A Data Throughput Prediction and Optimization Service for Widely Distributed Many-Task Computing

Dengpan Yin, Esma Yildirim and Tevfik Kosar*

Department of Computer Science
Center for Computation && Technology
Louisiana State University, Baton Rouge, LA 70803

MTAGS2009

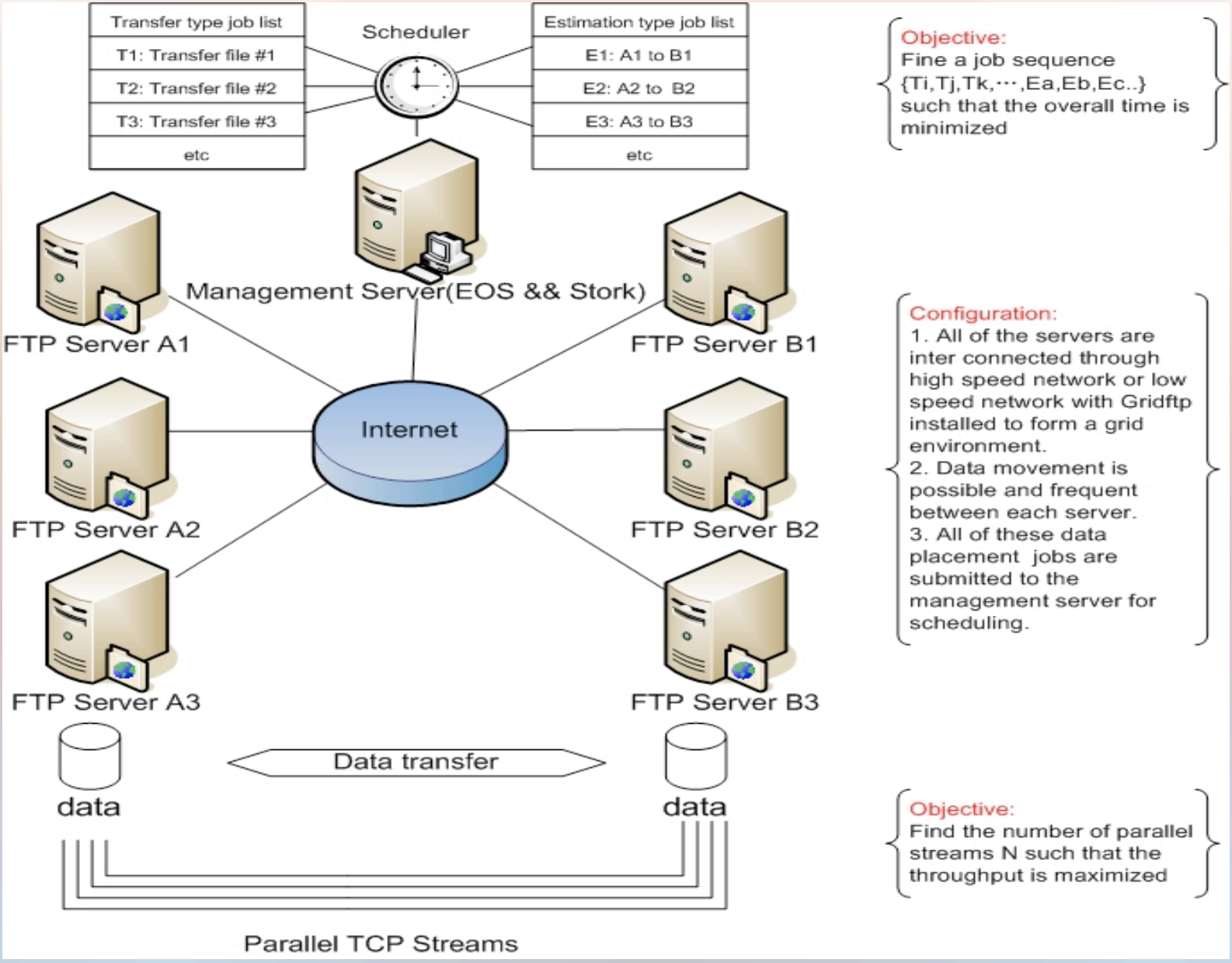
Presented by Dr. Tevfik Kosar

kosar@cct.lsu.edu

Outline

- Motivation
  - Grid and many-task environments configuration
  - Summary of objectives
- Significance
  - The throughput increases significantly with an optimal # of parallel streams
  - The time cost of each job is decreased significantly with the scheduling of Stork server and EOS.
- Approach
  - Overview of the estimation && optimization process
  - Model comparison and selection
  - Dynamic sampling and model instantiating
  - Many-tasks scheduling
- Experimental results

Grid and Many-task environments



Objective:
Fine a job sequence {Ti,Tj,Tk,⋯,Ea,Eb,Ec..} such that the overall time is minimized

Configuration:
1. All of the servers are inter connected through high speed network or low speed network with Gridftp installed to form a grid environment.
2. Data movement is possible and frequent between each server.
3. All of these data placement jobs are submitted to the management server for scheduling.

Objective:
Find the number of parallel streams N such that the throughput is maximized

Objectives

- Maximize the throughput by predicting the optimal number of parallel streams.

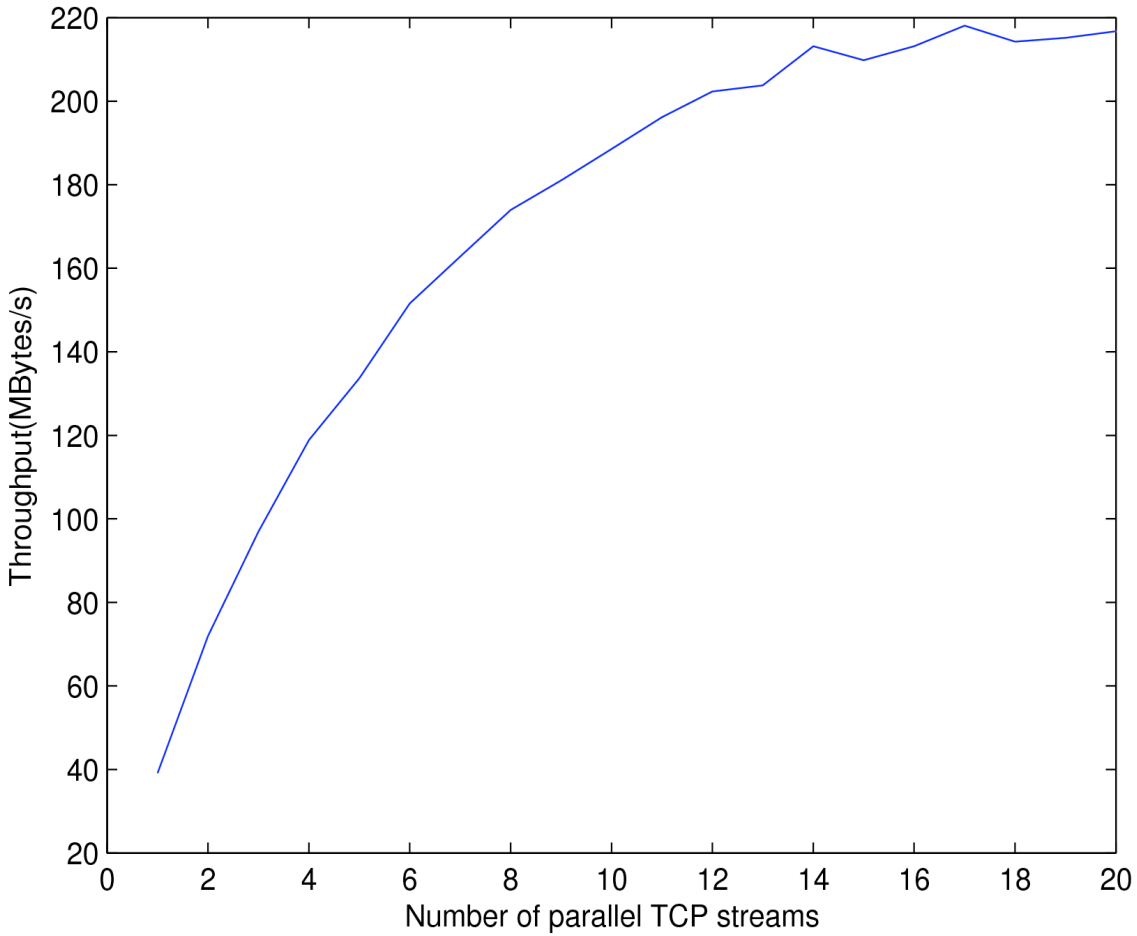
$$\hat{N} = \underset{N}{\text{argmax}} Throughput(N)$$

