Time Series Analysis for Efficient Sample Transfers

Hemanta Sapkota, Bahadir A. Pehlivan, Engin Arslan

University of Nevada - Reno

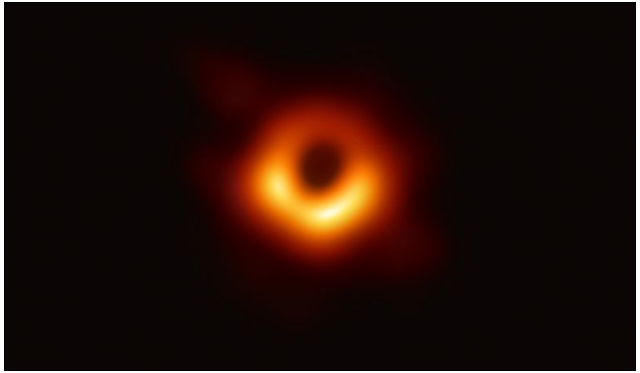
Big Data Era

New Gen Aircrafts will be generating 5 terabytes of data per flight

Facebook generates

4 petabytes new data per day

Each telescope of EHT generates 350 terabytes of data per day



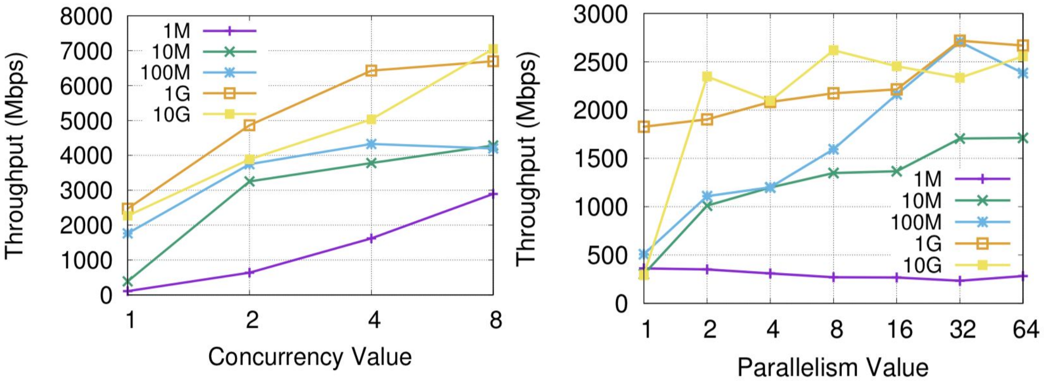
Underutilization of High Speed Networks

There exists 100 Gbps high speed networks but they are not utilized efficiently. Some reasons are:

- **CPU** limitations
- Poor **file system** performance
- **Buffer size** performance
- **Transport protocol** inefficiency

