age, file counts by file age, and bytes stored by file age. As the *read operations* line indicates, in our fleet, a very large fraction of reads are directed to very young data: 30% of reads go to data under 6 hours old, and almost no reads go to data over a week old. The fraction of data that is under 6 hours of age is much smaller than the fraction of the files of that age, which indicates that there is a larger fraction of small files among the young data than overall, but they do not survive beyond a few weeks.

Combining data size and read information tells us that the

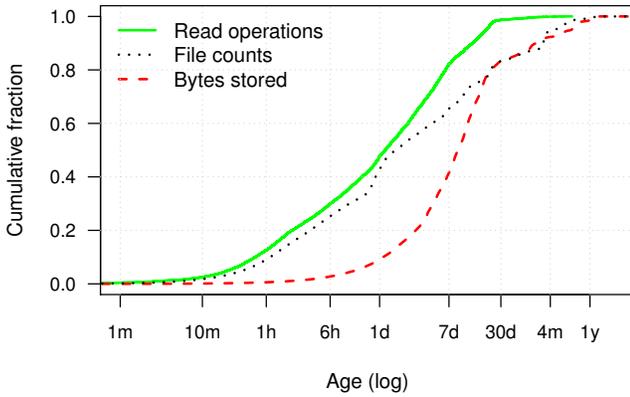


Figure 5: CDF of read operations, file counts, and bytes stored by age.

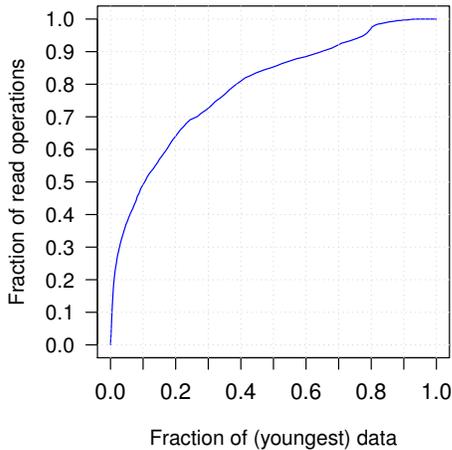


Figure 6: The fraction of read operations for a given fraction of the youngest data.

vast majority of reads are directed at a very small fraction of the data. Figure 6 shows the fraction of reads directed to a given fraction of the youngest data. For example, 50% of all read operations go to the 10% of the data that was most recently written. The newer the data is, the more often it is accessed by read operations. For the purpose of designing a caching system for the hot data, we conclude that it is adequate to estimate the amount of young data — say, data that is less than a week old.

4. MAKING STORAGE REQUIREMENTS FORECASTABLE

The variation observed in the storage time series (Figure 2) described in the previous section is the primary challenge in forecasting storage requirements. Many time series exhibit abrupt trend changes, step functions, periodicity, and significantly different growth rates. The remainder of this section describes aggregation and classification techniques to generate smoother time series for forecasting.

4.1 Aggregation by Region and User

In this section we study two specific aggregations by region and by user, and in the following section we describe a

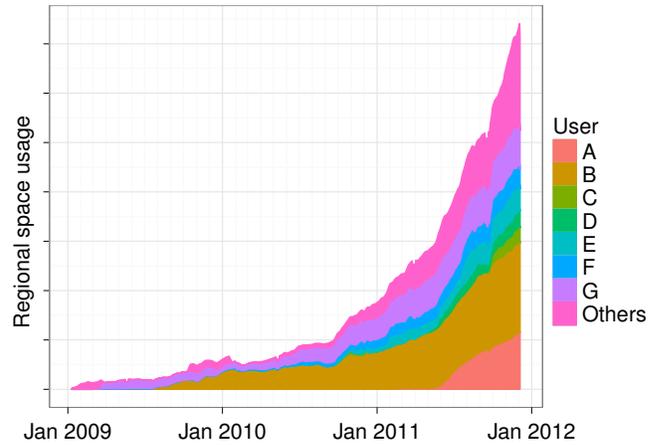


Figure 7: Regional space usage over time showing the impact of a single new large user (“A”) affecting the aggregate trend line.

clustering methodology to produce additional aggregations more amenable to forecasting. Aggregating usage by region is useful for forecasting because users can relatively easily be relocated to a different storage cluster within the same region. Aggregating by user across multiple storage clusters is useful because, in some cases, the user has simply switched to using a different storage cluster (perhaps due to capacity constraints), and hence the per-installation usage trend changes abruptly, but the aggregate usage trend does not.