- Estimate the transfer time corresponding to the optimal number of parallel streams

$$Time = \frac{File\ size}{Throughput(\hat{N})}$$

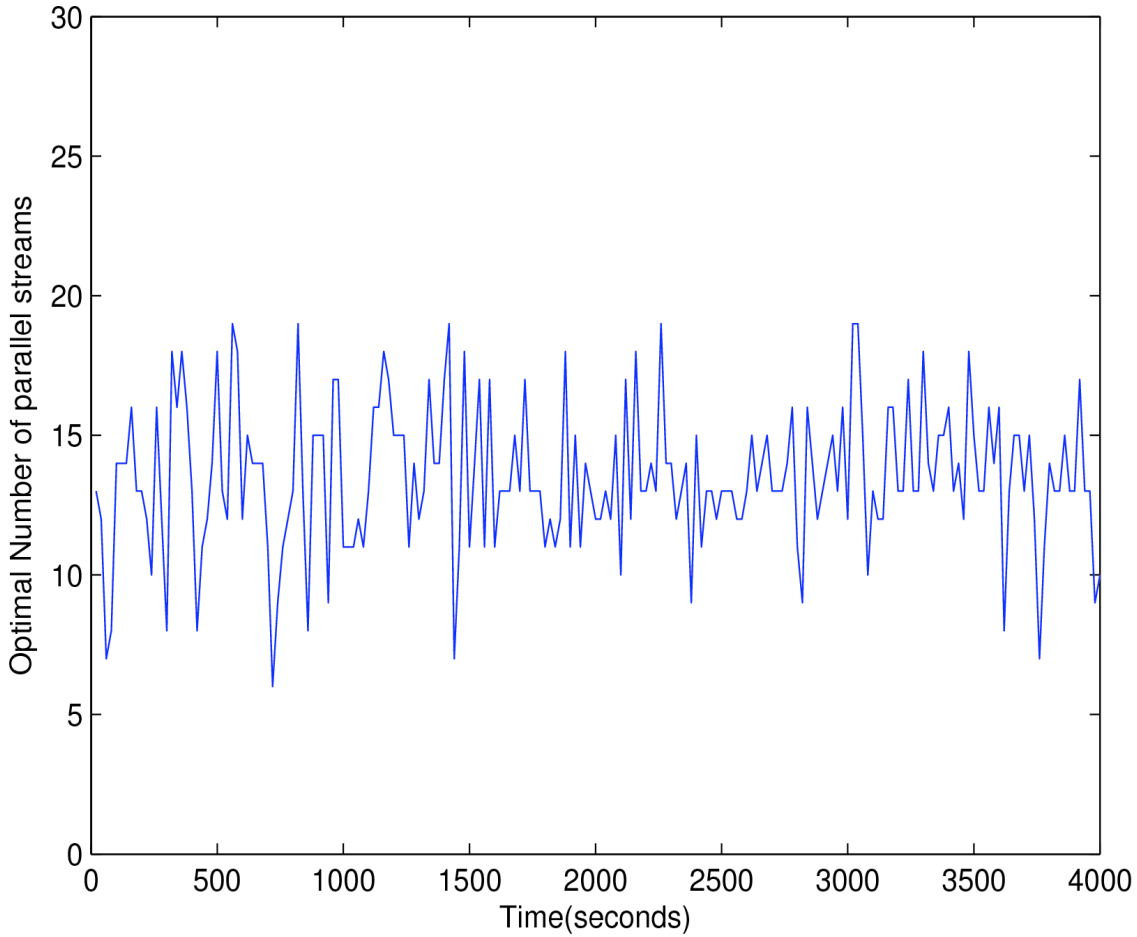
- Minimize the overall execution time by scheduling the estimation and optimization type jobs

Significance of predicting the optimal



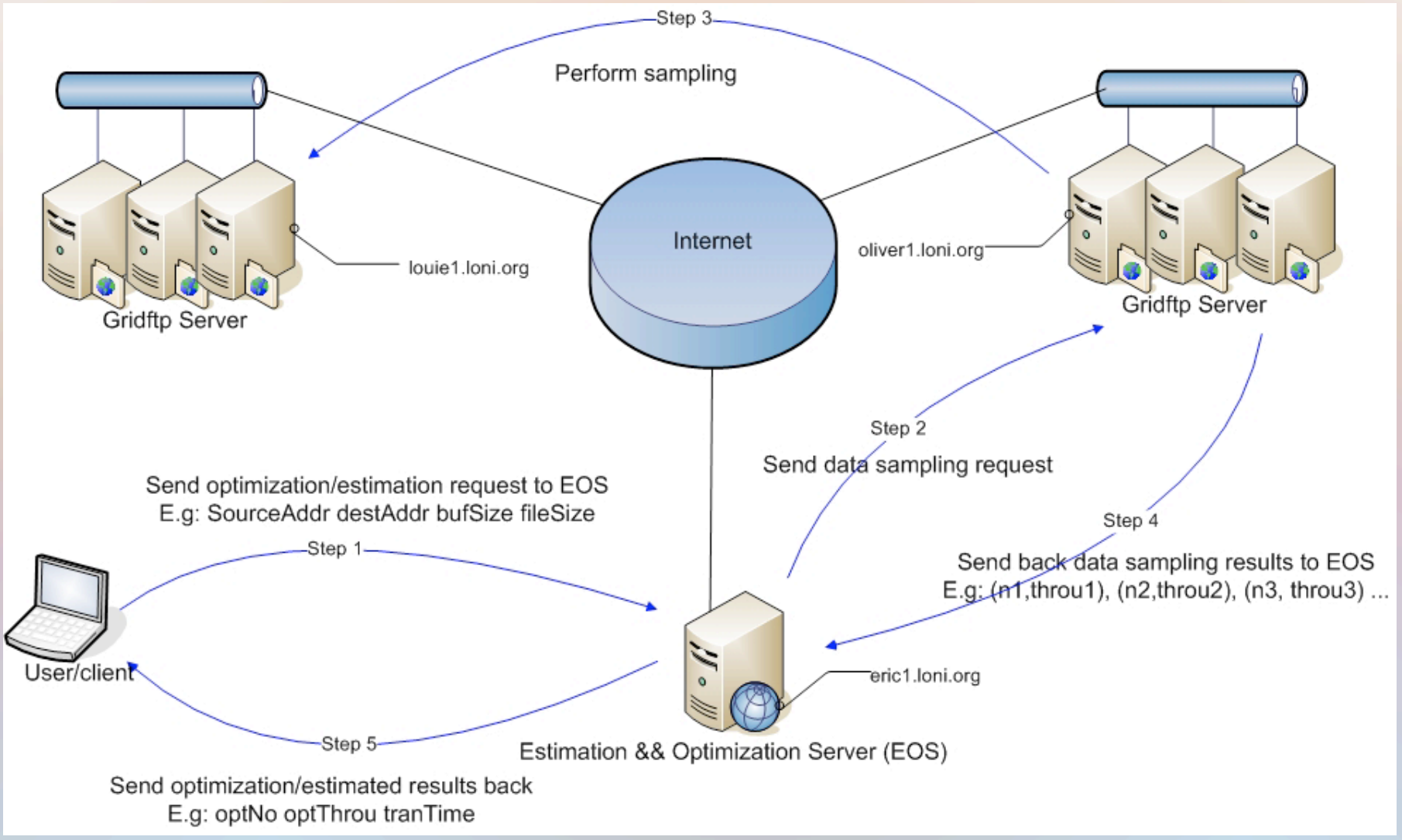
- The throughput increases significantly along with increasing the # of parallel streams when the parallel number is small.
- The throughout will be stable when the number is beyond its optimal value
- The optimal number of parallel streams is different when the source and destination are different.

