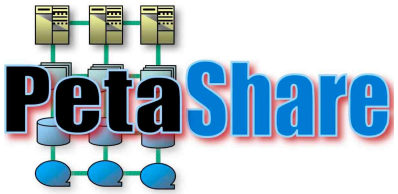
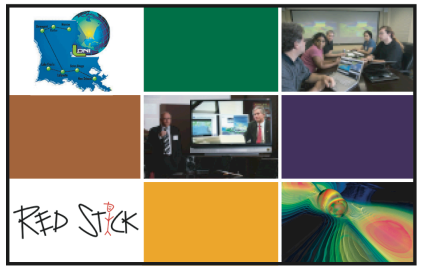


Dynamically Tuning Level of Parallelism in Wide Area Data Transfers

Esma Yildirim, Mehmet Balman, **Tevfik Kosar***

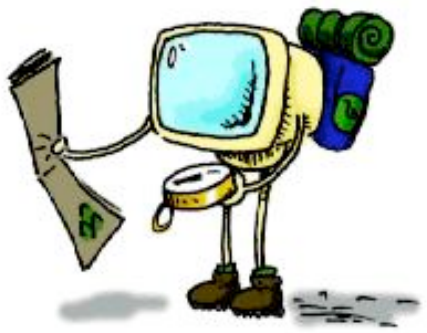
Center for Computation & Technology
Louisiana State University

June 24, 2008
DADC'08

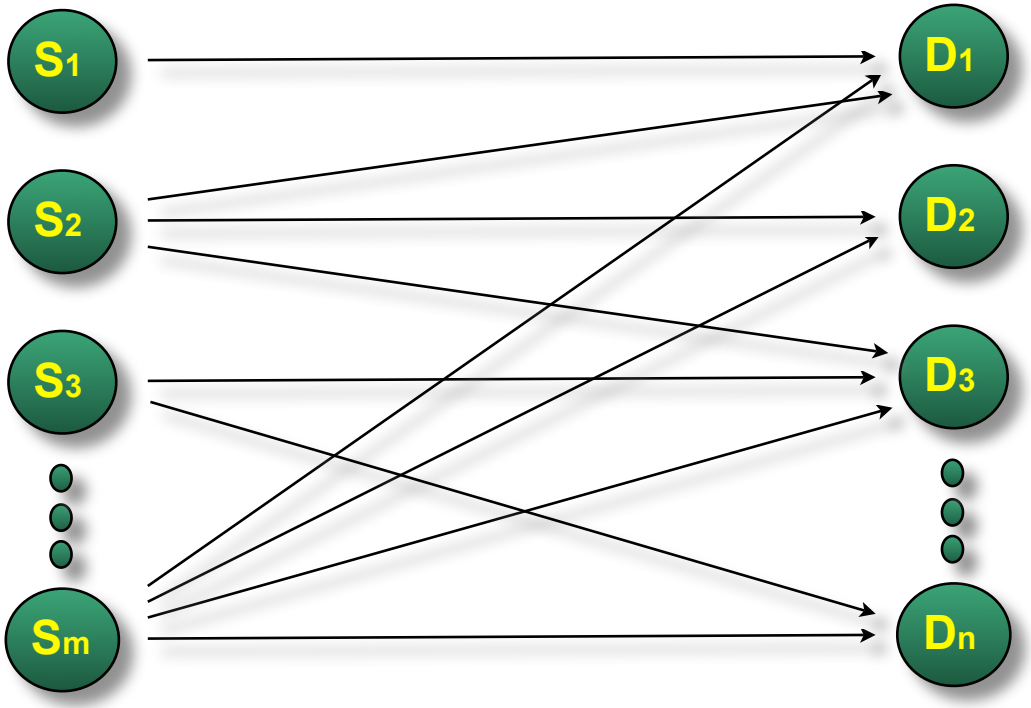


Roadmap

- Data Scheduling Problem
- Tuning Number of Parallel Streams
- Existing Models
- Proposed Models
- Results and Comparison
- Conclusion and Future Work