Figure 7 shows usage in a single geographic region broken down by users across all clusters in that region. The aggregate usage pattern for the region is smoother than the individual patterns, but still has a large trend change in 2011. One of the largest users, *User A*, began storing data in mid-2011 and was responsible for a large part of the trend change of the total storage usage for the region. The regional aggregate becomes smoother with this outlier removed. This example underscores the importance of separating out users with large changes in the usage trend.

4.2 Incorporating User Signals

While we have argued that users are generally poor at predicting their future requirements, input from users can be quite helpful — and sometimes critical — in making provisioning decisions. For example, when a new product launch is planned, the user probably knows that additional storage capacity will be required, and this increase cannot otherwise be predicted by considering only the historical usage.

Quota requests are often used in the cloud to provide a signal to the operators about future resource needs; once granted, quotas also serve to assure users that the resources will indeed be available. In our fleet, installations are currently required to purchase quota if they use or plan to use more than a threshold amount of disk space. Figure 8 compares cumulative quota relative to the cumulative disk space used in those installations. We find that in aggregate, users use only 55% of the quota they purchase. However, large users are better at predicting how much storage they need than smaller users: the top 10 quota requesters use 69% of their quota. Since large users also have a disproportionate impact on aggregate trends, we can use quota requests from large users who are expecting significant changes to

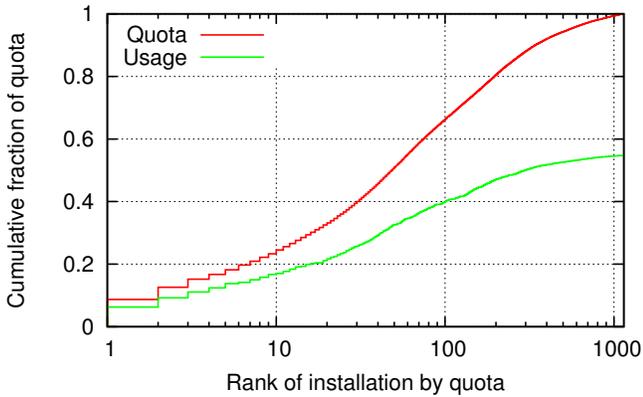


Figure 8: Cumulative distribution functions of quota and disk usage for 1144 installations currently required to get quota.

their storage requirements to improve the accuracy of our forecasts. Therefore, our goal is to automatically identify a small subset of users and require them to use traditional planning tools. These inputs provide an additional signal to our forecasting methods.

There are two general characteristics of storage users who may have an adverse impact on the predictability of the aggregate trend:

1. Users storing a large aggregate amount of data.
2. Users whose usage has been unpredictable in the past.

We want to select only the users for which both of these conditions hold. Fortunately, as we showed in Section 3.1, the first criterion applies to only a small number of the users, and they account for the majority of the usage.

For the second criterion, we propose a procedure where individual forecasts are prepared for the large users, and only users whose past usage has been unpredictable, or who expect their capacity requirements to differ from this prediction by more than a threshold percentage, are required to file a quota request. These users can then be provisioned for manually, or the quota request signals can be incorporated directly into the forecasts using multivariate regression models on the signaling time series. We define a user as having an unpredictable usage pattern if the forecast error (see Section 5.3) observed in the past exceeds 60%. Back-testing over the largest 70 users in the fleet, we found that 12% of the users were unpredictable for 1 month predictions and 22% for 6 month predictions. We use the high 60% unpredictability threshold to minimize the number of users who need to file quota requests, and also because we found that including users in the regional aggregate generally improves its predictability unless the user has an extremely unpredictable usage pattern. In the example shown in Figure 7, using the 60% threshold only marked one user as unpredictable.

One weakness of this proposal is that it requires constant enforcement to ensure the largest users are filing accurate capacity plans in instances where they start to exceed their forecast usage. To combat this, we also seek to make our forecasting methodology as responsive as possible to abrupt trend changes, as described in the next section.

