Cloud Storage I/O Workload Characterization: A Time-Series Analytic Approach

Abiola Adegboyega
abiola.adegboyega@gmail.com

***Abstract*—The cloud continues to host diverse applications based on existing and emergent areas all with different workload characteristics. Given this, workload traces released by public cloud operators provide opportunities for analysis to gain further insight into their environments. This can inform capacity planning, autoscaling and forecasting among other functions integral to efficient and profitable cloud operations. This paper presents the statistical analysis & modeling of a recent storage workload trace from the Alibaba public cloud. It characterizes the trace by the isolation and aggregation of all read and write time-series per recorded workload. Application of statistical methods has yielded novel distributional fitting for aggregated read and write workloads. Furthermore, performance models integrating second order statistical measures of time-varying variance were employed to capture workload burstiness. This was done by methods employed in econometric literature. Finally, the set of workload time-series has been made available online for further analysis by the research community.**

Keywords—cloud; storage; workloads; time-series

I. INTRODUCTION

The increasing diversity of applications and service models existent in cloud-computing environments provides both opportunities and challenges for operators. Organizations obviate private datacenter maintenance & infrastructural costs by migration of their applications to the cloud. Increasingly, applications are developed and deployed natively in the public cloud with legacy ones refactored to meet current demands. There also remain options for hybrid deployments where applications are integrated across private and public cloud environments. Furthermore, service delivery models [1] have expanded beyond their initial cloud definitions each with unique use-cases and requirements across different platforms and cloud operator environments. Understanding the dynamics of public cloud workloads is necessary to enable operators manage computing resources at scale while meeting service guarantees for their customers. The major operators of public cloud-computing infrastructure continue to release traces from their environments to the research community to foster better understanding of such workload dynamics [2],[3],[4].

This paper carries out a characterization study of a storage workload trace from the Alibaba cloud. The trace is composed of I/O traffic from 1000 block storage volumes spanning 31 days of observation. While not the first study of the trace, the uniqueness of approach realizes each recorded workload as a time-series different from prior work. In addition, this study employs statistical measures of time-varying variance to capture workload burstiness. This initial analysis focuses on statistical measures applicable to the elicited time-series for each recorded workload in the publicly available trace. The methods applied have yielded insights given workload run-lengths, load intensity, observations of volatility and a traffic mix for which modeling methods can be determined applicable in areas such as scheduling, capacity assignment and forecasting. The contributions of this paper are:

- The isolation, aggregation & statistical analysis of all read and write workload time-series in the studied trace.
- Statistical time-series fitting with an ensemble of the *t*-distribution family & the Lognormal model.
- Workload burstiness performance modeling with Generalized Score methods employed in econometrics.
- A performance comparison of realized models with state-of-the-art demonstrating where there are improvements.

The rest of the paper is organized as follows. Related work in workload trace analysis is discussed in section II. Section III provides time-series realization & initial global observations & that of selected subsets. Section IV presents statistical analysis of aggregated time-series. Burstiness modeling is given in Section V. Section VI provides performance evaluation with conclusions provided in section VII.

II. RELATED WORK

Analysis of workload traces remains an active area of research. The major cloud providers continue to release workload traces from their environment, and these have been the subject of recent study. A diversity of workloads from Google, Facebook and Wikipedia were utilized in the evaluation of predictive methods in [5]. A cloud storage workload characterization study by Ren et al [6] analyzes trace characteristics also from the Alibaba cloud including latency, locality and load intensity in their study. The challenges of analyzing workload traces both in terms of high dimensionality and volatility were observed by Chen et al [7]. Jiang et al [8] analyze another set of traces from the Alibaba cloud focusing on CPU and disk I/O characteristics. They isolated statistical distributions for some workloads and identified environment bottlenecks. In addition to these, several studies have been conducted on earlier released traces from the provider in [9],[10],[11].

Specific to the trace studied in this paper, a similar study was done in [12]. While their work included analysis like the time-series methods here, they focused on predictive models employed for capacity planning. The work by Wang et al [13] employs the same trace in a study focused on I/O optimization and orthogonal to the work here. Zhou, Wang, Shi and Feng [14] use analysis from the same trace to analyze block cache usage in the design of optimization methodologies. The

previous work where characterization is concerned employed summary statistical observations about load intensity and usage patterns. They have also provided necessary details regarding spatial and temporal patterns necessary for resource provisioning. The work by Zou et al [15] also studies the read & write workloads from the same trace employing statistical methods similarly done here. This work, however, differentiates itself given the modeling of workload burstiness with time-varying volatility models as well as the realization of novel statistical models with a methodology different from the prior. However, there remain important characteristics that can provide unique insights by tracking workload dynamics over the entire trace observation period. This method is here applied differently from current work and is reported subsequently.

III. TIME-SERIES REALIZATION & SUMMARY STATISTICS

A. Trace Details

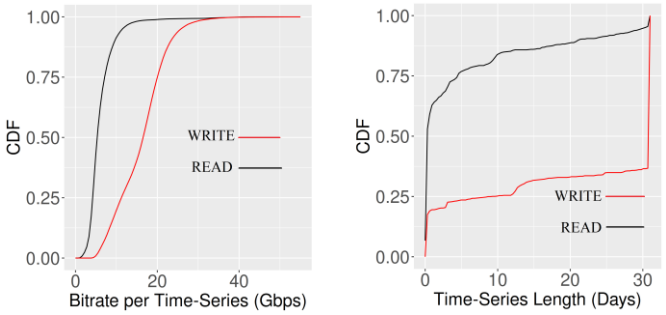
The workload trace was released in 2020 and first studied in the work by Li J et al [16]. The trace is available for download online (https://github.com/alibaba/block-traces). The cloud block storage system studied serves as middleware that integrates virtual disks described as volumes and the physical storage clusters of the cloud provider. Applications typical of the cloud are assigned dedicated volumes and issue read or write requests via the volume to storage clusters. The authors indicate the absence of response time details for recorded I/O requests without which latency analysis can be done. Furthermore, the application types are not provided while the traces also do not provide any usable metrics regarding the physical storage volumes. The trace observation period was from December 31st, 2019, to January 30th, 2020. The trace data was 20.2 billion rows. Each volume in the trace is identified by a tuple of {device id, opcode, offset, length, timestamp} The device id field is a numeric identifier from 1 to 1000. The opcode is either R/W for Read/Write operations respectively. Offset and length (of operations) are in bytes.