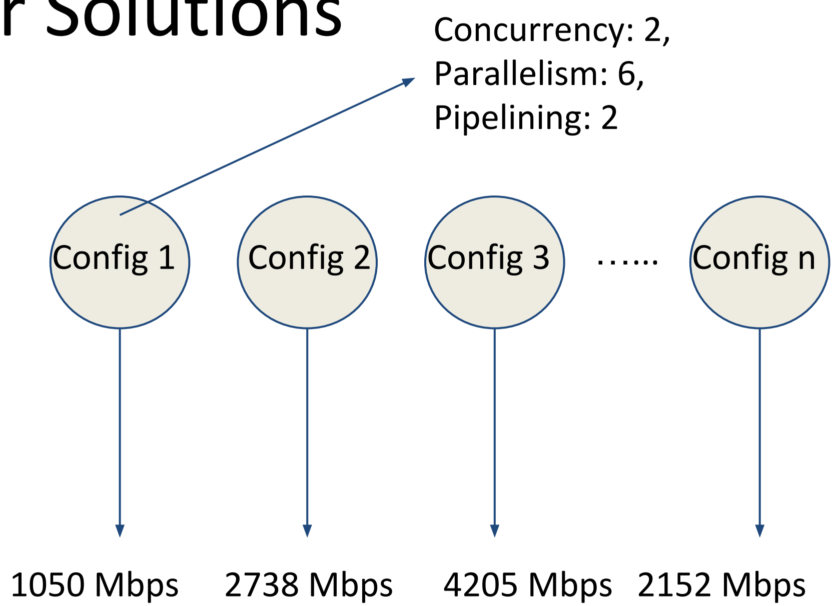
Application Layer Solutions

- Different application layer **parameters impact throughput** of transfer.
- But **maximum throughput yielding parameter is not always known** and vary with network conditions.
- So, to find a good parameter configuration which yields maximum throughput, we want to **do multiple sample transfer with different parameter configuration**.



Application Layer Solutions

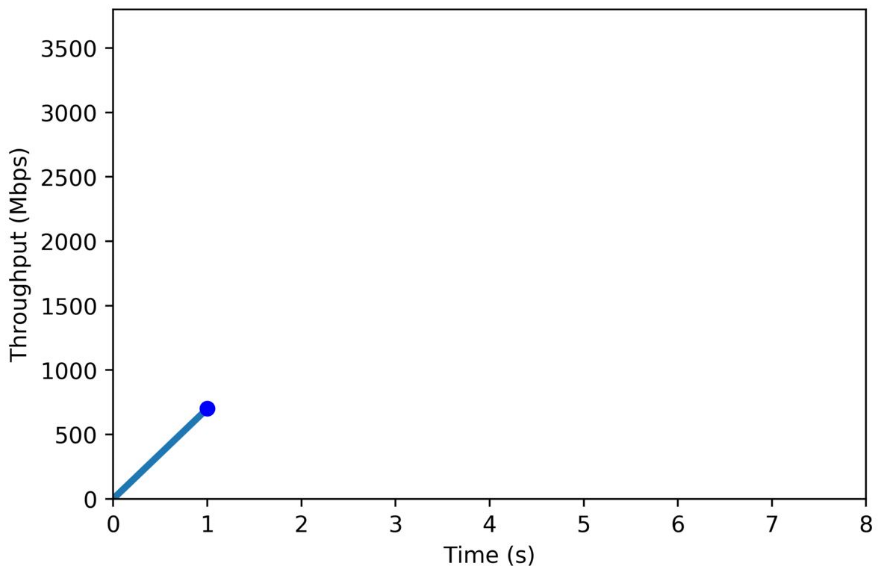
- Let's say we have large file to transfer, then we would run **sample transfer** and predict the throughput with each configuration, that way we can **choose configuration with maximum throughput**.
- In the following example we would use **configuration 3** to transfer the whole file.



Configurations: Parallelism, Pipelining, Concurrency, Buffer Size.

Problem

- So now the **problem is predicting convergence throughput as fast as we can**.
- In this paper we will show experiments involving different methods and **find a method which predicts convergence throughput with high accuracy and fast.**



Related Work: Fixed Data Size

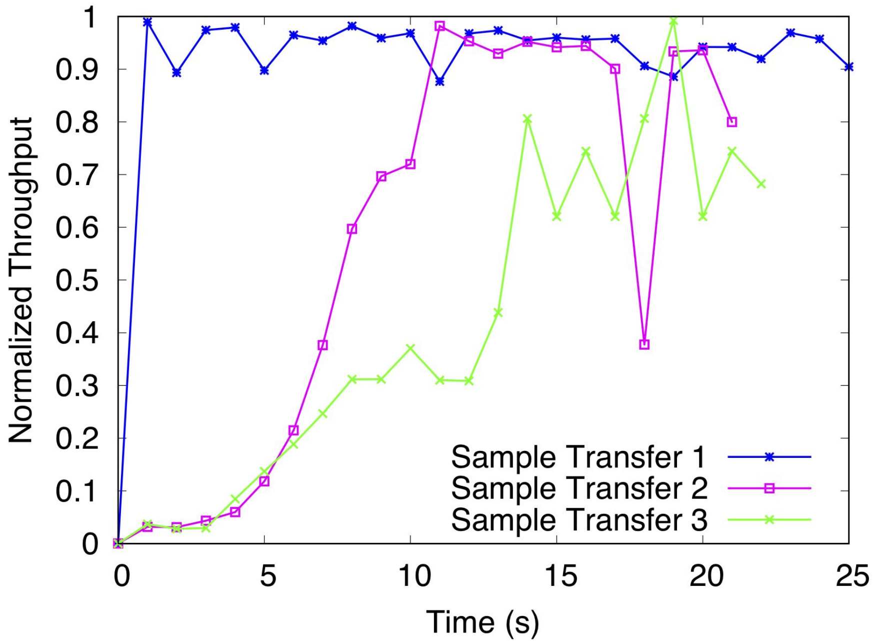
- In this approach, certain **percentage of data (e.g. 15%)** is transferred.
- And the predicted average throughput is the **average throughput of that percentage of data transfer**.
- Problem with this approach is that 15% of large file (e.g. 100GB) is very large and 15% of small file (e.g. 1GB) is very small. This will produce **long convergence time** in some cases and **high error rate** in other cases.