5. FORECASTING METHODS

The second challenge in forecasting these storage requirements lies in the difficulty in finding a single statistical model to capture all of the different types of growth in a large cloud environment. Even after the aggregations described in the previous section, we may still have many hundreds of time series across different clusters or geographic regions that will need to be forecast accurately with minimal human intervention. Traditional statistical forecasting methods would need to be tuned for the growth patterns in each case. For example, some time series exhibit linear growth, others exponential, and others may have a seasonal component. We address this challenge with an ensemble forecasting methodology that enables us to build a robust model across a large number of time series with minimal manual parameter tuning.

In the next three subsections we provide a high-level overview of our forecasting methodology.

5.1 Ensemble Methods

Instead of fine-tuning a single model which suits a particular application, we generate forecasts by averaging an ensemble of forecasts from different models ([13], [3], [4], [6]). Averaging out the various errors from individual models yields variance reduction and robustness. While this methodology might not provide the best forecast for every single user, it consistently produces adequate forecasts for hundreds of aggregated disk usage time series, where human intervention would be impractical. The ensemble combines several different statistical models, including linear and exponential regressions, autoregressive integrated moving average (ARIMA) models, and Bayesian structural time series models [7]. The individual forecasts from these methods can be weighted in different ways according to accuracy and aggregated to form the final forecast. We use a two-step aggregation method designed to be robust to outliers and poorly tuned models in the ensemble. In the first step, we exclude forecasts that are three or more standard deviations away from the ensemble mean. In the second step, we average the remaining forecasts that have z-scores between -0.85 and 0.85 at each time point (corresponding to the 20th and 80th percentiles of the Standard Normal Distribution). This ensures that when given a set of forecasts that are skewed in one direction, we exclude only those in the tail as opposed to equally removing forecasts from both directions as would be the case with a simple trimmed mean.

5.2 Tuning the Training Period

One of the key observations in our storage usage data is that these time series often exhibit significant trend changes. This is visible in the individual user time series of Figure 2(d) as well as the regional aggregate time series of Figure 7. These trend changes adversely affect forecast accuracy, especially when we try to forecast six months or more into the future to allow provisioning of new servers or full datacenters. Our simple solution to this problem is to try to avoid using the trend change points when possible. For each forecast date, we use a 1 month *validation* period and a variable *training period*, as shown in Figure 9. In order to find the best training period, we start with a 1-month long training period and slide the training set begin date backward (in increments of 1 month) until a 6-month long training period is reached. Starting with the shortest training period, we

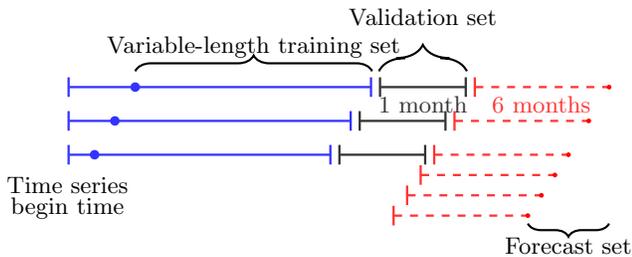


Figure 9: Tuning the forecast training period and forecast back-testing.

t (%)	MAPE				
	R_1	R_2	R_3	R_4	Average
0	11.8	13.9	22.1	33.2	28.0
10	11.2	13.9	22.0	31.9	27.3
20	11.3	13.7	21.9	33.0	27.2
50	12.4	14.3	22.2	34.1	25.8
100	13.3	14.4	22.2	34.8	26.7
∞	21.5	16.6	25.8	38.1	31.1

Table 1: Sensitivity analysis for the tuning threshold parameter.

calculate the mean absolute percentage error (MAPE) over the fixed validation set and repeat this process for the incrementally increasing training windows. Sliding the training set begin date backward increases the training set length and if there are no trend changes in the data, this will usually result in lower forecasting errors. In this scenario, if despite increasing the training set length, the forecasting error increases by more than $t = 10\%$, we infer that we have started including training data from a period before a trend change. Therefore, we decide to cut the training set at the previous shorter training window to only train on data after the trend change. The selected training period is then used to forecast the unknown future requirements. We find that this tuning method limits the impact of trend changes on the accuracy of the forecast.