B. Time-Series Realization

Time-series realization employed a methodology whereby each workload (device ID 1 to 1000) was filtered on a tuple of {Day, Hour, Minute, Bandwidth (Mbps, Gbps etc.)}. The original Unix timestamp in microseconds was converted and formatted as (Day, Hour, Minute). This was done to facilitate observation of workload dynamics at different timescales for inference of possible weekly, daily, or hourly patterns. To further reduce the noted high dimensionality, the workload throughput was summed over each minute to facilitate easy viewing. This method produced time-series with a maximum length of 44,640 variables of Read and Write throughput for each volume over the 30-day observation period. Furthermore, 3 volumes {811, 819, 822} recorded no write output in the trace while a set of 67 volumes recorded no read data. We proceed with the analysis of the volumes for which recorded read/write data were observed in the trace.

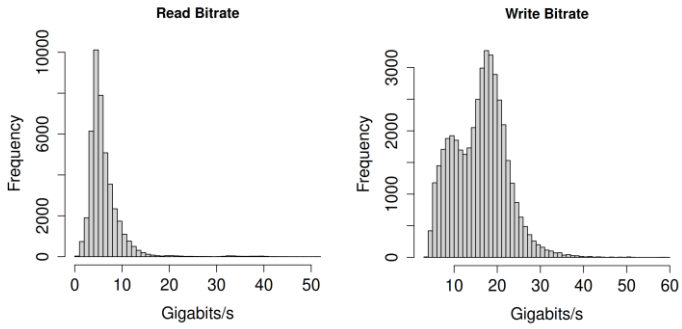
Table 1: Summary R/W time-series statistics for all volumes

Type	Volumes	μ (Gbps)	SD	Total (TB)
Read	933	6	3.72	267
Write	997	16	6.11	695



(a) (b)

Figures 1a & 1b: CDF for bitrate & run-length for all extracted read/write time-series.



(a) (b)

Figures 2a & 2b: Histogram of Read and Write Throughputs across all extracted read/write time-series.

C. Summary global workload analysis

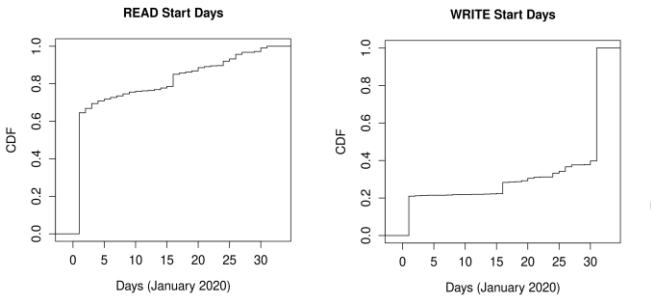
Upon the realization of all read/write time-series for all volumes, observations were made of their summary statistics which are provided in table 1 (below left). Furthermore, the realized time-series were validated against current work in their analysis to ensure accuracy. To this end, the Cumulative Distribution Function (CDF) for the time-series workload throughput is illustrated in Figure 1a. Figure 1b provides the CDF of the length in days for which the volumes actively read and wrote data over the observation period. Figure 1b is validated by the same time occupancy CDF graph reported in the original work by Li J et al [16]. Further to this, as reported in the prior work, Figure 1b illustrates that the workload is write-dominated in which case, 60% of all volumes had full occupancy for writes (30 days or more) compared to 20% of all volumes for reads over the same period. Figure 1a and 1b illustrate the bitrates for read and write workloads. While prior work reported on request size in bytes, the bitrates observed here show that 75% of overall average workload throughput is about 10 Gbps for writes and even smaller at 5 Gbps for reads.

Beyond observations of occupancy and returned bitrates, the knee of the CDFs for both reads and writes (Figure 1a) suggest global skewness measurable as outliers or extreme values. Specific to the study at hand, these would define burstiness captured in periods of intense reads/writes. This has been reported in current research for storage workloads [17],[18]. To investigate this further at the global time-series level, the read and write histograms for all recorded volumes are given in Figures 2a and 2b respectively. The read histogram provides visual evidence of a right-tailed distribution. This is evidenced by bitrate values from 20 up to 40 Gbps. The write histogram suggests a multimodal distribution while also being right-tailed as illustrated. While no distributional assumptions are yet made for the sake of brevity, further statistical exploration can isolate performance models able to track the workload dynamics given the global properties here observed. We proceed with a visual analysis of a subset of isolated workloads that supports the global properties observed and their subsequent aggregation and volatility modeling.

D. Diversity motivating aggregation & volatility modeling

Further to the global analysis completed, the isolation of individual time-series has enabled observation of some usage properties unreported in similar studies on the same trace. In addition to the CDF of the time-series *lengths*, Figure 3a and 3b provide the *start-time* CDFs for all measured read and write time-series respectively. This illustrates the start-times for the month of January 2020. As shown, at least 80% of writes begin shortly after January 1 compared to 40% for reads. The CDF evolution supports the earlier observations for the dominance of write operations of Figure 1b. Further observations supporting start-times and their contributions to diversity are exemplified in the read and write time-series plots for volume 248. The write time-series is illustrated in Figure 4 while Figure 5 illustrates the read throughput. It was observed that the write operations for this volume returned data equally over the observation period (30 days) with 44,640 values recorded. The read time-series for the same volume recorded data spanning the observation period of 30 days but with 4608 values as illustrated in Figure 5. Finally, Figure 6 illustrates a subset of 15 random workloads that exhibit one-time burstiness patterns which is here described as an "impulse" workload. These sustain a high write throughput but only for a few seconds at the start of the operation as illustrated upon which they revert to low throughput. While this pattern suggests the write-once workloads specific to some cloud databases, more investigation into their read operations is needed for confirmation.

The foregoing observations of workload diversity given time-series lengths, start-times, bitrates and visual usage patterns would require 3 possible statistical methods for analysis. In the first method, each time-series is subject to statistical analysis to identify their underlying model. A second method is the classification of all time-series by clustering methods to track workload dynamics. A third method is one by which all of them are aggregated prior to further analysis. This paper pursues the third method. The motivation for this considers first the nonuniform time-series across 1000 volumes. Hierarchical



(a) (b)

Figures 3a & 3b: CDF of start-times for all read/write volumes.