Why not a constant # of parallel streams

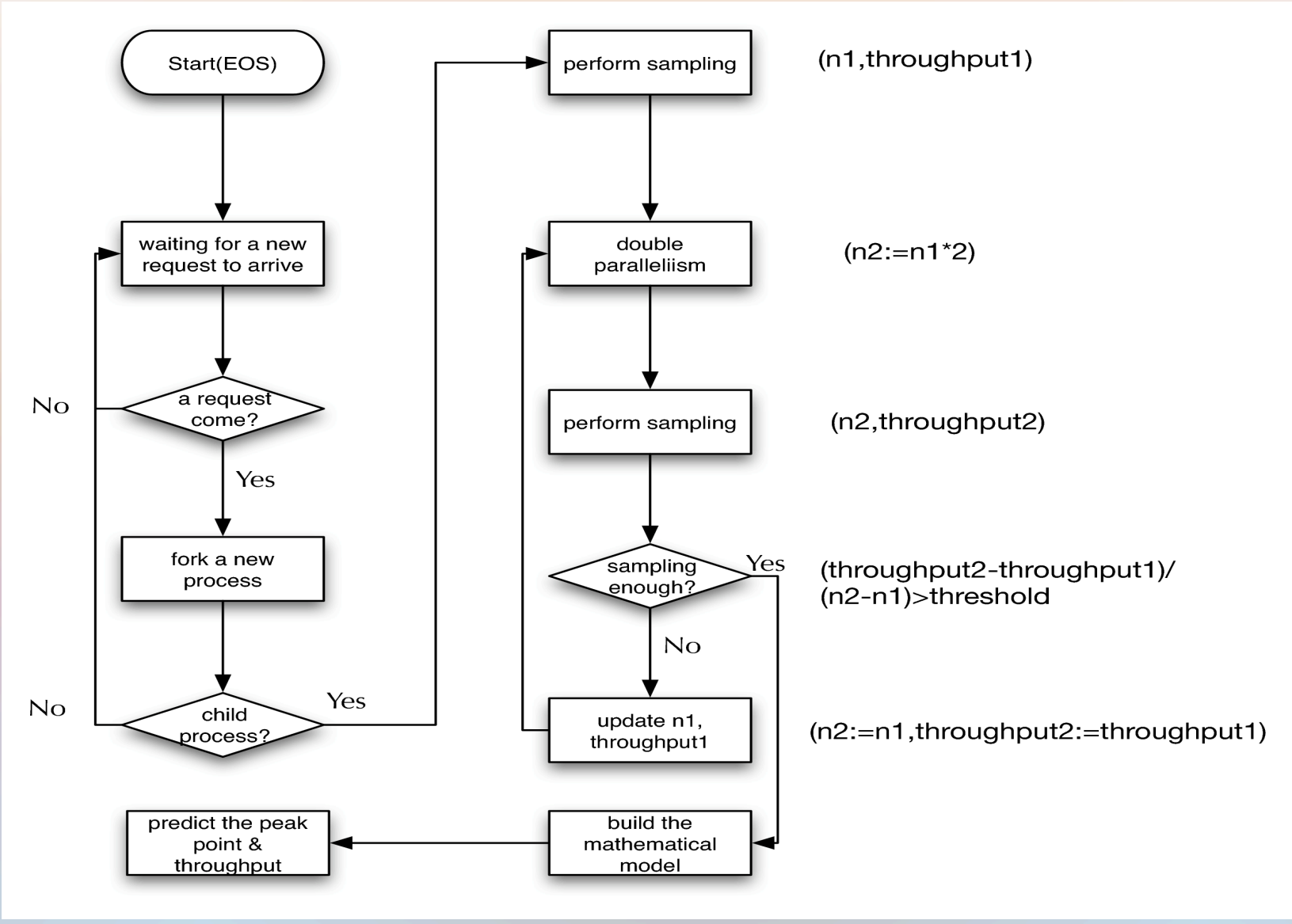


- Measured between two LONI clusters named Eric and Oliver for 4000 seconds.
- The results show that the optimal number of parallel streams fluctuates a lot along with time elapsing.
- The optimal number is different between different sites.

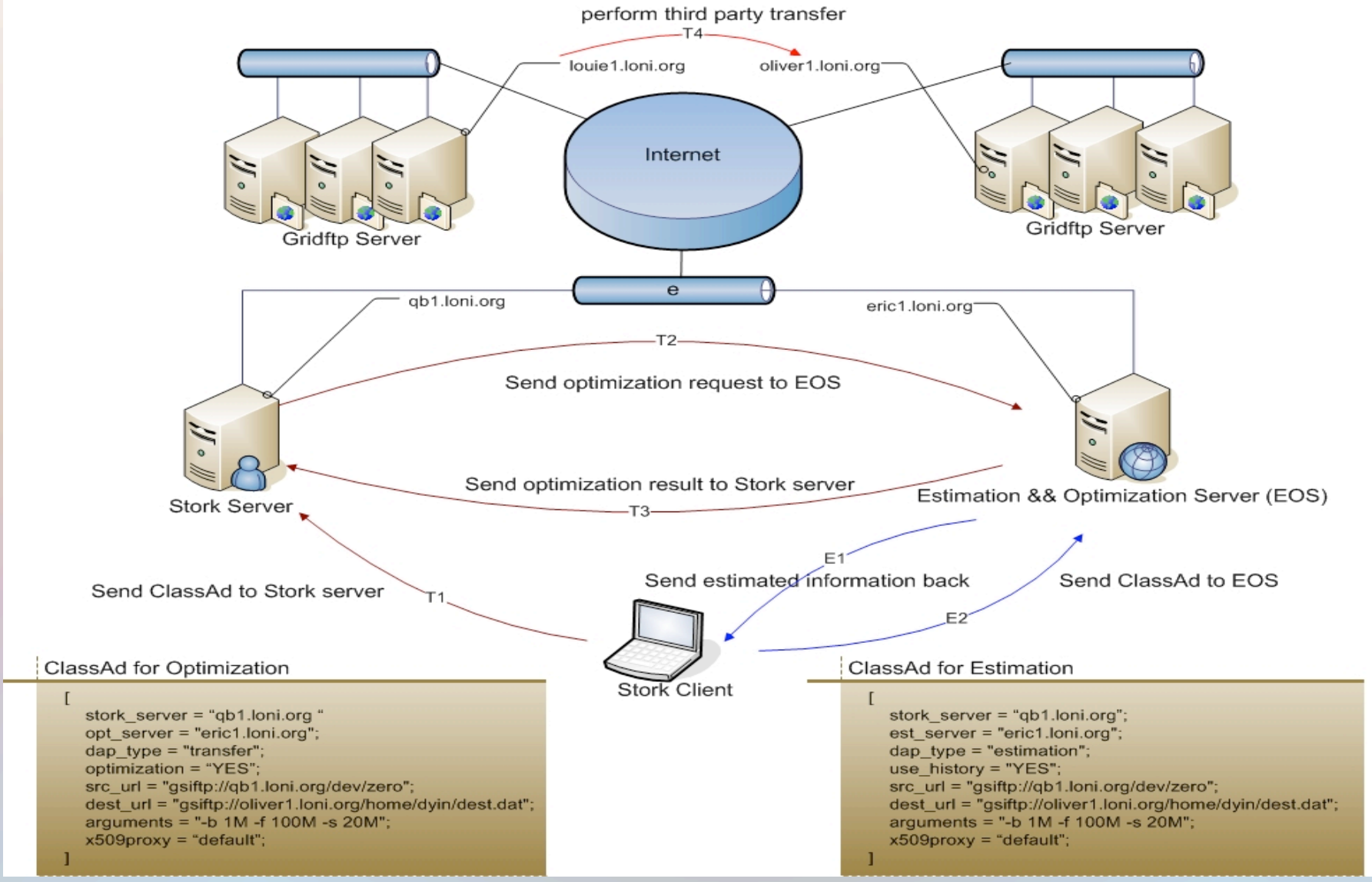
Overview of Estimation & optimization service(EOS)--standalone version