Sensitivity analysis

To determine the sensitivity of this approach to the parameter t , we examine the above method with different threshold values and measure MAPE for several representative time series using the back-testing method described in the next section. Table 1 shows the results of five different possible values for t along with the case when no tuning is performed and a fixed 6-month long training period is considered ($t = \infty$). The analysis is performed for several regions (four of which are shown). We also report the average MAPE values (taken over all the regions) for different values of the threshold parameter. We observe that tuning the training period results in substantially smaller forecasting error. This is observed consistently over all the regions. Averaging over all the regions, a threshold parameter in the range of $10\% \leq t \leq 50\%$ results in lowest values for MAPE. Furthermore, the algorithm is robust to small changes in the threshold value. The results in section 6.1 are obtained by applying the above methodology to region R_1 using $t = 10\%$.

5.3 Evaluation and Risk Analysis

To evaluate our forecasting methodology, we back-tested each week over the last two years to calculate what our best projection would have been at that point in time and compared that to the actual usage 3 months later. To do this, we perform 1-month and 3-month out forecasts for each data point in the time series. We start from the end of the time series and go backward until there is not enough data left to perform the above methodology. We then calculate the MAPE over each forecast set and use it as a measure of forecast accuracy.

In practice, an extra buffer is added to the forecast value to decrease the risk of capacity shortage. To quantify this risk, we calculate the probability of capacity shortage for different values of added buffer based on historical trends. This probability is computed by measuring the number of times the sum of the forecast and the added buffer is below the actual usage value. In particular, we measure the minimum buffer needed such that the risk of capacity shortage drops to less than 1% for a 1-month horizon, below 5% for a 3-month horizon, and below 25% for a 6-month horizon. In this context, a better forecast results in a lower extra buffer required to limit the capacity shortage risk, and hence a more efficient resource allocation scheme.

6. CASE STUDY

In this section, we use the methodology just described to forecast the three dimensions of disk usage characterized in Section 3. Specifically, we forecast disk usage, I/O usage, and the growth of hot / recent data for the datacenters in a single geographic region. We evaluate the methodology by back-testing for each week over the period for which we have data. We calculate 1-month, 3-month and 6-month forecasts, and compute the MAPE with actual usage.

6.1 Forecasting Regional Space Usage

We forecast space usage across multiple time horizons. Short term forecasts can be useful for operational decisions about data migration, and longer term forecasts can be useful for influencing purchasing decisions for new storage hardware. As we showed in earlier sections, many individual user time series are not forecastable, but the aggregation of all users across clusters in a given region follows a much more stable trend. Figure 10 shows how our aggregation, classification, and forecasting methodology worked by back-testing this procedure over the previous three years and comparing against the actual observations. We also include a 6-month out linear regression forecast for comparison purposes.

Table 2 breaks out the MAPE we measured. Although forecasts of 6 months or longer in duration still have a high degree of uncertainty, forecast errors are nevertheless lower than the amount of allocated but unused quota. Furthermore, using the classification methodology to identify large users with unstable trends and requiring them to file manual capacity plans improved the region-level forecast in 1-month and 3-month horizons. In this case, only one user needed to file a plan.

Figure 11 presents the risk analysis for the same region for 1-month, 3-month, and 6-month provisioning using our forecasting methodology. As we expect, for shorter horizons the forecasts are more accurate and therefore, less extra buffer is required to ensure a low (e.g., 1%) risk of capacity short-

Data	MAPE	
	3m forecast	1m forecast
Regional total	13.2	6.5
Region - User "A"	12.3	4.4

Table 2: 3-month and 1-month out forecast errors for Region and Region excluding User "A".