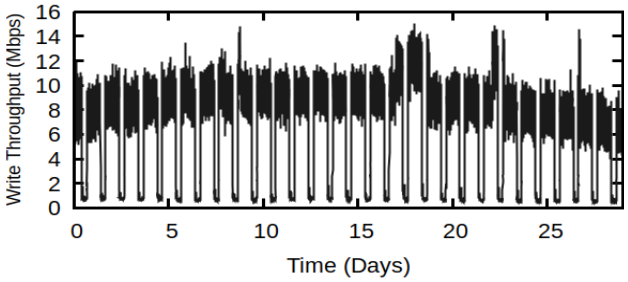


Figure 4: Vol 248 Write throughput time-series for 1 month.

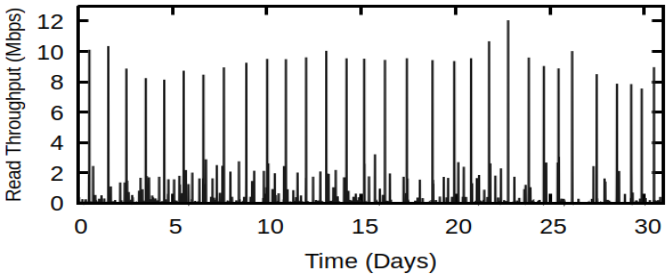


Figure 5: Vol 248 Read throughput time-series for 1 month.

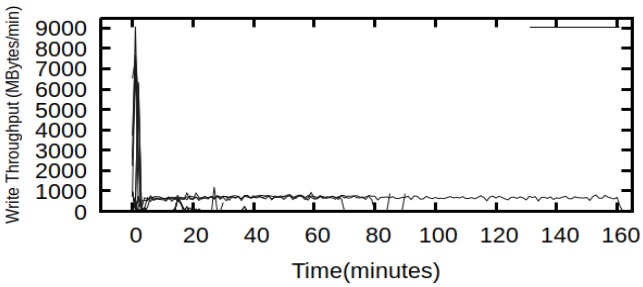


Figure 6: "Impulse" workloads for 15 random volumes

clustering methods exist for nonuniform multivariate time-series [19],[20] but that is its own study. In addition, the statistical modeling in this paper targets cloud resource provisioning methods for which data is aggregated across cloud infrastructure components [21],[22].

IV. STATISTICAL ANALYSIS OF AGGREGATED TIME-SERIES

A. Analysis Methodology

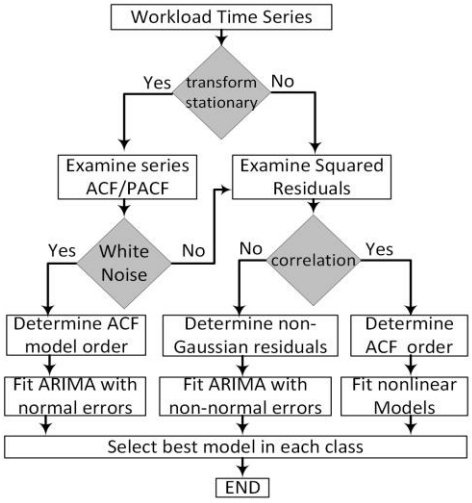


Figure 7: Analysis methodology

Given the foregoing observations, the read and write time-series were aggregated across all volumes. For completeness, both the aggregated (summation over the observation period) and averaged time-series were analyzed. For the read time-series, this study includes analysis detailing findings for both average and aggregate read data. For the write data, only the aggregate time-series analysis is included. This, given that there was no statistical significance between average and aggregate time-series. The methodology employed for analysis is illustrated above in Figure 7. This is based on our earlier work done in [23]. It employs standard methods of model fitting and forecasting [24]. Here linear time-series are modeled as a signal with additive errors. This follows standard methods that explore linear modeling of time-series with determination of statistical fit before exploration of nonlinear models. With reference to Figure 7, the method begins with an initial visual analysis of the time-series for observable properties like trends and seasonality. Subsequently, the time-series is transformed if need be to remove possible outliers to arrive at stationarity for the modeling phase to commence. Then, the Auto-Correlation Function (ACF) given in equation 1 below is used to determine possible relationships that exist over different periods in the time-series:

$$ACF(t,u) = \frac{E[(Y_t-\mu_t)(Y_u-\mu_u)]}{\sigma_t\sigma_u} \quad (1)$$

Here, μ and σ are the mean and standard deviation respectively while the lag (t-u). Upon the determination of the model order from the ACF, the series residuals are examined. Standard linear regressive models assume these residuals to be Gaussian [25]. If tests bear out this assumption, linear models are fitted to the time-series. The observation of non-Gaussian errors with stationarity measures leads to the determination of a fit of nonlinear models through residual squaring as illustrated in the diagram. This is the method employed for both the aggregated read and write time-series whose analyses follow.

B. Aggregate Read Workload Analysis

The analysis of the aggregated read workload time-series begins with an examination of its visual characteristics over multiple timescales. For this, a random selection was done over 1 day, 1 week and the entire observation period of 1 month. Figures 8 illustrates the aggregate daily read throughput for January 5. Figure 9 provides a random week of read throughput. The commonality of Figure 8 and 9 is the workload burst sustained for 15 to 20 minutes each day beginning at 10:00 am as illustrated in Figure 8. Figure 10 provides the *average* read throughput over the observation period while Figure 11 provides the aggregate throughput also over the observation

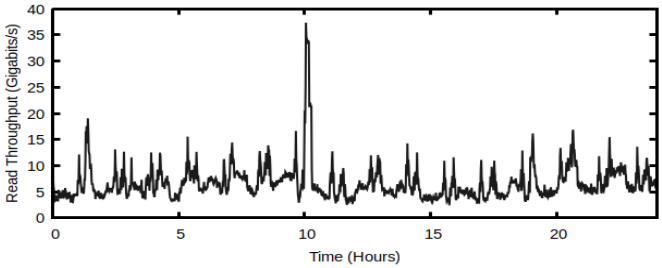


Figure 8: Aggregate read workload time-series (Jan 5).

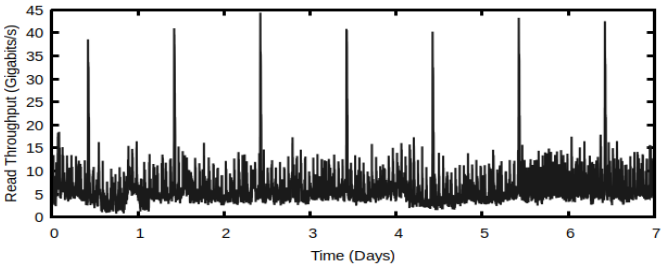


Figure 9: Random aggregate weekly read throughput.