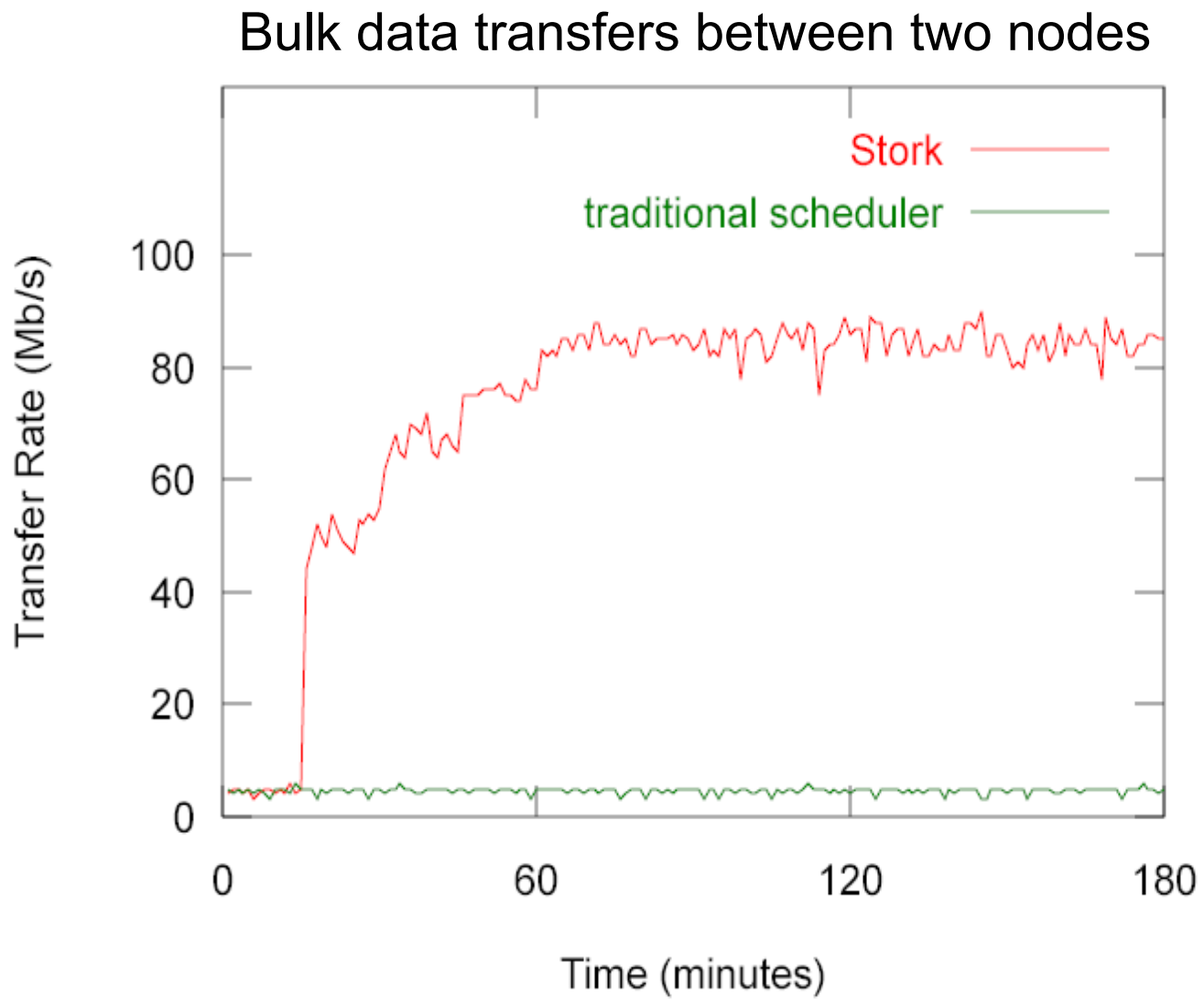
Data Scheduling Problem



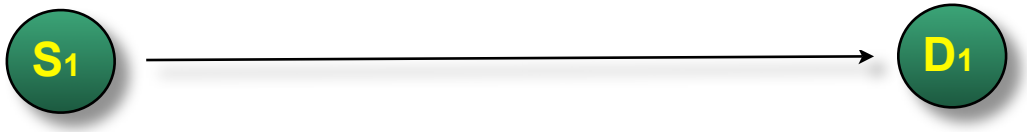
Transfer *k* files between *m* sources and *n* destinations, optimize by:

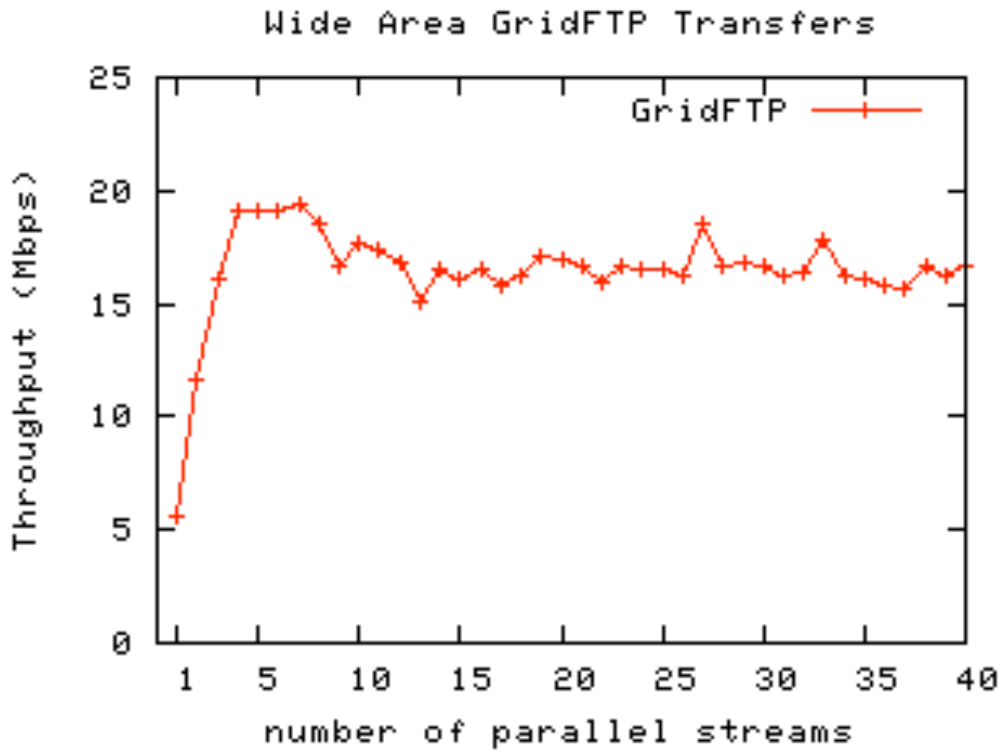
- Ordering requests (considering priority, file size, etc.)
- Throttling - deciding number of concurrent transfers (considering available target storage space, network capacity, etc.)
- Tuning protocol transfer parameters (considering current network condition)