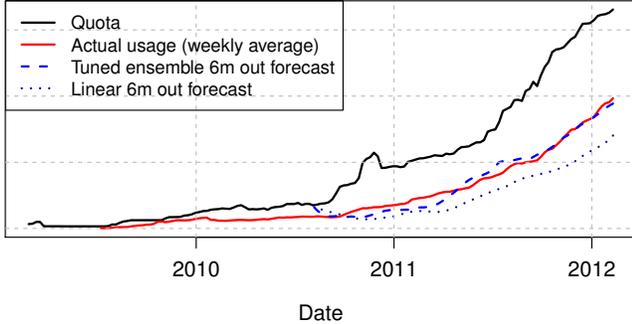


Figure 10: Regional forecast (region R_1): our proposed forecasting method is more accurate than the conventional linear forecast and results in a more efficient capacity planning compared to quota-based planning.

age. As the forecasting horizon increases, the amount of such buffer increases as well. For our target of 25% risk for a 6-month horizon and 5% for a 3-month horizon, we needed a buffer of about 33% over the forecast.

6.2 Forecasting Regional I/O Usage

To test the effectiveness of our forecasting methods, we also evaluated the technique against the I/O usage time series characterized in Section 3.2. Like available disk space, I/O bandwidth is an important dimension of disk capacity. Accurate forecasts of future I/O requirements can be used to migrate workloads optimally to take advantage of the predicted future capacity.

Figure 12 shows the historical mean daily aggregate I/O rate observed in a cluster across the last four months. The first plot shows three peaks, and the second plot shows the single user responsible for those peaks as found by the filtering methodology described earlier. The final plot shows the result of removing this single unpredictable user from the aggregate; the series is clearly smoother. We back-tested our forecasting methodology with and without this user in the aggregate, generating 7-day forecasts using the prior 14 days of history. Excluding the user with the unstable usage trend improved the MAPE of our forecasts from 15.2% to 13.4%.

6.3 Forecasting Recent Data

Finally, we test this methodology on the time series of data age that was introduced in Section 3.3. Figure 13 shows the time series of the total amount of data stored in a representative cluster over the last year and the time series of the amount of data that is less than seven days old over the same time period. Each line is accompanied by the best forecast value for that point in time given the available data 30 days prior. The forecast MAPEs are 13% and 14% for the last year.

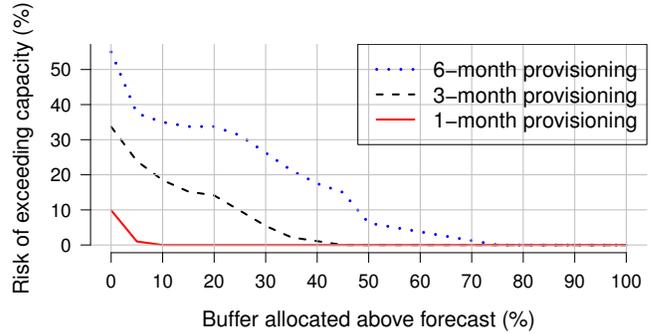


Figure 11: Risk/Buffer analysis for 1-month, 3-month and 6-month provisioning.

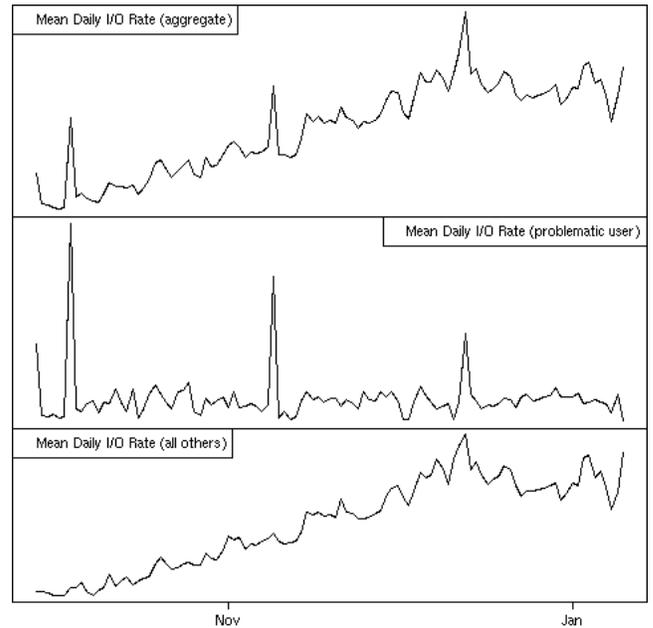


Figure 12: (1) the regional I/O usage, (2) a single user responsible for large periodic spikes in usage, and (3) the smoother trend of all remaining users.