Related Work: Adaptive

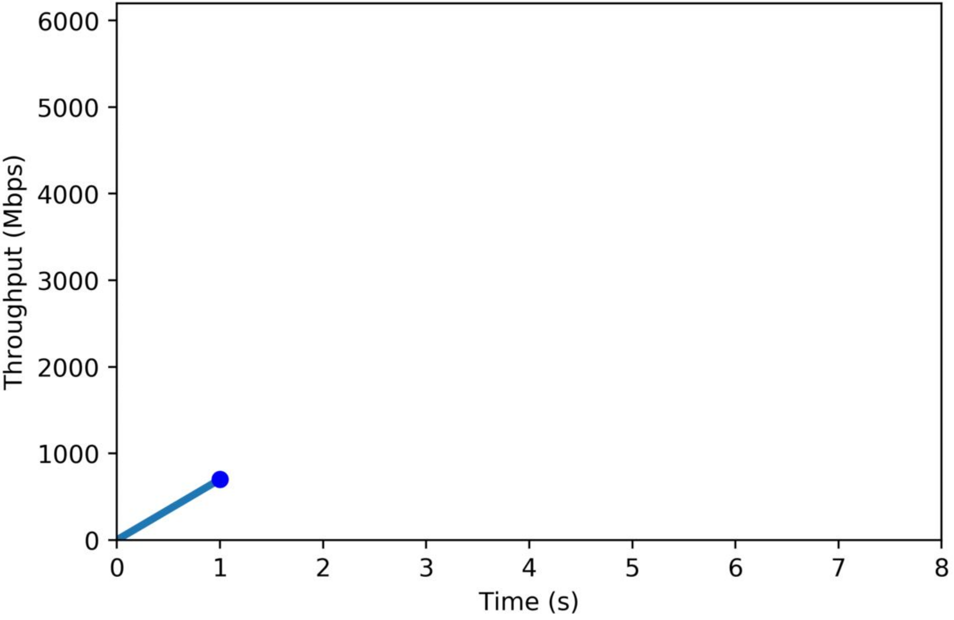
- This approach was defined in “Big data transfer optimization through adaptive parameter tuning” paper.
- The basic idea is that we will only say the throughput converged, when **two points are close to certain threshold.**
- And we use average of those two throughput as predicted average throughput.
- But **fluctuation on network throughput** can have impact on this approach.

Related Work: Adaptive

- Sometimes throughput converges fast and has less fluctuation like in **Sample Transfer 1.**
- But most of the time **there is network fluctuation** like in **Sample transfer 3** and adaptive approach would predict average throughput at 10 sec, but it is converging at 15 sec.



Time Series Analysis For Throughput Prediction



- **Initial points** are fed into model.
- Model makes a **prediction**.
- When value of next point in time is known, **If prediction is close** enough algorithm terminates.
- Otherwise it predicts next point considering recently added data as well.
- When it terminates **predicted points are used** to estimate average throughput.