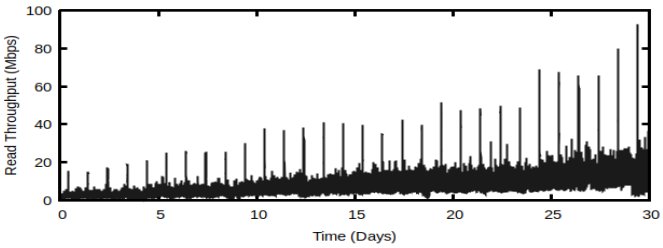


Figure 10: 30-day average read throughput.

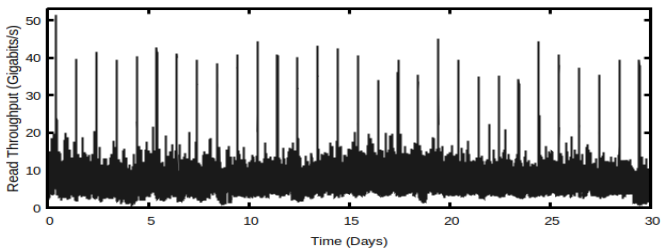


Figure 11: 30-day aggregate read throughput.

period. The observation from Figure 10 shows a steadily increasing average read workload as well as low level variance as illustrated. Analysis now proceeds with model fitting for the aggregate read time-series.

C. Statistical analysis and model isolation for read workload

Burstiness as nonconstant variance

With reference to the methodology of Figure 7, the aggregate read workload time-series was differenced to arrive at stationarity. The model order was determined upon examination of the ACF lags upon which a linear model was selected. Given linear models developed by Box and Jenkins realized as Autoregressive Integrated Moving Average (ARIMA) methods [24] are employed to fit the aggregated read throughput time-series of Figure 11. The autocorrelation function of equation 1 is used to examine the relationships that exist for the observations of the time-series. Figure 12a shows the ACF of aggregate read time-series of Figure 11 after a first order differencing is done to bring it to stationarity. It can be observed that the correlation decays as the lag increases which is necessary to initiate the second stage of model fitting. To ensure accurate fitting, the lag observed in Figure 12a and a model search was conducted with maximum likelihood estimation (MLE) employed in arrival at the final model with selection through the Akaike Information Criterion (AIC) [26]. Going by the linear modeling method ARIMA(*p*,*d*,*q*) where *p* is the lag order (from the ACF of Figure 12a), *d* is the order of differencing and *q* the moving average order, a linear ARIMA(5,1,3) was employed to fit the aggregate time-series of Figure 11. A subsequent examination of the residuals captured in the ACF of Figure 12b shows correlations of lags up to 20. Figure 13a illustrates the squared residuals which support nonconstant variance. Figure 13b is the histogram of the residuals which is skewed right. Final evidence is given in Figure 14b where the residuals are plotted against the original time-series. For constant variance, the points will spread uniformly from the origin in support of normality. As illustrated, we have an expanding pattern from the origin in both directions which is the behavior for increasing variance. For statistical confirmation, the Ljung-Box test [25] to determine serial correlation was done on the residuals with a 95% confidence interval. The test returned a metric that rejected the null hypothesis of the test in support of a linear model.

Tailedness of distributions

Figure 14a shows the tail of the distribution for the residuals in a quantile-quantile plot. As shown, it leans more to the right in the upper quantiles than the lower left-tail. Thus, fitting with a family of right-tailed distributions was conducted for the original aggregate read time-series of Figure 11. Table 2 below shows the goodness of fit statistics conducted both for the normal as well as 3 common right-tailed distributions. The lowest statistic shows the goodness of fit both for the Akaike's

Table 2: Goodness-of-fit statistics for read time-series.

Test	Normal	Lognormal	Gamma	Weibull
AIC	236251	201689	207509	219205
BIC	236269	201706	207527	219223

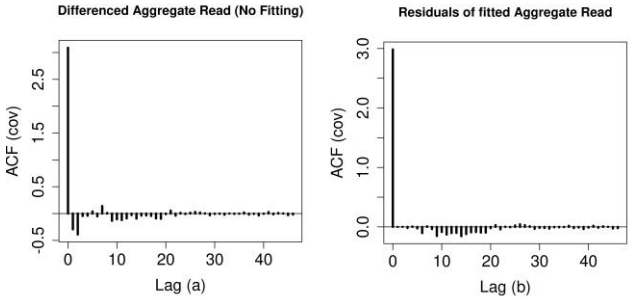


Figure 12a & b: ACFs for aggregate read & residuals.

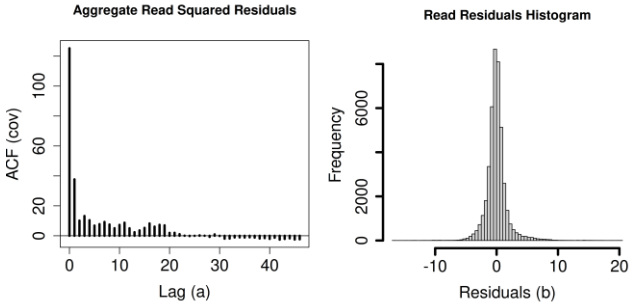


Figure 13a & b: Squared residuals ACF & histogram

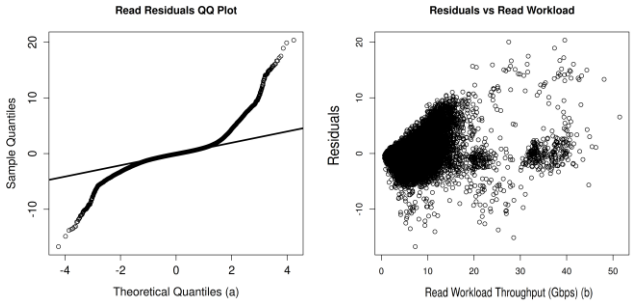


Figure 14a & b: Residuals QQ plot; Residuals vs. Read.

& Bayesian Information Criterion. From table 2, the lognormal distribution shows the initial best fit for the comparisons done.