7. CONCLUSION

We proposed an alternative methodology for provisioning storage resources in a large cloud-computing environment. This methodology involves aggregating small users together, classifying users based on their trend changes, and using ensemble forecasting methods to provide accurate predictions of future use for the majority of users.

Our preliminary experiments show that our proposed method results in substantially better predictions of future usage than relying on manual operator-set quotas. As with all forecasting methodologies, accuracy is particularly sensitive to the length of training data and the forecast time horizon. For long-term predictions to be safe (i.e., have a low chance of under-provisioning), we need to add a substantial buffer over the predictions, but the resulting provisioning is still considerably leaner than using quotas, and the proposed method requires very little manual effort.

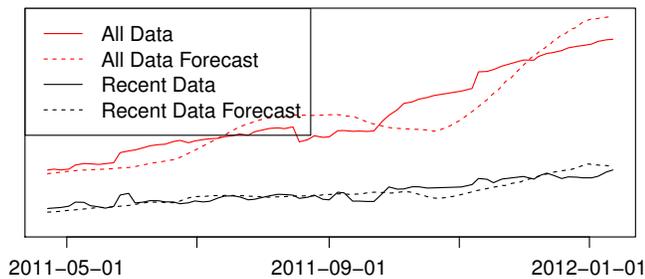


Figure 13: Time series of data stored and recent (< 7 days) data stored along with the back-tested 30-day forecast values for a representative cluster.

8. ACKNOWLEDGMENTS

We would like to thank Jordyn Buchanan and Rob Ewaschuk for motivating much of this work based on their years of experience managing distributed storage operations at Google. We are greatly indebted to our colleagues from Google’s forecasting team for assistance and code. Eric Tassone, Farzan Rohani, Nate Coehlo, and Steve Scott were especially generous with their time. Most of our experimental results were gathered through the use of an R package written primarily by Angus Lees.

9. REFERENCES

- [1] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch. A five-year study of file-system metadata. *ACM Trans. Storage*, 3, October 2007.
- [2] J. Allspaw. *The Art of Capacity Planning: Scaling Web Resources*. O’Reilly Media, Inc., 2008.
- [3] J. S. Armstrong. *Combining Forecasts, Principles of forecasting: A handbook for researchers and practitioners*.
- [4] J. S. Armstrong. Combining forecasts: The end of the beginning or the beginning of the end? *International Journal of Forecasting*, 5:585–588.
- [5] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: a distributed storage system for structured data. In *OSDI ’06: Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, pages 205–218, Nov. 2006.
- [6] R. Clemen. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5:559–583, 1989.
- [7] J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Number 9780198523543 in OUP Catalogue. Oxford University Press, 2001.
- [8] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google file system. In *SOSP ’03: Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 29–43, Oct. 2003.
- [9] C. Z. Lobo. Cloud resource usage: extreme distributions invalidating traditional capacity planning models. In *Proceedings of the 2nd international workshop on Scientific cloud computing*, ScienceCloud ’11, pages 7–14, New York, NY, USA, 2011. ACM.
- [10] D. A. Menascé and P. Ngo. Understanding cloud computing: Experimentation and capacity planning.
- [11] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das. Towards characterizing cloud backend workloads: insights from Google compute clusters. *SIGMETRICS Performance Evaluation Review*, 37(4):34–41, 2010.
- [12] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.
- [13] M. Stokely, F. Rohani, and E. Tassone. Large-scale parallel statistical forecasting computations in R. In *JSM Proceedings, Section on Physical and Engineering Sciences*, Alexandria, VA, 2011. American Statistical Association.
- [14] M. Stokely, J. Winget, E. Keyes, C. Grimes, and B. Yolken. Using a market economy to provision compute resources across planet-wide clusters. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE, 2009.
- [15] A. Verma, L. Cherkasova, and R. H. Campbell. ARIA: automatic resource inference and allocation for MapReduce environments. In *Proceedings of the 8th ACM international conference on Autonomic computing, ICAC ’11*, pages 235–244, New York, NY, USA, 2011. ACM.