Time Series Models

- Autoregressive (AR) model
- Autoregressive Moving Average (ARMA) Model
- Autoregressive Integrated Moving Average (ARIMA) model

Autoregressive (AR) Model

- The model uses last "**p**" points to fit the model and estimate all values of "φ" and "$\boldsymbol{\varepsilon}_t$".
- Here, the assumption is that final value is **linearly dependent on previous values** in addition with some error.
- Autoregressive model uses past forecast to predict the new forecast.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t.$$

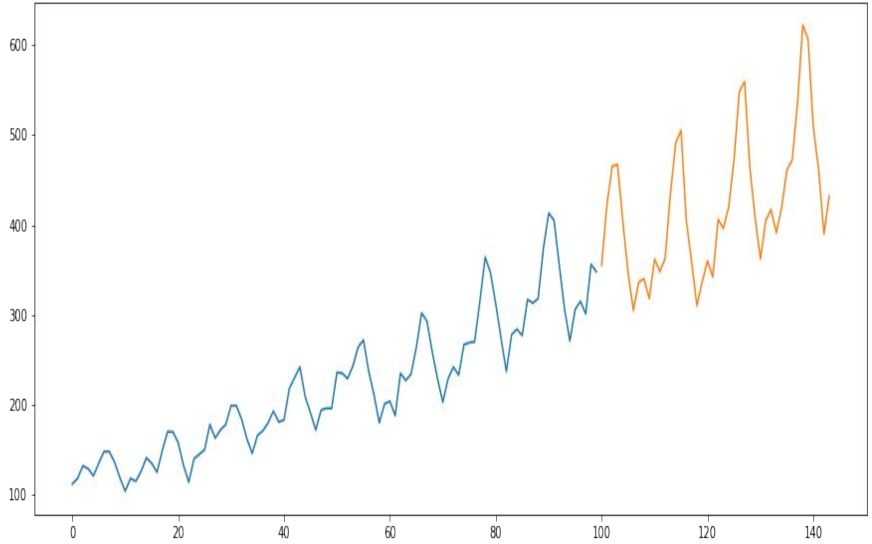
ARMA Model

- In addition to Autoregressive, ARMA also uses Moving average part.
- Moving average uses **error of previous forecast to make new prediction**. So, as we have more data to fit, the error starts to fade away.

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

ARIMA Model

- In addition to Autoregressive and moving average, ARIMA also **makes data stationary** if there is seasonality in the data.
- **Throughput doesn't seem to have any seasonality** in it because ARIMA performs worst of three Time series models.



EXPERIMENTS

System Specs of Experimental Networks

Specs	Storage	Memory (GB)	Bandwidth (Gbps)	RTT (ms)	Transfer Count
XSEDE	Lustre	96	10	40	28,209
ESnet	RAID-0	128	100	89	5,218
Pronghorn	GPFS	192	10	0.1	2,316
HPCLab	NVMe SSD	64	40	0.1	16,383
Total					**52,126**

What Do We Measure?

We use two metrics to measure the performance of models.

- **Convergence Time:** The time it takes for model to have a prediction.
- **Error Rate:** Percentage closeness of estimation to real average throughput.

Experiments - Optimal Convergence Time

- Takes average of last four points, stops when it is close enough to the **actual throughput of whole transfer**, which we know beforehand.
- We will calculate optimal convergence time **for different threshold**. If threshold is 20%, it will stop when the **average of last four points is less than equal to 20%.**
- **Time it requires to stop** is Optimal Convergence time.