D. Aggregate Write Workload Analysis

The same methods employed in the analysis of the read workload were applied for the aggregate write time-series across all volumes observed. Figure 15 illustrates the aggregate daily write throughput for January 27, 2020. Observations from a sample of random days not illustrated given space constraints show a dormant period before 4 am each day after which there's a transition to sustained higher workload intensities until 9 pm daily. Figure 16 however shows this pattern to be diurnal with the periodicity sustained not only at the weekly level but for the entire 30-day observation period illustrated in Figure 17. This periodicity is corroborated in the recent characterization study by Zou et al [15].

Applying the same methodology as before, a linear ARIMA (3,1,2) model was determined through an optimized search with residuals examined thereafter. Only the squared residuals and their histogram are illustrated here given that residuals and initial fitted original series exhibit the same properties as illustrated in Figures 12a and b for the aggregate read time-

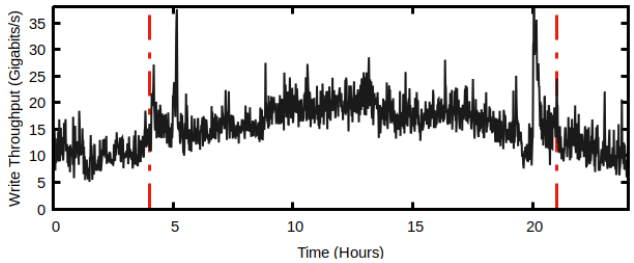


Figure 15: Aggregate read workload time-series (Jan 27)

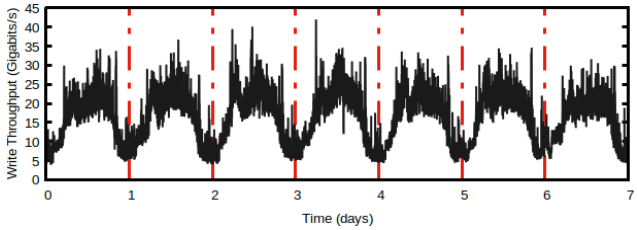


Figure 16: Random aggregate weekly write throughput.

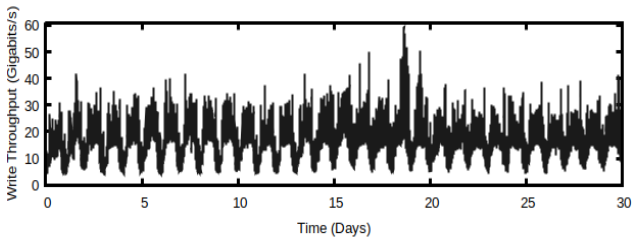


Figure 17: 30-day aggregate write throughput.

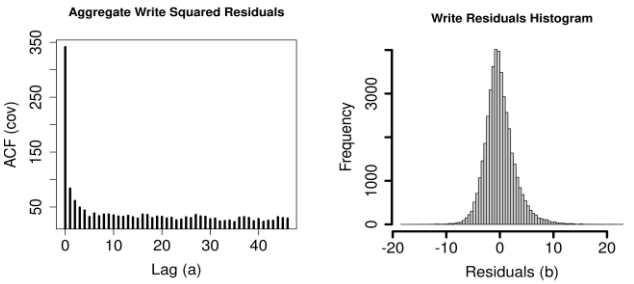


Figure 18a and b: Squared residuals ACF & histogram

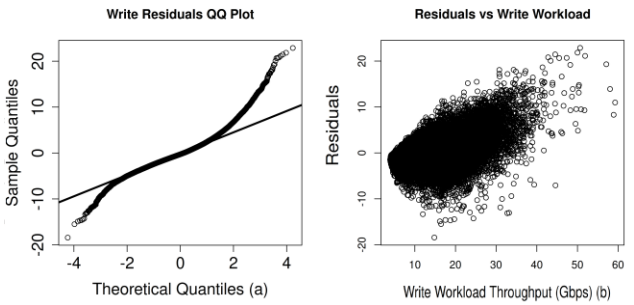


Figure 19a and b: Residual quantiles and Residuals vs. Write Workload

series. Figures 18a and b illustrate the squared residuals and the residual histogram respectively. Figure 19b similarly shows the

Table 3: Goodness-of-fit statistics for Write time-series.

Test	Normal	Lognormal	Gamma	Weibull
AIC	279059	281857	278773	277826
BIC	279076	281875	278791	277843

residuals plotted against the original time-series with the outward spread from the origin as an indication of nonconstant variance. Figure 19a illustrates the quantile plots with the tail leaning right. Finally, table 3 provides the goodness-of-fit for the aggregate time-series of Figure 17 with the Weibull distribution having the lowest initial returned value. It is noted that initial fitting omits the *t*-distribution given parameter estimation is required for bootstrapping through maximum likelihood estimation. The fitting is here employed to illustrate evidence of heavy tailed distribution with a full family of distributions reserved for the model evaluation phase.

V. BURSTINESS MODELING

While select distributions have been isolated given statistical measures that suggest nonlinearity in aggregated read/write time-series under study, discussion here centers on burstiness measures given the earlier diagnostic testing of the residuals of the time-series. We proceed with state-of-the-art models that capture burstiness through volatility and our motivation for their extension.

A linear model y_t is considered, composed of independent variables x_t with additive noise z_t such that $y_t = x_t + z_t$. Regressive polynomials with coefficients a ($a_1, a_2, \ldots$) and b ($b_1, b_2, \ldots$) of order p and q for x and z respectively specifies the ARIMA model:

$$a(B)y_t = b(B)z_t \quad (2)$$

Here, $B = B(d)$, the order of differencing determined for a stationary time-series y_t. Gaussian assumptions in the error term are modeled as zero mean with bounded variance expressed as $W(0, \sigma^2)$.

The Auto-Regressive Conditional Heteroskedastic (ARCH) model of Engle [25] models returns (and in our case, the read/write time-series variables) as:

$$y_t = \sigma_t z_t \quad (3)$$
$$\sigma_t^2 = \beta_0 + \beta_1 y_{t-1}^2 \quad (4)$$

Where z is $W(0, \sigma^2)$. The conditional variance is dependent on the previous value of the series (equation 4). The generalization of the ARCH process is conditional also on past variance with further details found in [25] .