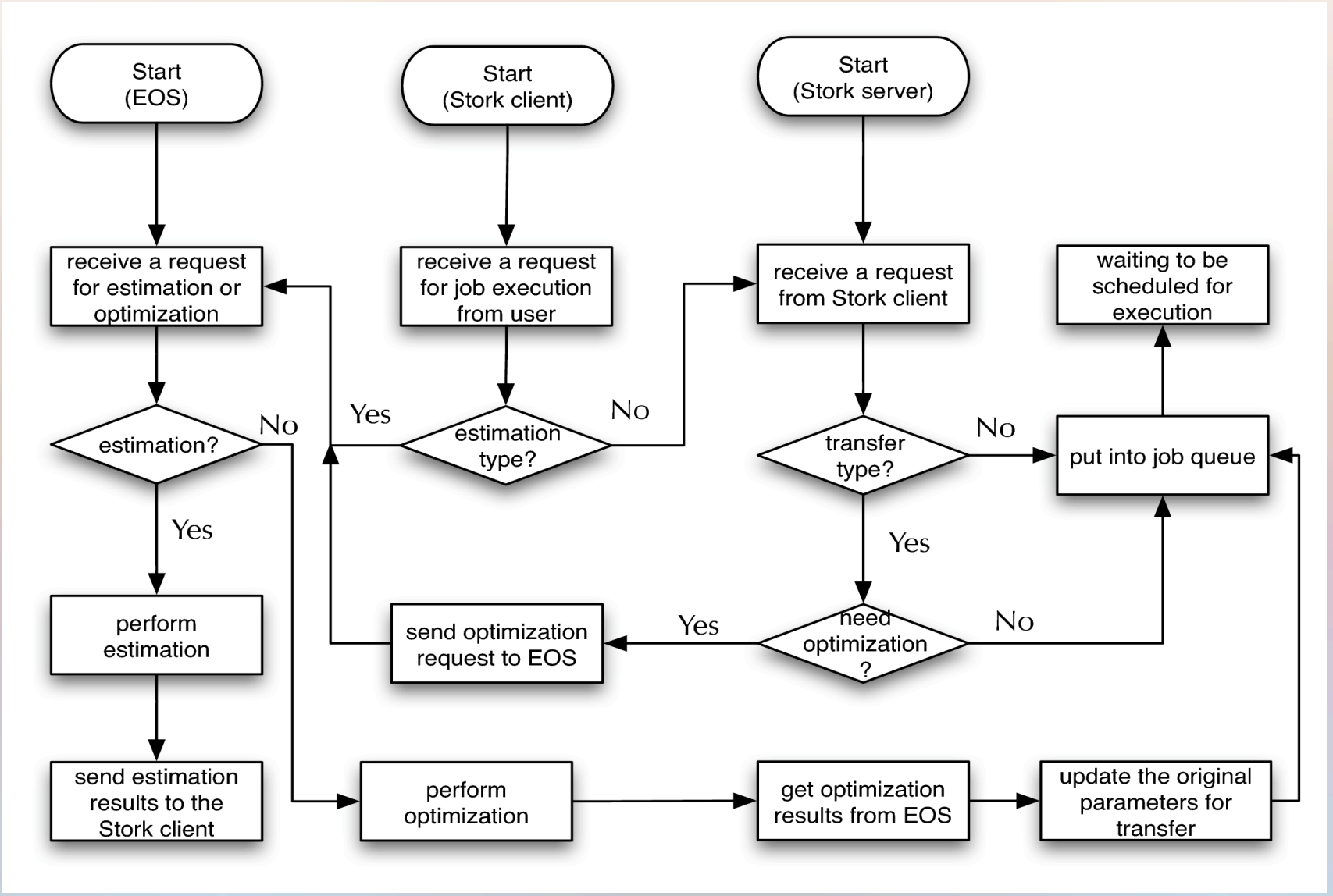
Flowchart of EOS



Overview of EOS and Stork



Flow char of EOS & Stork



Full second order Mathematical Model applied in EOS

$$Th_n = \frac{n}{\sqrt{p'_n}} = \frac{n}{\sqrt{a'n^2 + b'n + c'}}$$

$$Th_{n_1} = \frac{n_1}{\sqrt{a'n_1^2 + b'n + c'}}$$

$$Th_{n_2} = \frac{n_2}{\sqrt{a'n_2^2 + b'n + c'}}$$

$$Th_{n_3} = \frac{n_3}{\sqrt{a'n_3^2 + b'n + c'}}$$

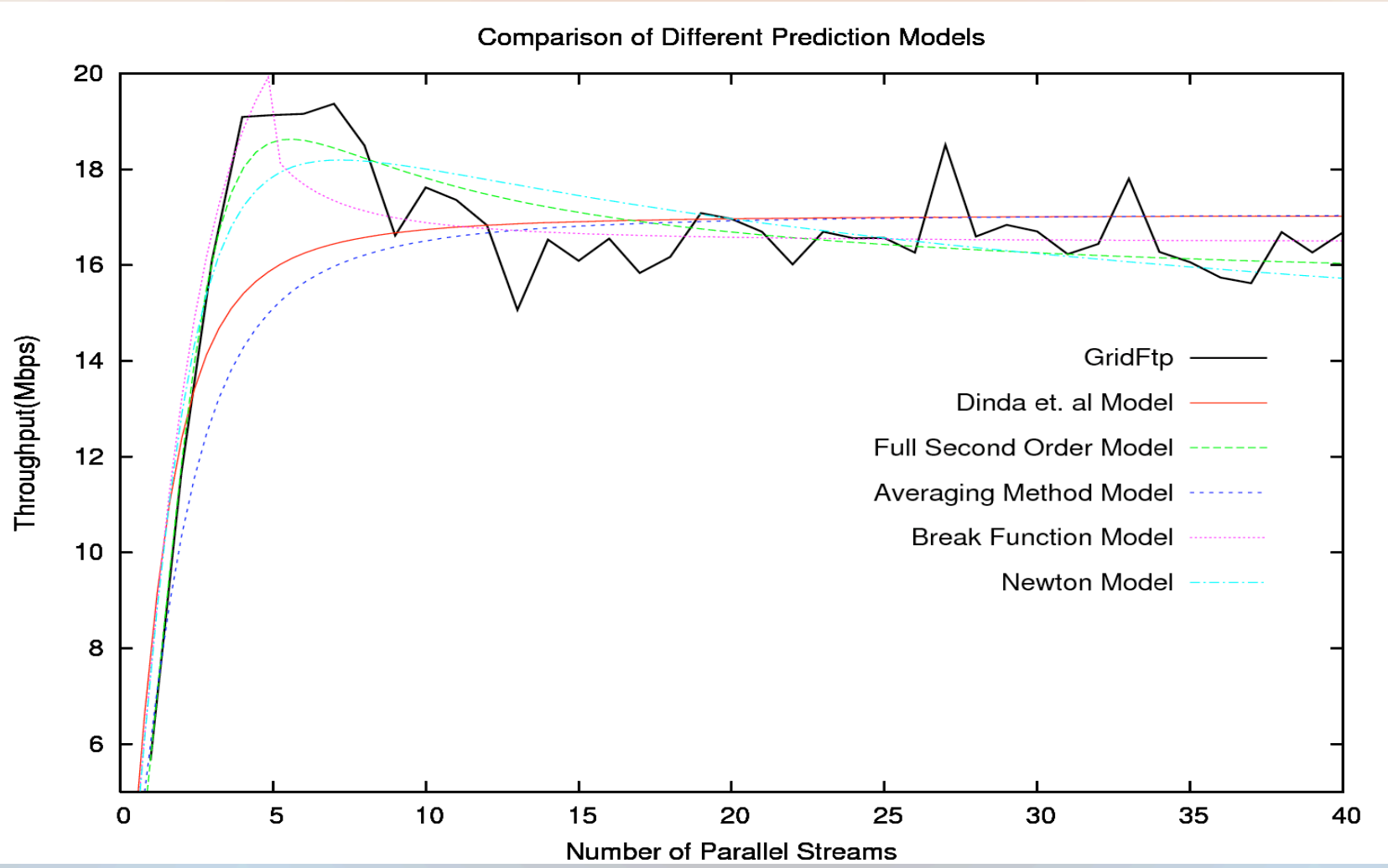
$$a' = \frac{\frac{\frac{n_3^2}{Th_{n_3}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_3 - n_1} - \frac{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_2 - n_1}}{n_3 - n_2}$$

$$b' = \frac{\frac{n_2^2}{Th_{n_2}^2} - \frac{n_1^2}{Th_{n_1}^2}}{n_2 - n_1} - (n_1 + n_2)a'$$

$$c' = \frac{n_1^2}{Th_{n_1}^2} - n_1^2 a' - n_1 b'$$

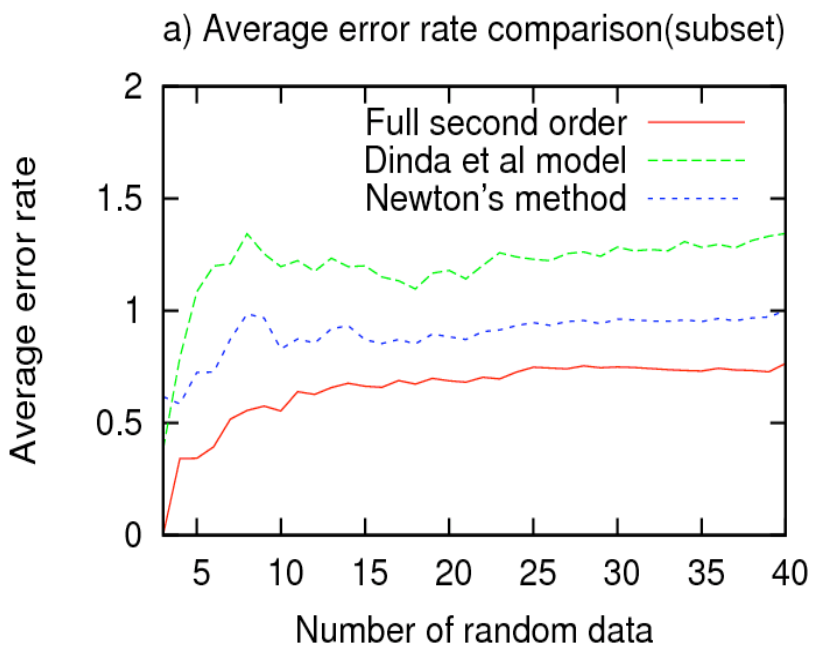
- We propose a throughput function with respect to the number of parallel streams and three unknown variables.
- Construct the corresponding equation systems using three sampling data points.
- Solve the equation systems and derive the three unknown parameters.