Experiments - Optimal Convergence Time

Threshold **10%**, average throughput: **200Mbps**

40, 90, 150, 190 (**Average:** 117.5) (**Closeness:** (200-117.5)/200=**41.25%**)

40, **90, 150, 190, 160** (**Average:** 147.5) (**Closeness:**(200-147.5)/200=**26.25%**)

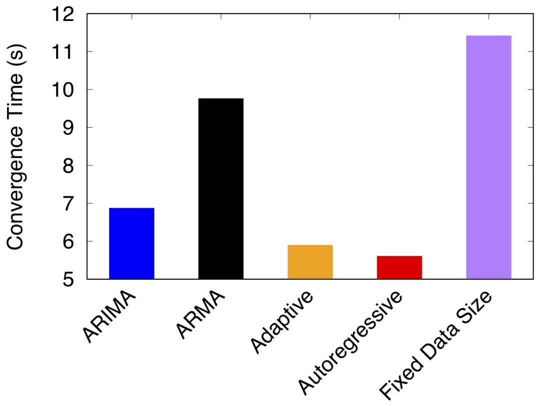
40, 90, **150, 190, 160, 210** (**Average:** 177.5) (**Closenes:** (200-177.5)/200=**11.25%**)

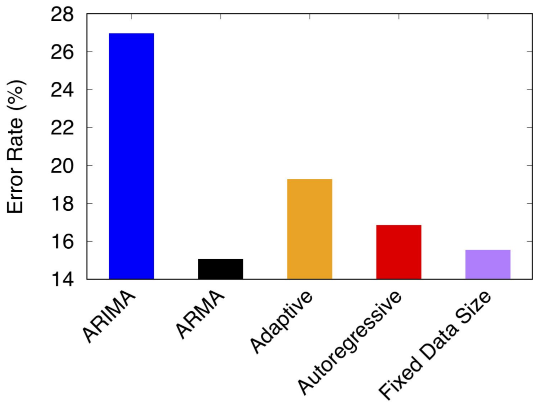
40, 90, 150, **190, 160, 210, 200** (**Average:** 190) (**Closeness:** (200-190)/200=**5%**)

Error Rate: **5%** Optimal Convergence Time: **7 seconds**

Experiments - Overall

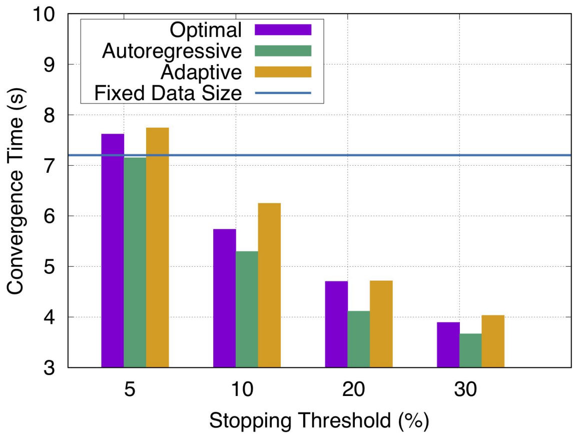
- Fixed data size has good error rate but **very long convergence time.**
- ARIMA and ARMA are performing bad in terms of **error rate and convergence time.**
- Adaptive is getting **high error rate on overall.**
- Autoregressive has **low convergence time and low error rate.**

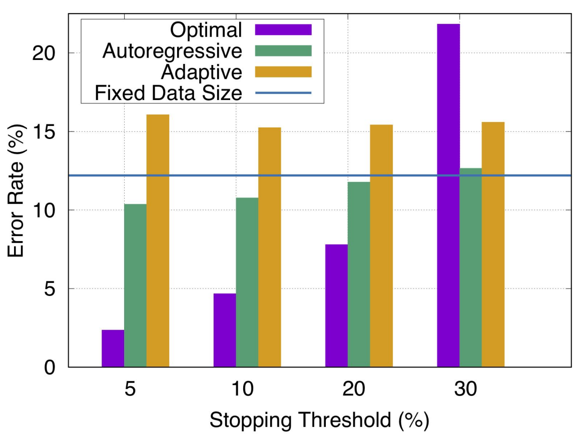




Experiments - HPCLab

- In terms of convergence time, Autoregressive is doing even **better than Optimal.**
- Adaptive and Fixed data size have **very high error rate** and **high convergence time.**