Tuning Protocol Parameters

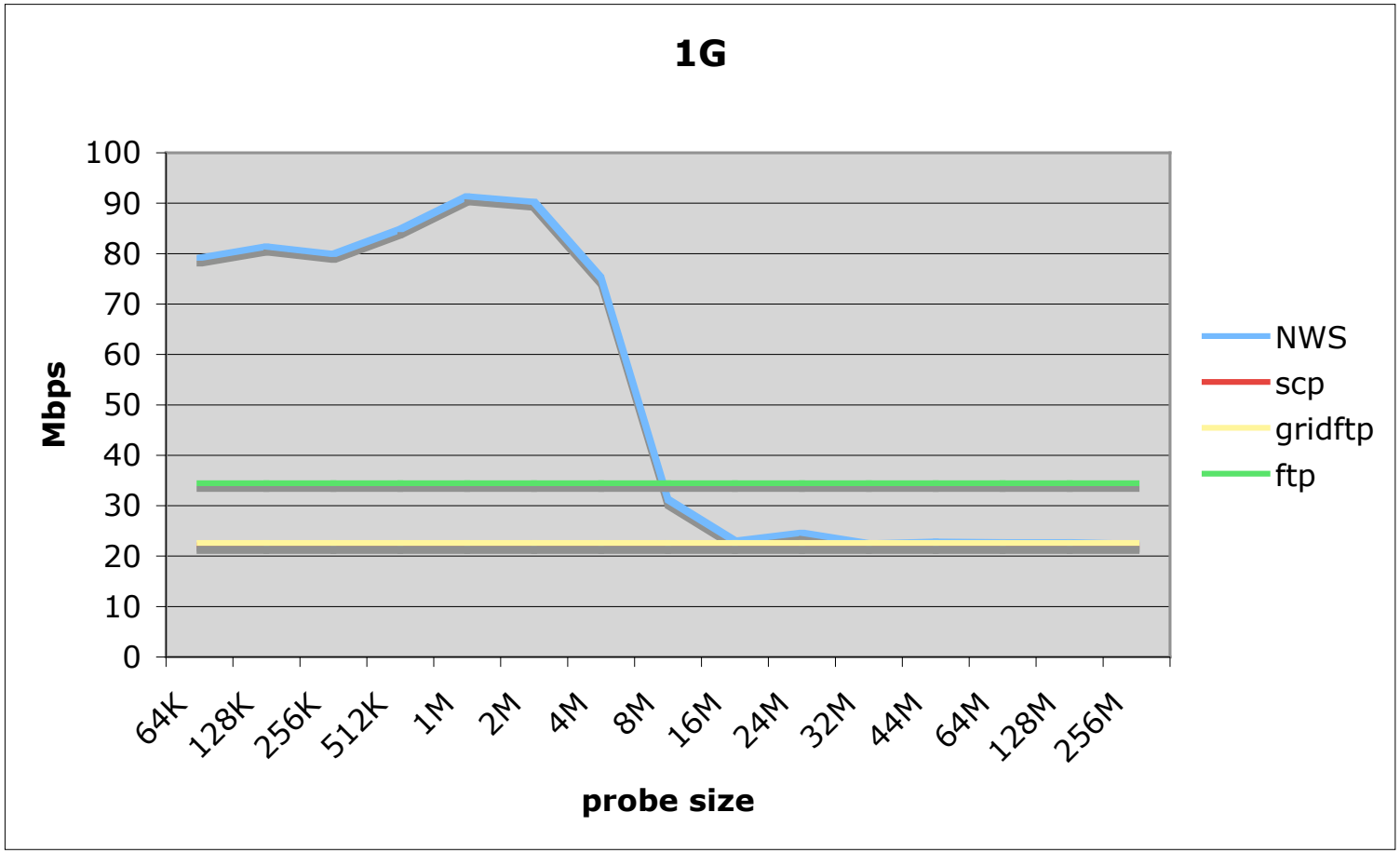


If No History Data?

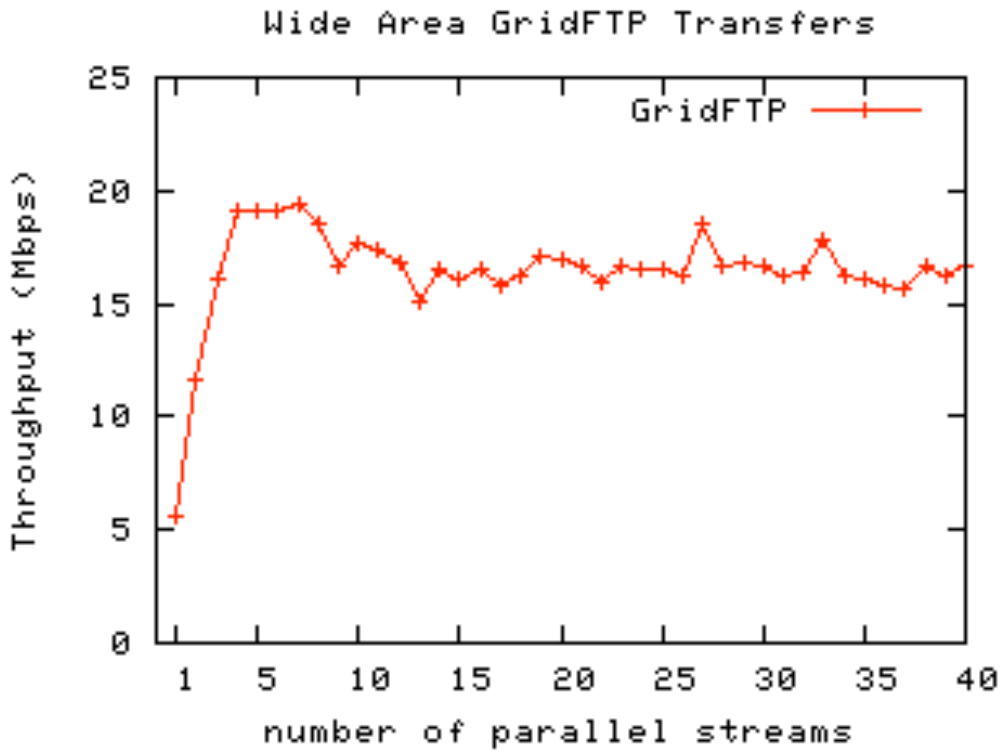




Use Predictions?