Why Full second order model



Error rate comparison between models

- The sampling data set
- The whole actual data set



Sampling strategy

Algorithm 2 Sampling data transfers

1: $threshold \leftarrow \alpha$
2: $streamNo1 \leftarrow 1$
3: $throughput1$ is the throughput corresponding to $streamNo1$
4: **repeat**
5: $streamNo2 \leftarrow 2 * streamNo1$
6: $throughput2$ is the throughput corresponding to $streamNo2$
7: $slop \leftarrow \frac{throughput2 - throughput1}{streamNo2 - streamNo1}$
8: $streamNo1 \leftarrow streamNo2$
9: $throughput1 \leftarrow throughput2$
10: **until** $slop < threshold$

- The sampling size affects the accuracy of the prediction. The larger, the better in accuracy. However, the overhead of sampling will be higher meanwhile.

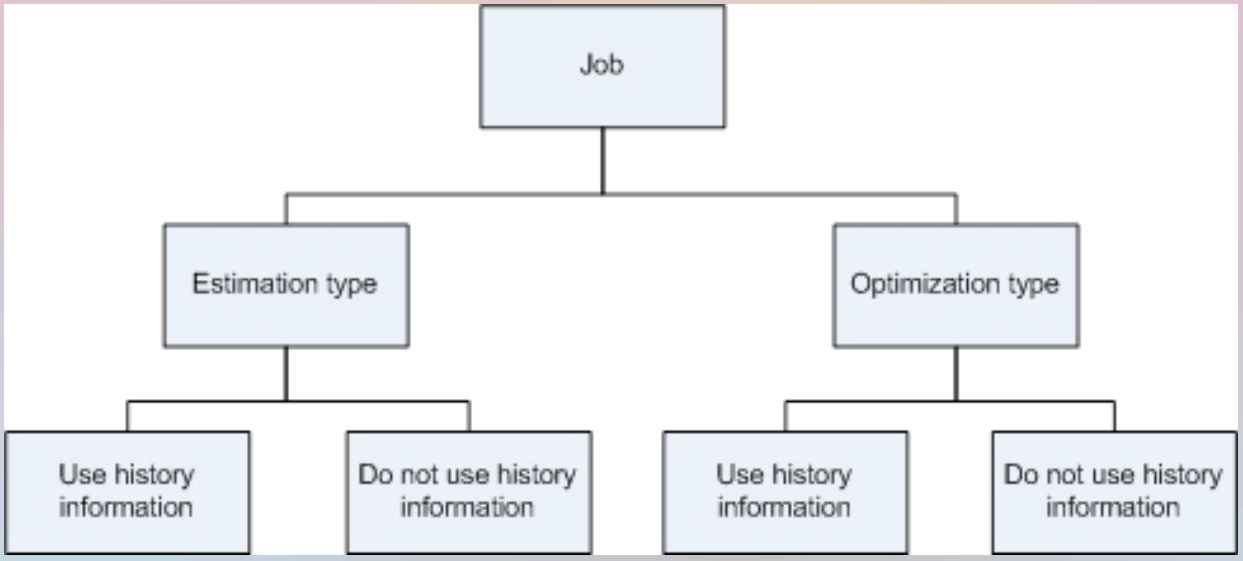
- The basic idea is exponentially increase the number of parallel streams until the throughput increases very slowly or start to decrease.
- Suppose the optimal number of parallel stream is N, then the number of sampling times is proportional to log(N)
- In most cases, N<64, so the number of sampling times is less than 7. Usually N is less than 20, so the number of sampling times is less than 5.

Model instantiation

- When the sampling data are collected, three of them are needed to construct the equation systems in order to calculate the unknown coefficients.
- Every combination that consists with three items from the sampling data is tested to calculate the coefficients. Choose the best combination which minimize the error rate of all the sampling data.
- Suppose there are N sampling data during the sampling phase, then there will be C(N,3)=N*(N-1)*(N-2)/6 different kinds of combination. This large number of candidate coefficients improve the accuracy and effectiveness of the model.
- Choosing the best model from C(N,3) is based on the error rate of the sampling data. It turns out this is also close to the optimal solution when considering all the actual data in most cases.

Many-task scheduling of EOS

- Tasks are submitted to the Stork scheduler via classAds, which is conceptually quite similar to Condor.
- We extended the classAds schema such that it has more features to satisfy our requirements.
- The tasks we considered in this paper are data intensive tasks such as data transfer estimation and optimization.



An example of ClassAds

```
[
 dap_type          =    "transfer";
 optimization      =    "YES";
 use_history       =    "NO";
 stork_server      =    "qb1.loni.org";
 opt_server        =    "qb1.loni.org";
 src_url           =    "gsiftp://eric1.loni.org/work/dyin/test1.dat";
 dest_url          =    "gsiftp://oliver1.loni.org/work/dyin/dest1.dat";
 arguments         =    "-tcp-bs 128K -s 30M";
 x509proxy         =    "default";
]
```

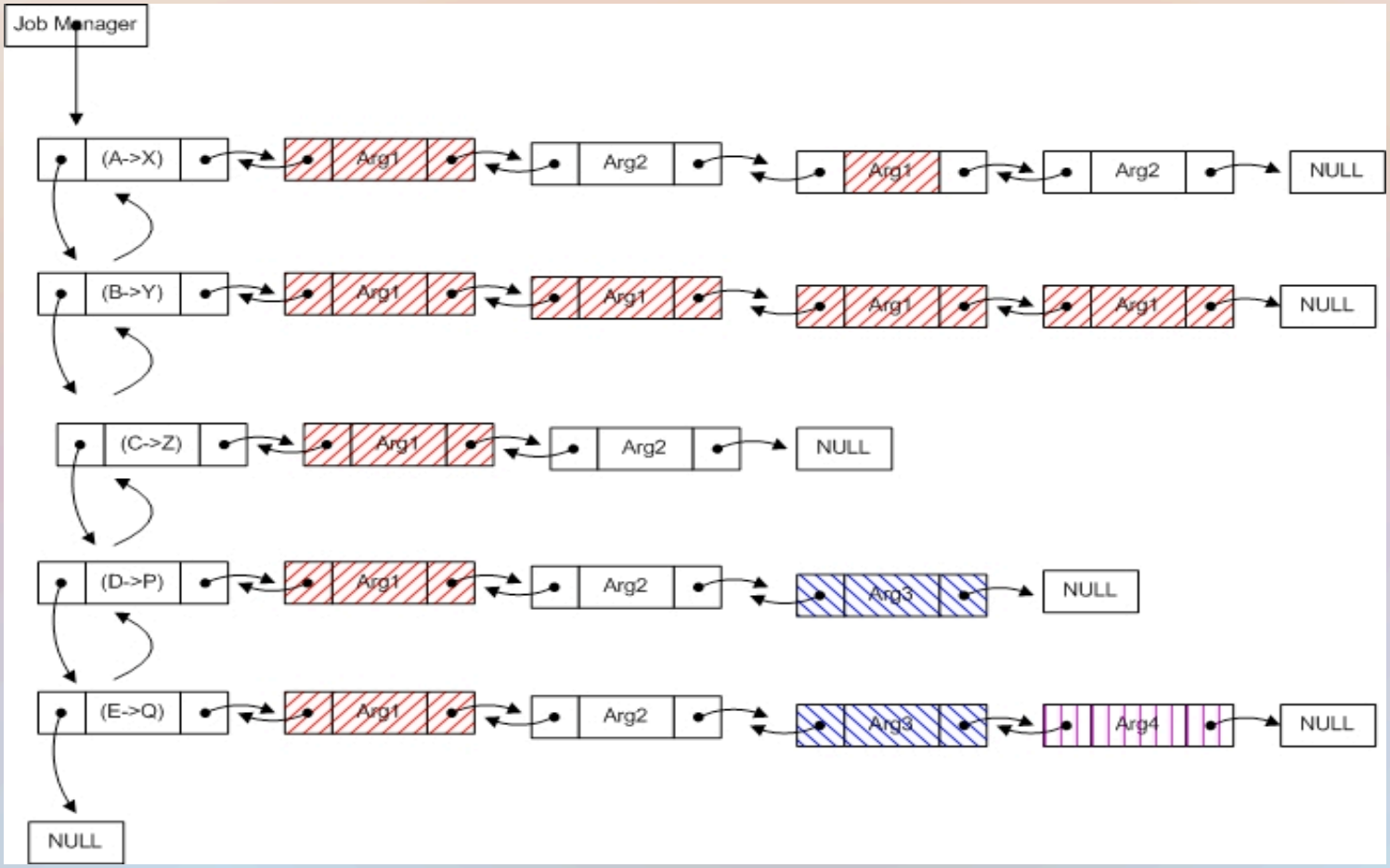
- dap_type: the type of the task, such as transfer, estimation
- Optimization: If it is 'YES' then this task will be optimized by EOS before execution
- use_histoy: If it is 'YES' then database will be checked.
- stork_server: The server with Stork scheduler installed.
- opt_server: The server with EOS installed. Not necessary the same to stork_server.
- Arguments: The arguments used for transfer or optimization.