Conditional variance as a measure of volatility in cloud workloads do not account for non-gaussian distributions [15] [23][27]. Given this observation, ARIMA models can be incorporated with statistics of a higher order than the variance to account for non-gaussianity [27]. Generalized Autoregressive Conditional Score (GAS) models developed independently by Creal et al [28] and Harvey [29] consider conditional densities for observations y_t where a time-varying parameter f_t is employed to model the variance. If we consider

such a parameter vector defined by θ_t for the observational variable y_t, we have $y_t \sim p(y_t|f_t, \theta_t)$. The distinguishing feature of this model is the time-varying vector θ_t is arrived at by the score of the distribution of y_t along with the autoregressive component. Furthermore, f_t:$f_t = \gamma f_{t-1} + \alpha s_t$. The parameters γ,α are to be estimated. The score being the log-likelihood derivative given by:

$$s_t = S_t(\theta_t)\nabla_t(y_t, \theta_t) \quad (5)$$

$$\nabla_t(y_t, \theta_t) = \frac{\partial log p(y_t, \theta_t)}{\partial \theta_t} \quad (6)$$

The GAS model was initially specified by the t-distribution. Given its selection in subsequent modeling, the GAS-*t* specification is given in terms of the mean and scale $\varphi = \sigma^2$ as:

$$f(y|\mu, \sigma^2, v) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\sigma^2\sqrt{\pi v}}\left(1 + \frac{(y-\mu)^2}{v(\sigma^2)^2}\right)^{-\frac{(v+1)}{2}} \quad (7)$$

Where Γ is the gamma function. The second derivative of the log-likelihood of equation 7 evaluates to the standard GAS-*t* model which can be found in [28]. With Gaussian assumptions, the derivative of the log-likelihood evaluates to the variance of the score & we have $y_t \sim p(y_t,|\mu,\sigma^2)$. The GAS model is then specified by the (G)ARCH model of Engle in equations 3 & 4. With the models isolated for the aggregated read/write time-series, the GAS model is specified by score per distribution. Given prior model realization for IaaS workloads which is observed in this study for the aggregate read workload, the GAS model specification with log-normal errors according to the log-likelihood is given by:

$$\sigma_t^2 = \gamma\sigma_{t-1}^2 + \alpha(\ln y_{t-1}^2 - \sigma_{t-1}^2) \quad (8)$$

Detailed further analysis can be found in [23][29]. We proceed with the performance evaluation.

VI. Performance Evaluation

Performance evaluation was done with a comparison of forecasting accuracy of the discussed models. While the goodness-of-fit tests were completed to isolate the best models, there exist some degree of ambiguity regarding how even the established methods (AIC/BIC) tackle specificity where model complexity is concerned [30]. Given this, a representative set of realized forecasting models were evaluated against the read and write time-series. Tables 4 and 5 list the top 6 models that returned the best accuracy for forecasts. The rest not tabulated given space are {Skewed Normal, Asymmetric Laplace Distribution, Skellam, Exponential, Beta, Gamma (location & scale)}. From the methodology employed, state-of-the-art models are often represented by a specification of the ARIMA model. However, given focus on volatility modeled with the variance, the GARCH model with gaussian errors was chosen as the current benchmark. This also permitted an optimal comparison with the newer contributions done with the GAS

Table 4: Aggregate Read goodness-of-fit for GAS models.

GAS Model	Short Name	AIC	BIC
Lognormal	LGN	146949	147001
Asymmetric Student T (1 Tail)	AT(1T)	149062	149166
Asymmetric Student T (2 Tail)	AT(2T)	149909	150039
Symmetric Student T	T	155155	155233
Normal (GARCH)	GARCH	163370	163422
Asymmetric Student T	T(AT)	163580	163684

Table 5: Aggregate Write goodness-of-fit for GAS models.

GAS Model	Short Name	AIC	BIC
Asymmetric Student T (1 Tail)	AT(1T)	201419	201549
Asymmetric Student T (2 Tail)	AT(2T)	202353	202457
Asymmetric Student T	T(AT)	203896	204000
Lognormal	LGN	204081	204133
Normal (GARCH)	GARCH	206451	206485
Symmetric Student T	T	206727	206805

model given the latter's reduction to the former with Gaussian assumptions. The R statistical computing environment was employed for the fitting and forecasting exercises. Original code that was realized for the GAS-lognormal model was integrated with the models developed by Ardia et al [31]. The code along with its extensions are made available online at (https://github.com/aadegboyega/GASModel-Extensions). In terms of performance and returned AIC/BIC values however, the ambiguity noted in [30] is corroborated by departures in performance from the results in tables 4 and 5. To validate the model fitting phase, a forecasting exercise was conducted with the models listed in tables 4 and 5. The Mean Absolute Percentage Error (MAPE) was the metric of comparison given below in equation 9:

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \bar{y_t}}{y_t}\right| \quad (9)$$

Here, y_t is the real & $\bar{y_t}$ is the predicted value over the forecast horizon. Forecasting was divided into short and long-term forecasting. Each is discussed in turn.

Short-Term Forecasting

Ten random days were selected from the set of aggregated read and write workloads. The first 23 hours per day were used to train the models with a 1-hour out-of-sample forecast horizon. Five models with the best performance for the aggregate read and write time-series were selected. Figures 20 & 21 illustrate the results. From Figure 20, the GAS-*t* family of models performed only marginally better overall than the GARCH model, the state-of-the-art, 50% of the time. These models performed worse than GARCH for 5 out of 10 random days with measured improvement in accuracy at 14% only 2 out of 10 days. These initial results suggest that the GARCH model's nonconstant variance tracks the dynamics of the read workload effectively for performance prediction purposes. For the aggregate write series, the initial observation is that *all* in the *t*-distribution models performed better than the GARCH model across all ten random days of the performance evaluation.

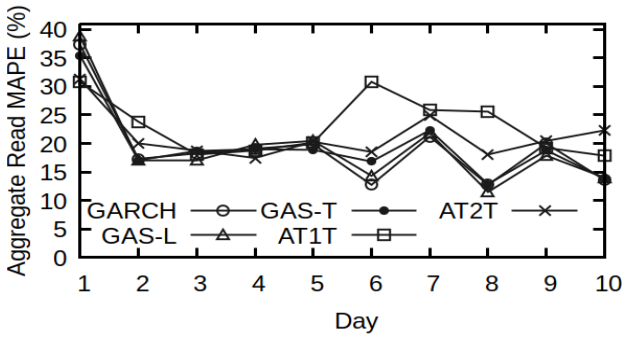


Figure 20: Aggregate hourly Read MAPE for 10 random days.