Can we predict this?



Hacker et al (2002)

- Theoretical calculation of throughput based on MSS, RTT and packet loss rate

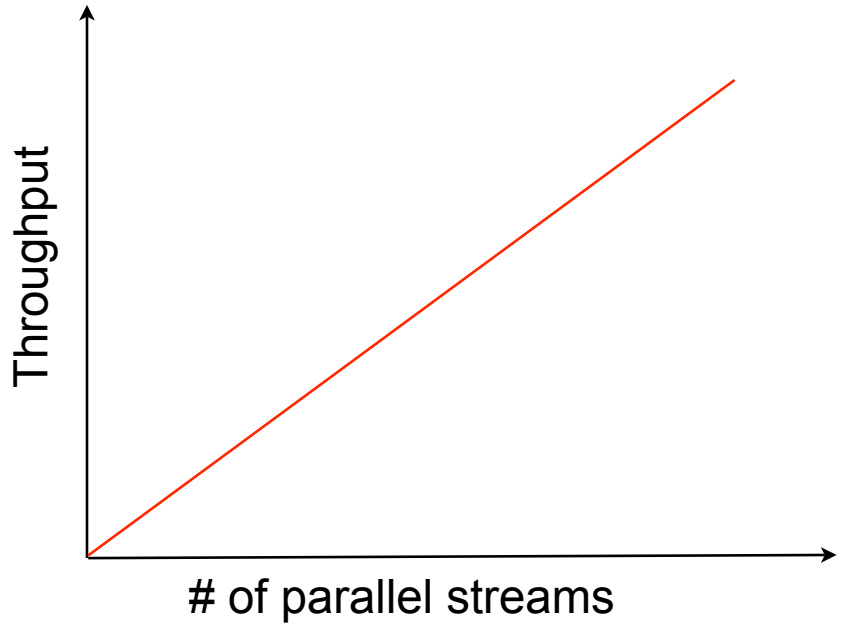
$$Th <= \frac{MSS}{RTT}\frac{c}{\sqrt{p}}$$

- An application opening n streams gains as much throughput as the total of n individual streams can get

$$Th_n <= \frac{MSS \times c}{RTT}\left(\frac{n}{\sqrt{p}}\right)$$

Hacker et al (2002)

- Throughput linearly increases until the point of congestion
- Packet loss is assumed stable
- Applicable only for noncongested networks



Dinda et al (2005)