EOS Scheduling

- If the task specifies that history information is used, then EOS will process it immediately since database operation is time efficient compared with model instantiation via sampling.
- The tasks that optimize the transfers from the same source and destination are put together in the same task list.
- Each task list corresponds to one thread in EOS. The task in the head of each list will be executed by the thread. All the other tasks in the same list will be removed from the list when the first task is finished provided that they have the same arguments.
- Different task lists form into another list. This list manages the concurrent threads being executed. The number of maximum threads can be configured by EOS.
- EOS scheduling is two dimensional. On one hand, it tries to decrease the number of tasks by task classification based on arguments, source and destination. On the other hand, it tries to enlarge the concurrency of tasks if they are not relevant to each other.

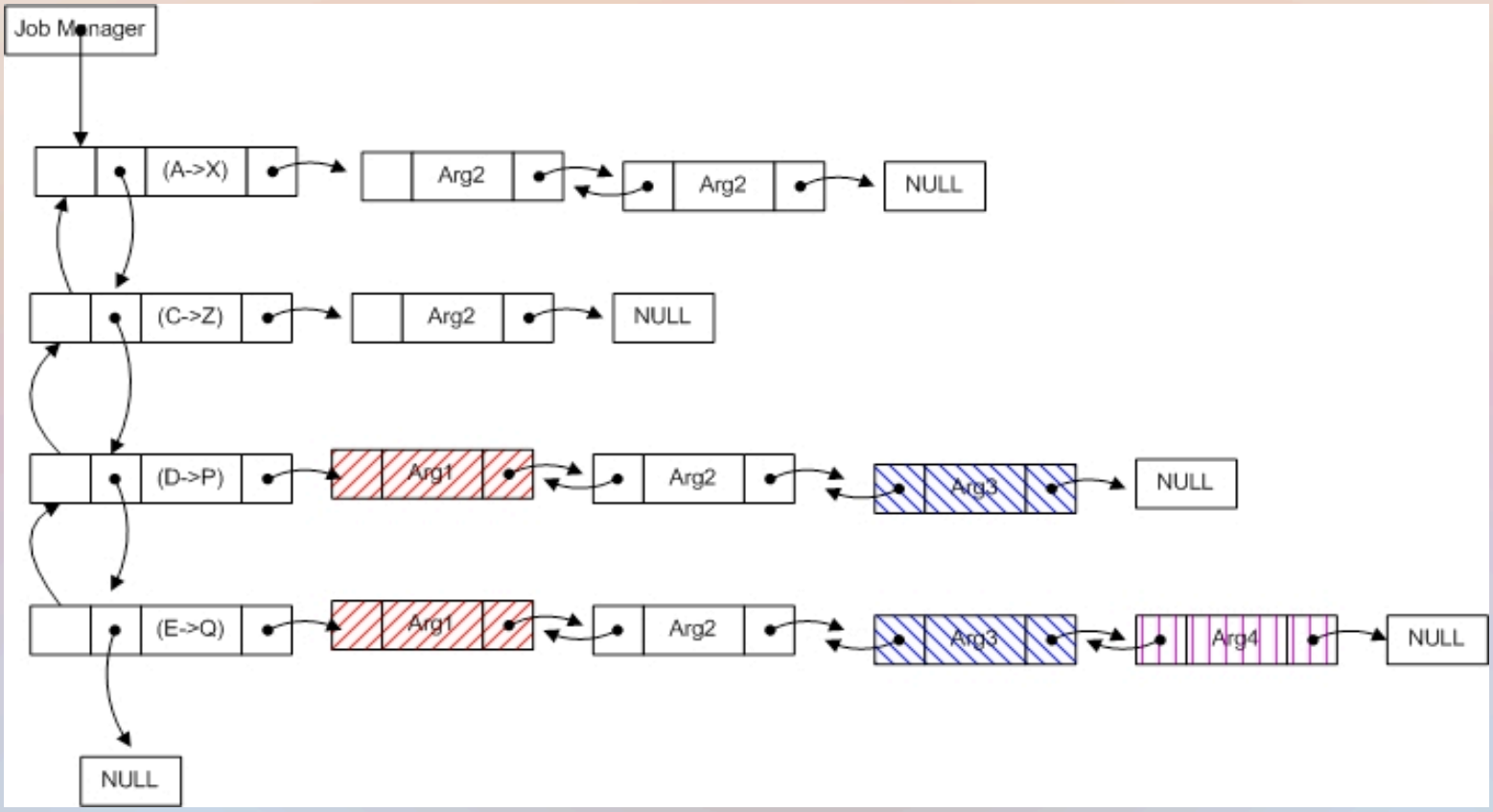
An illustration of EOS scheduling

- Suppose they are 5 task lists at given time T. The maximum number of concurrent tasks is limited to 3. (In reality, the task list can be several thousands, and the concurrent tasks can be several hundreds)



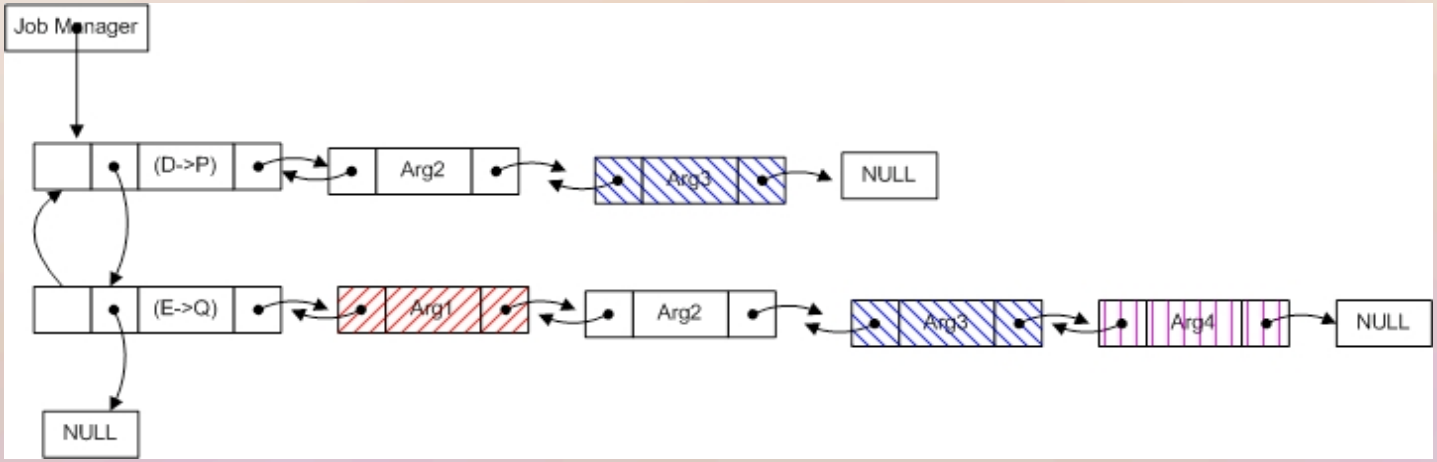
Step 1

- The nodes with red colors corresponding to the same arguments(arg1) are removed from the first three job lists. The following shows the results.

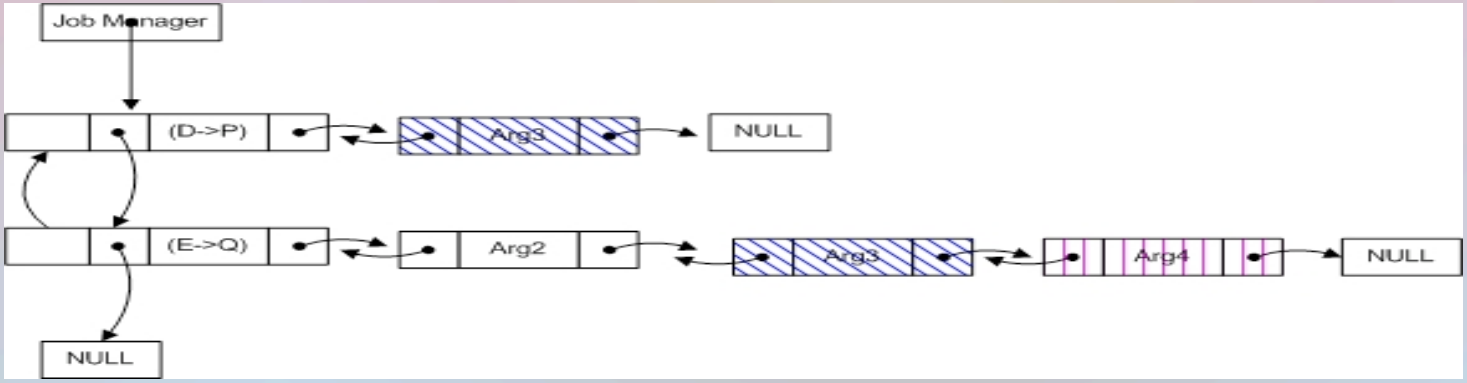


Step 2 && 3

- The nodes in the first two task lists are removed since they have the same arguments in each list. Only the first node in the third task list is removed since no other tasks have the same arguments to the head node.