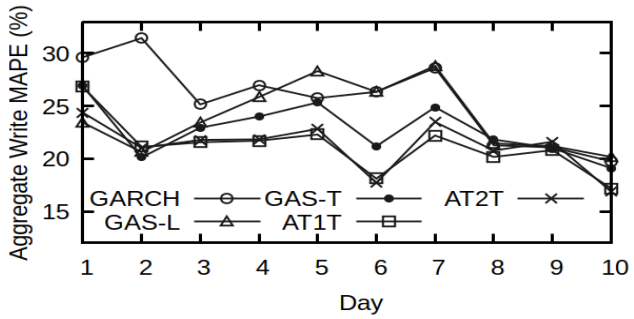


Figure 21: Aggregate hourly Write MAPE for 10 random days.

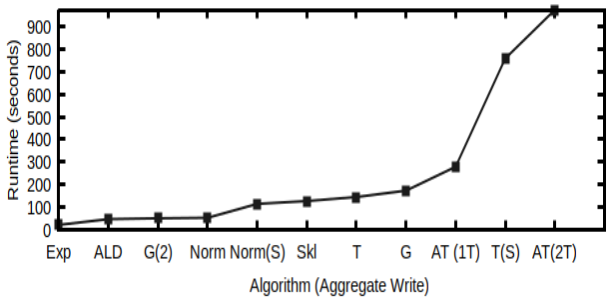


Figure 22: GAS algorithm runtimes for aggregate write.

Specifically, the AT(1T) and A(2T) improved forecasting efficiency by 20 to 25% compared to GARCH 80% of the time. This is illustrated in Figure 21. Furthermore, the 3-parameter models with mean, variance and skewness or shape (AT(1T), AT(2T), GAST(T)) outperformed the 2-parameter models (GARCH, GAS-L). While the principle of parsimony determines model selection is better with fewer parameters, a trade-off with performance in the light of the returned MAPE results should inform model selection. Figure 22 further illustrates the attendant runtime for all models evaluated ahead of selection. It shows the *t*-distribution family of models incur the longest runtimes while providing the best performance for the aggregate write traffic. The graph is similar for the aggregate read but not included given constraints.

Long-term Forecasting

Capacity assignment often requires resource provisioning spanning hours to days. For long-term forecasting, the models were trained for forecast horizons of 1 to 4 days. To this end, the forecast horizon the models were similarly trained on the

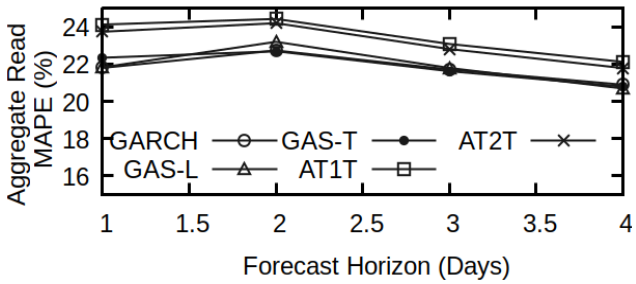


Figure 23: Aggregate Read MAPE for forecast over 4 days.

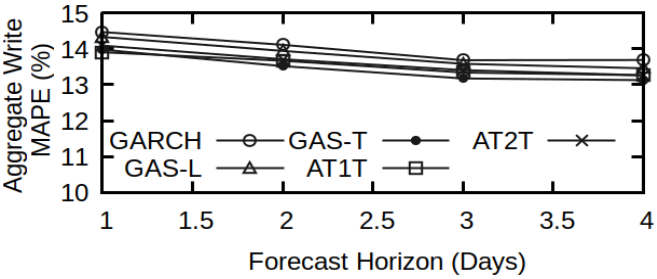


Figure 24: Aggregate Write MAPE for forecast over 4 days.

time-series data but with longer forecast horizons. Time-series lengths of 4 days were used to train the models with subsequent forecast horizons over 1 to 4 days. Figure 23 illustrates the read results where the GAS-T and GAS-L models perform equally well as GARCH. Figure 24 illustrates the write results where the t-distribution family outperforms GARCH with an 8% improvement in accuracy.

VII. Conclusion

This paper realized the aggregated read and write workload time-series of a recent trace from the Alibaba public cloud. Analysis of the time-series demonstrated time-of-day/week effects similarly observed in current research in cloud workloads. Observations of heavy-tails in time-series histograms enabled workload modeling with econometric models that track such workload dynamics with time-varying parameters of GAS models. A forecasting evaluation has further isolated models that provide improved performance with regard to the aggregate write traffic over state-of-the-art. These observations here can inform cloud storage capacity planning and forecasting. Further analysis is intended to explore the bimodality observed in the histogram of the aggregated write time-series. Future work intends to conduct an expanded forecasting evaluation over multiple forecast horizons. The dataset containing all the realized time-series from all the volumes in the trace is available online (https://www.kaggle.com/datasets/aadegboyega/cloud-storage-workload-research).

References

[1] Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu, "Everything as a Service (XaaS) on the Cloud: Origins, Current and

Future Trends," in *2015 IEEE 8th International Conference on Cloud Computing*, Jul. 2015, pp. 621–628. doi: 10.1109/CLOUD.2015.88.

[2] F. Romero *et al.*, "Faa$T: A Transparent Auto-Scaling Cache for Serverless Applications," in *Proceedings of the ACM Symposium on Cloud Computing*, in SoCC '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 122–137. doi: 10.1145/3472883.3486974.

[3] O. Hadary *et al.*, "Protean: VM Allocation Service at Scale," in *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation*, in OSDI'20. USA: USENIX Association, 2020.

[4] Q. Weng *et al.*, "MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters," in *Symposium on Networked Systems Design and Implementation*, 2022.

[5] I. K. Kim, W. Wang, Y. Qi, and M. Humphrey, "CloudInsight: Utilizing a Council of Experts to Predict Future Cloud Application Workloads," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, pp. 41–48. doi: 10.1109/CLOUD.2018.00013.

[6] Y. Ren *et al.*, "Dissecting the Workload of Cloud Storage System," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022, pp. 647–657. doi: 10.1109/ICDCS54860.2022.00068.