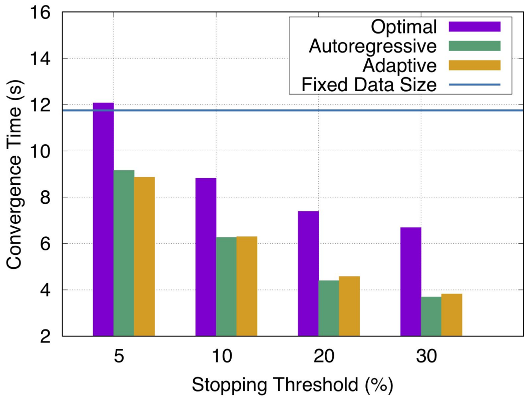


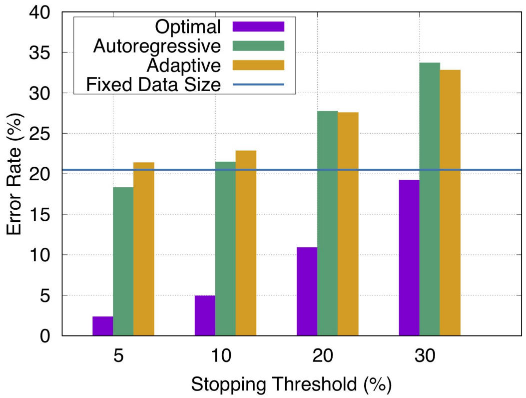


University of Nevada, Reno

Experiments - XSEDE Network

- Fixed Data Size is having **high convergence time** and about the same error rate with Autoregressive.
- Adaptive has slightly **higher convergence time and higher error rate.**



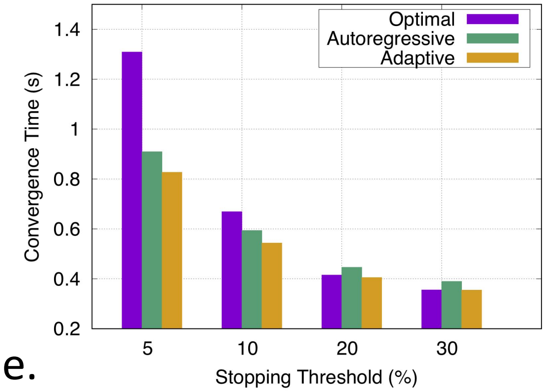


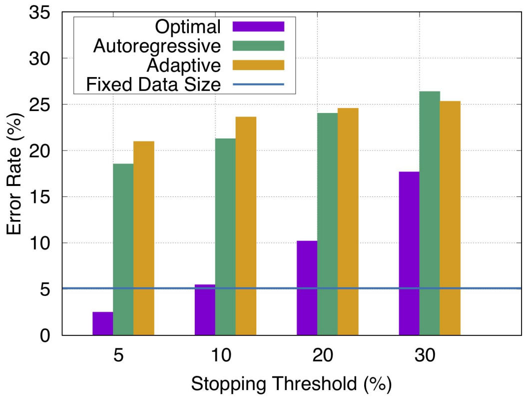
Data Collection with Higher Frequency

- So far on all the experiments done, throughput data was collected **once in every second**.
- So we tried to see if this experiment holds on throughput data collected with higher frequency, like we tried to experiment on throughput data collected **once in every 100 ms**.

Data Collection with Higher Frequency

- Fixed data size has **high convergence time** so it is not shown in figure.
- Autoregressive has **less error rate** than Adaptive but **longer convergence time by small margin.**
- With frequency data, convergence time can decrease.





University of Nevada, Reno

CONCLUSION

Conclusion

- Fixed data size model has a **high convergence time**. Adaptive approach has bad performance when there is **fluctuation on transfer throughput**.
- Autoregressive model has **lower convergence time (around 5 sec)** and **lower error rate (less than 20%) compared to actual average throughput** in most of the cases.
- And if the end goal is to have faster convergence time then, **frequency data can make 4-6x reduction in convergence time** with some increase in error rate.

Thank you

Future works

- Using neural nets to find the Convergence throughput seems to be working, we are currently doing research on that.
- And once we are confident with our results, we will start working on real-time throughput optimization problems.