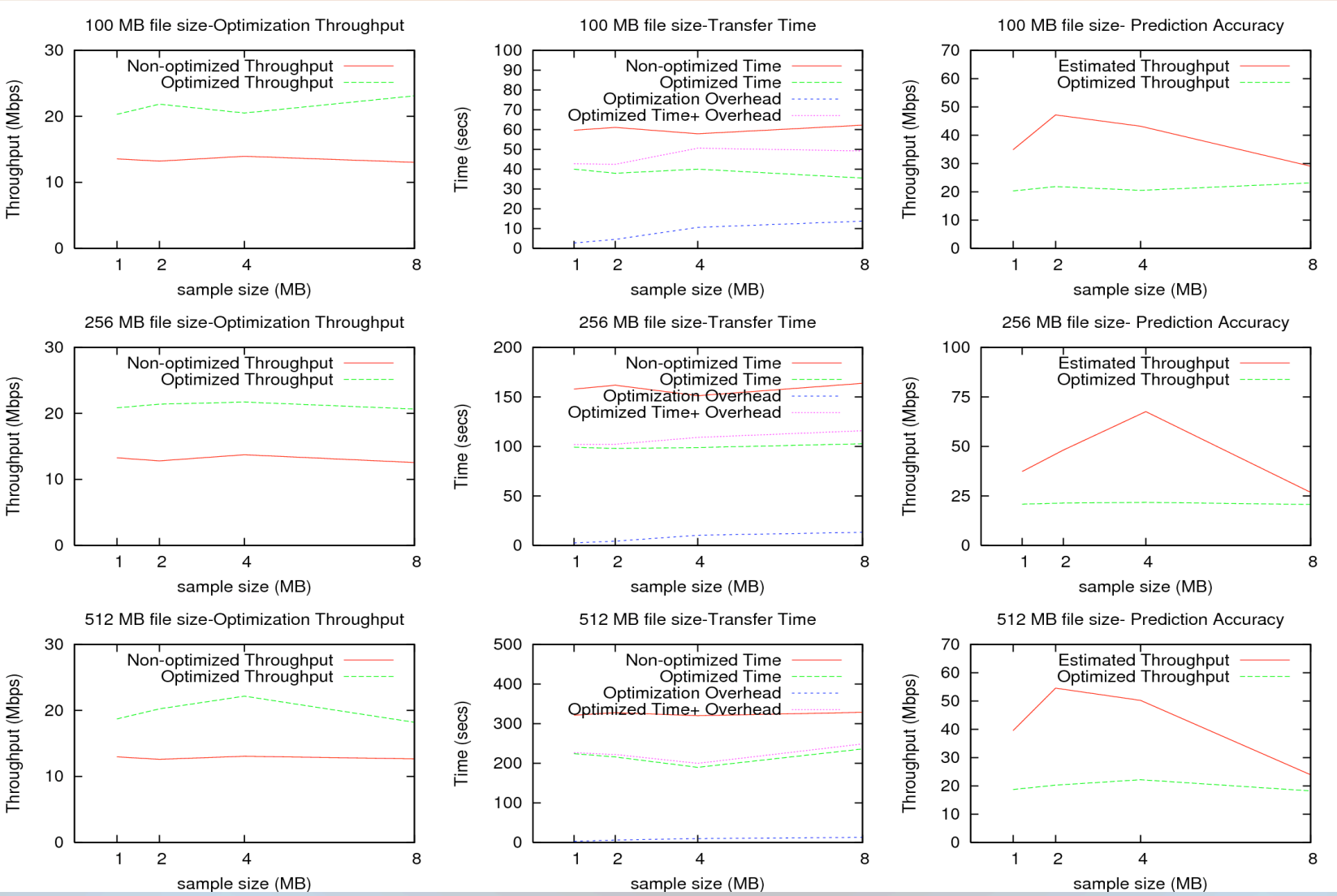


- The first node of each of these two task lists is removed.

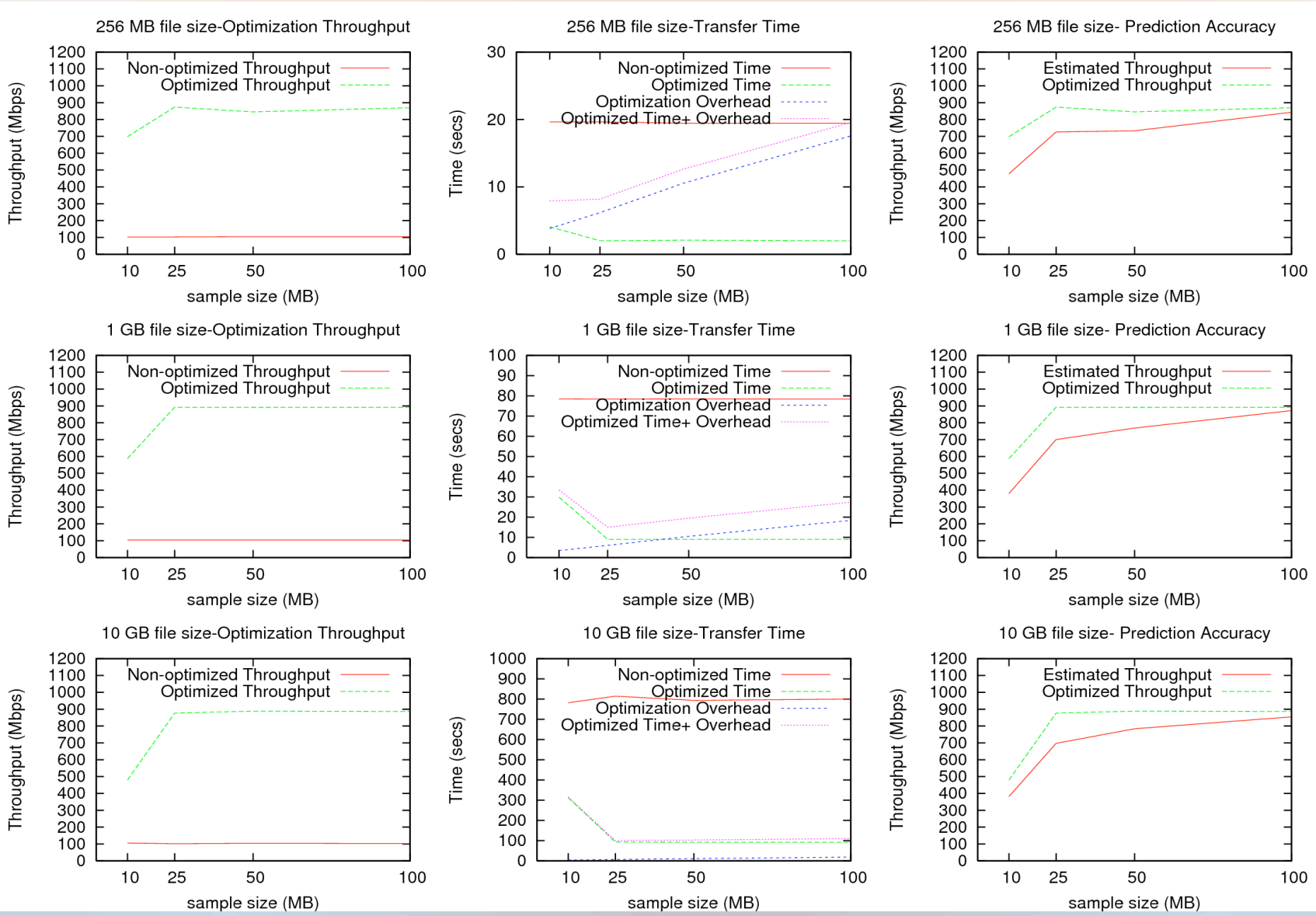


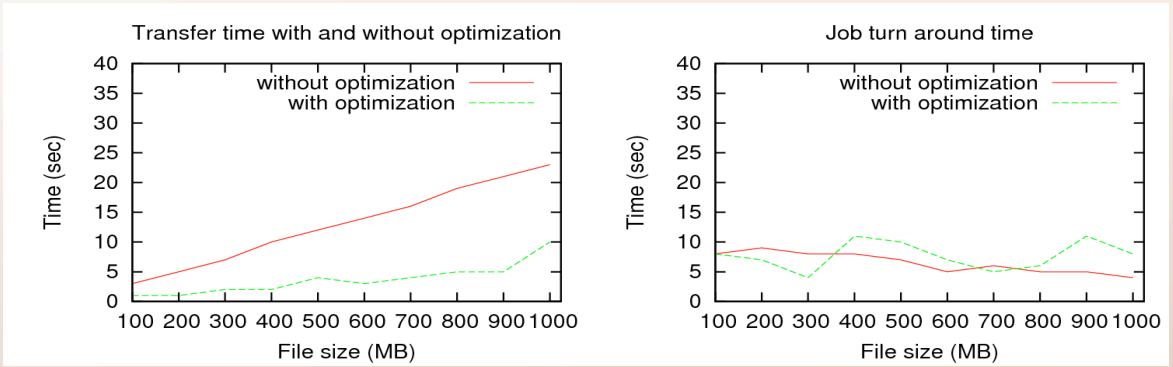
Experimental Results

Optimization results between LAN 100Mbps and LONI 1Gbps network interfaces based on GridFTP