[7] Z. Chen, J. Hu, G. Min, A. Y. Zomaya, and T. El-Ghazawi, "Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 923–934, 2020, doi: 10.1109/TPDS.2019.2953745.

[8] C. Jiang *et al.*, "Characterizing Co-Located Workloads in Alibaba Cloud Datacenters," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2381–2397, 2022, doi: 10.1109/TCC.2020.3034500.

[9] W. Chen, K. Ye, Y. Wang, G. Xu, and C.-Z. Xu, "How Does the Workload Look Like in Production Cloud? Analysis and Clustering of Workloads on Alibaba Cluster Trace," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 2018, pp. 102–109. doi: 10.1109/PADSW.2018.8644579.

[10] C. Lu, W. Chen, K. Ye, and C.-Z. Xu, "Understanding the Workload Characteristics in Alibaba: A View from Directed Acyclic Graph Analysis," in *2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, 2020, pp. 1–8. doi: 10.1109/HPBDIS49115.2020.9130578.

[11] S. Ruan, Y. Wang, H. Jiang, W. Xu, and Q. Guan, "BatchLens: A Visualization Approach for Analyzing Batch Jobs in Cloud Systems," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 108–111. doi: 10.23919/DATE54114.2022.9774668.

[12] C. Liang, L. Deng, J. Zhu, Z. Cao, and C. Li, "Cloud Storage I/O Load Prediction Based on XB-IOPS Feature Engineering," in *2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 2022, pp. 54–60. doi: 10.1109/BigDataSecurityHPSCIDS54978.2022.00020.

[13] Q. Wang, J. Li, P. P. C. Lee, T. Ouyang, C. Shi, and L. Huang, "Separating Data via Block Invalidation Time Inference for Write Amplification Reduction in Log-Structured Storage," in *USENIX Conference on File and Storage Technologies*, 2021.

[14] Y. Zhou, F. Wang, Z. Shi, and D. Feng, "A Multi-Factor Adaptive Multi-Level Cooperative Replacement Policy in Block Storage Systems," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*, Oct. 2022, pp. 67–75. doi: 10.1109/ICCD56317.2022.00020.

[15] Q. Zou, Y. Zhu, J. Chen, Y. Deng, and X. Qin, "Characterization of I/O Behaviors in Cloud Storage Workloads," *IEEE Transactions on Computers*, pp. 1–14, 2023, doi: 10.1109/TC.2023.3263726.

[16] J. Li, Q. Wang, P. P. C. Lee, and C. Shi, "An In-Depth Analysis of Cloud Block Storage Workloads in Large-Scale Production," in *2020 IEEE International Symposium on Workload Characterization (IISWC)*, Oct. 2020, pp. 37–47. doi: 10.1109/IISWC50251.2020.00013.

[17] M. Wajahat, A. Yele, T. Estro, A. Gandhi, and E. Zadok, "Distribution Fitting and Performance Modeling for Storage Traces," in *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Oct. 2019, pp. 138–151. doi: 10.1109/MASCOTS.2019.00024.

[18] Q. Zou and B. Mao, "Revisiting Temporal Storage I/O Behaviors of Smartphone Applications: Analysis and Synthesis," in *2022 IEEE International Symposium on Workload Characterization (IISWC)*, Nov. 2022, pp. 215–227. doi: 10.1109/IISWC55918.2022.00027.

[19] T. Rakthanmanon, E. J. Keogh, S. Lonardi, and S. Evans, "Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data," in *2011 IEEE 11th International Conference on Data Mining*, Dec. 2011, pp. 547–556. doi: 10.1109/ICDM.2011.146.

[20] A. V. Timoshenko, V. I. Goncharenko, A. Yu. Perlov, and V. A. Pankratov, "Algorithm for Assessing the Technical Condition of the Monitoring Radar Based on the Clustering of Multidimensional Nonuniform Time Series," *Russian Aeronautics*, vol. 65, no. 4, pp. 855–862, Oct. 2022, doi: 10.3103/S1068799822040262.

[21] V. K. Naik, K. Beaty, N. Vogl, and J. Sanchez, "Workload Monitoring in Hybrid Clouds," in *2013 IEEE Sixth International Conference on Cloud Computing*, Jul. 2013, pp. 816–822. doi: 10.1109/CLOUD.2013.145.

[22] V. P. Nzanzu *et al.*, "FEDARGOS-V1: A Monitoring Architecture for Federated Cloud Computing Infrastructures," *IEEE Access*, vol. 10, pp. 133557–133573, 2022, doi: 10.1109/ACCESS.2022.3231622.

[23] A. Adegboyega, "Time-series models for cloud workload prediction: A comparison," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 298–307. doi: 10.23919/INM.2017.7987292.

[24] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2014. [Online]. Available: https://books.google.ca/books?id=nmTQwAEACAAJ

[25] J. D. Cryer and K. S. Chan, *Time Series Analysis: With Applications in R*. in Springer Texts in Statistics. Springer New York, 2008. [Online]. Available: https://books.google.ca/books?id=bHke2k-QYP4C

[26] H. Bozdogan, "Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, Sep. 1987, doi: 10.1007/BF02294361.

[27] Z. Amekraz and M. Y. Hadi, "An Adaptive Workload Prediction Strategy for Non-Gaussian Cloud Service Using ARMA Model with Higher Order Statistics," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, Jul. 2018, pp. 646–651. doi: 10.1109/CLOUD.2018.00089.

[28] D. Creal, S. J. Koopman, and A. Lucas, "GENERALIZED AUTOREGRESSIVE SCORE MODELS WITH APPLICATIONS," *Journal of Applied Econometrics*, vol. 28, no. 5, pp. 777–795, 2013, doi: https://doi.org/10.1002/jae.1279.

[29] A. Harvey, "Beta-t-(E)GARCH," 2008, doi: 10.17863/CAM.5286.

[30] J. J. Dziak, D. L. Coffman, S. T. Lanza, R. Li, and L. S. Jermiin, "Sensitivity and specificity of information criteria," *Briefings in Bioinformatics*, vol. 21, no. 2, pp. 553–565, Mar. 2020, doi: 10.1093/bib/bbz016.

[31] D. Ardia, K. Boudt, and L. Catania, "Generalized Autoregressive Score Models in R: The GAS Package," *J. Stat. Soft.*, vol. 88, no. 6, pp. 1–28, Jan. 2019, doi: 10.18637/jss.v088.i06.