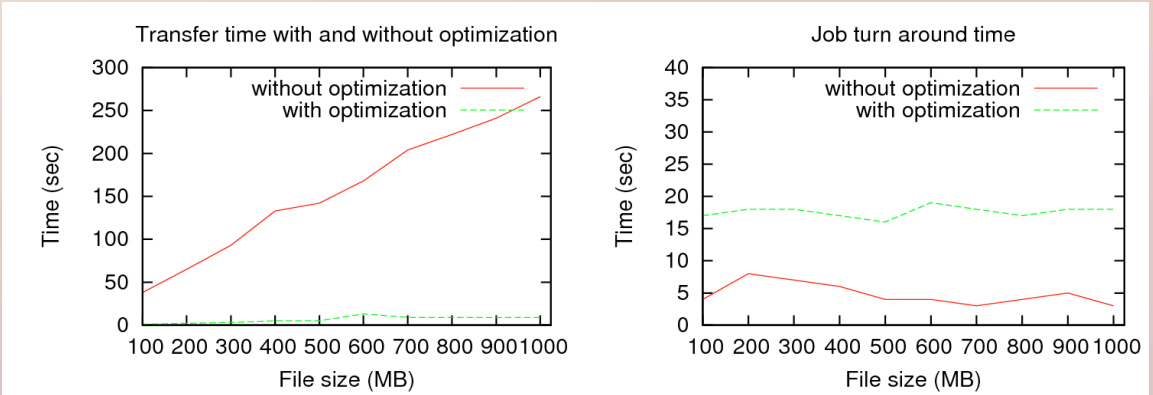


Optimization results over LONI network with 1Gbps network interfaces based on GridFTP

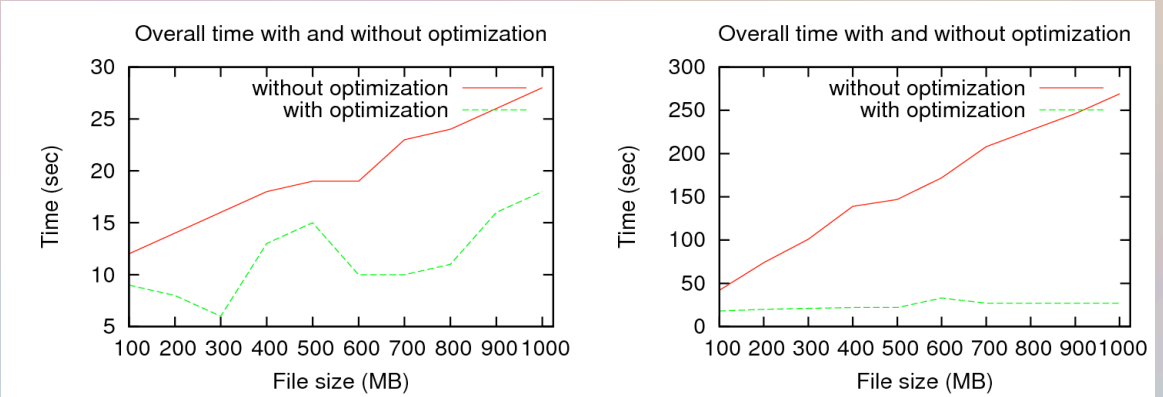




This measures the transfer time and job turn around time versus file size from Eric1 to oliver1 in LONI network .

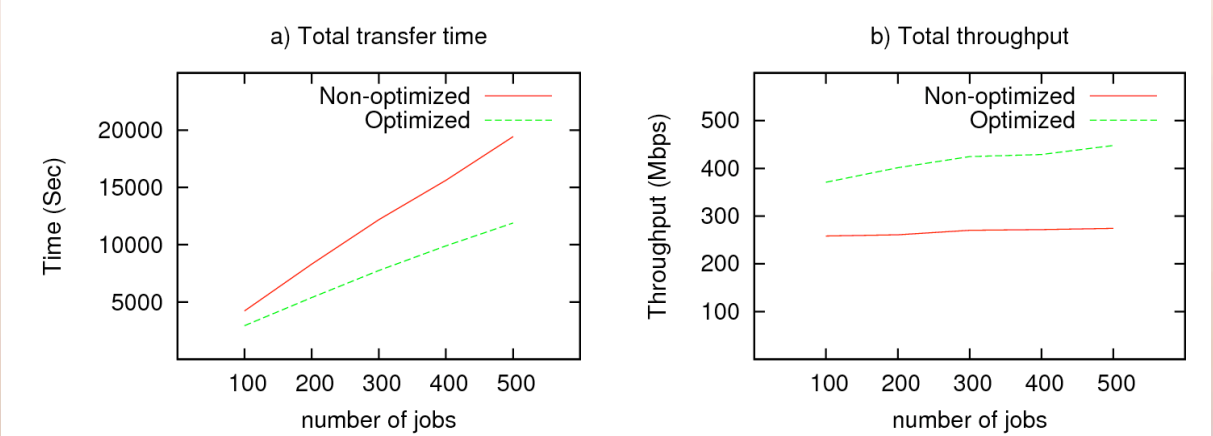


This measures the transfer time and job turn around time versus file size from louie1 to painter1 in LONI network .

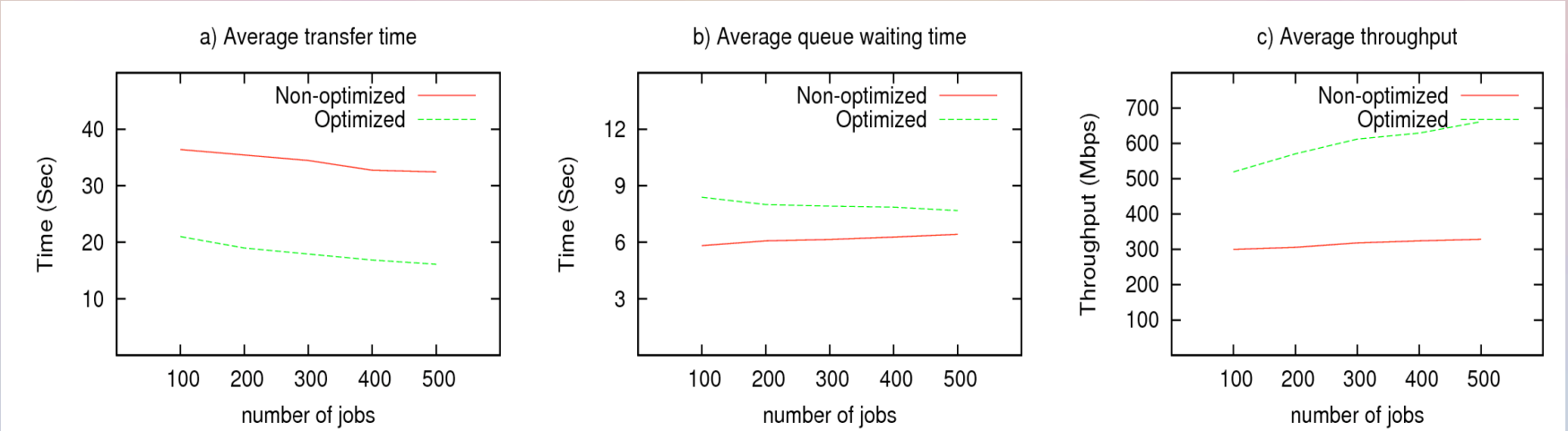


This measures the overall time versus file size from Eric1 to oliver1(left) and louie1 to painter1(right) in LONI network .

Total time and throughput of jobs submitted to Stork scheduler



Average transfer time, queue waiting time and throughput of jobs submitted to Stork scheduler



Thank You!

Questions?