• A relation is established between RTT, p and the number of streams *n*

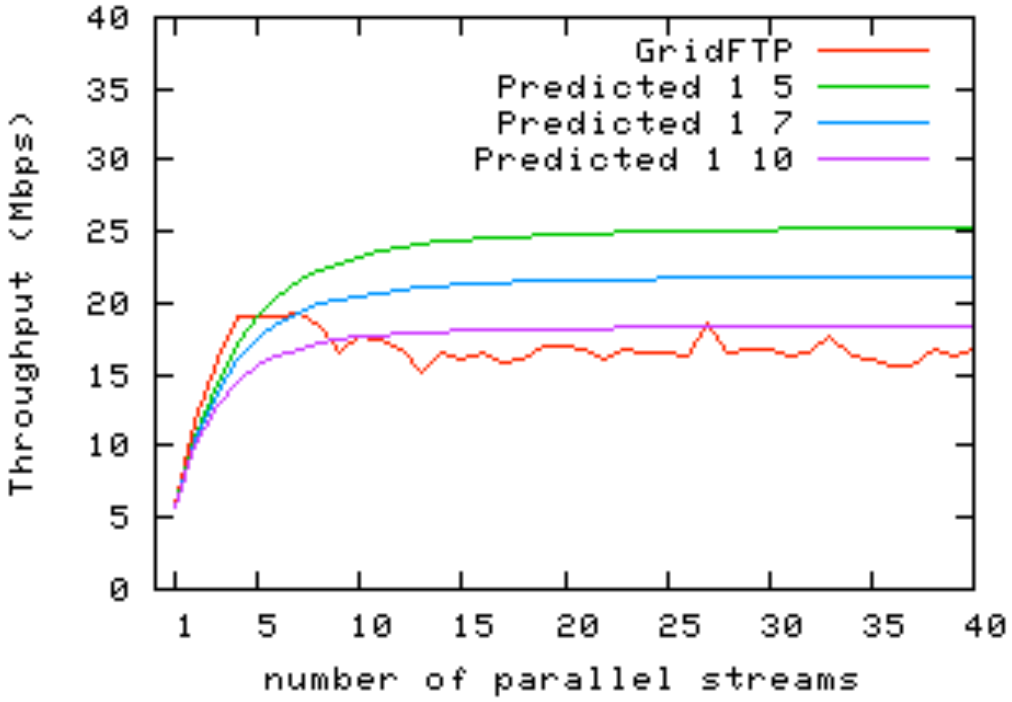
$$p'_n = p_n \frac{RTT_n^2}{c^2 MSS^2} = a'n^2 + b' \qquad Th_n = \frac{n}{\sqrt{p'_n}} = \frac{n}{\sqrt{a'n^2 + b'}}$$

• By solving a' and b' unknowns, the throughput of n streams could be found

• Only needs the measured throughput values of two different stream values

Dinda et al (2005)

- Throughput increases as n increases but never falls down
- Hard to find the optimal level of parallelism
- Two measurements can be taken with either a tool (Iperf, NWS) or from past transfers



1 - Modeling Packet Loss

- At the point of congestion packet loss starts to increase
- Lack of Hacker et al. Model: Unable to model the behavior of packet loss
- A similar method is used as in Dinda et al. Model
- Instead of predicting throughput we predict packet loss rate based on two measurements of different parallelism levels.

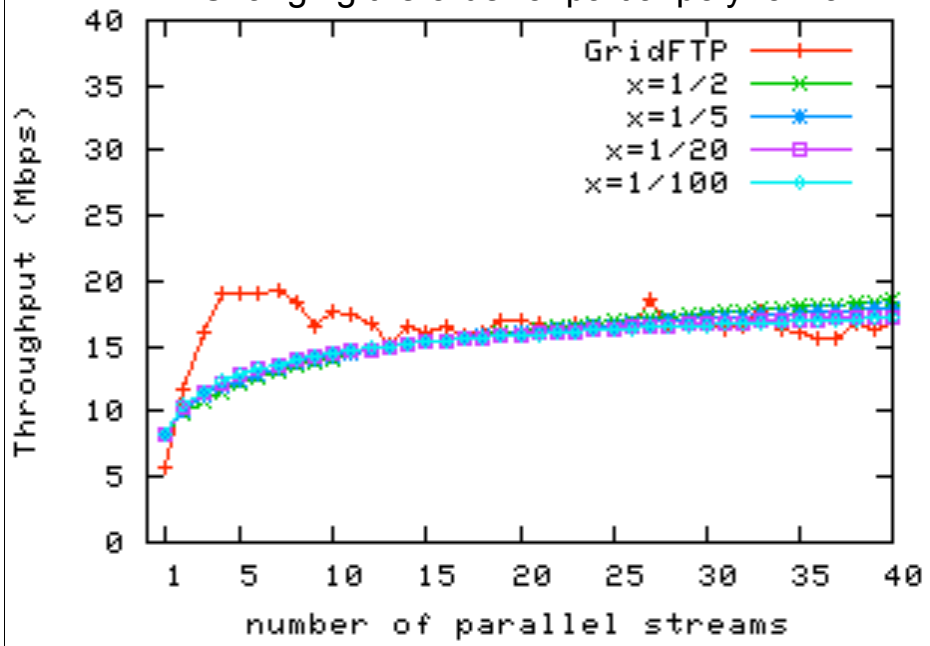
$$p_n = \frac{MSS^2 c^2 n^2}{RTT^2 Th_n^2}$$

$$Th'_n = \frac{RTT_n^2 Th_n^2}{MSS^2 c^2} = a' n^{1/x} + b'$$

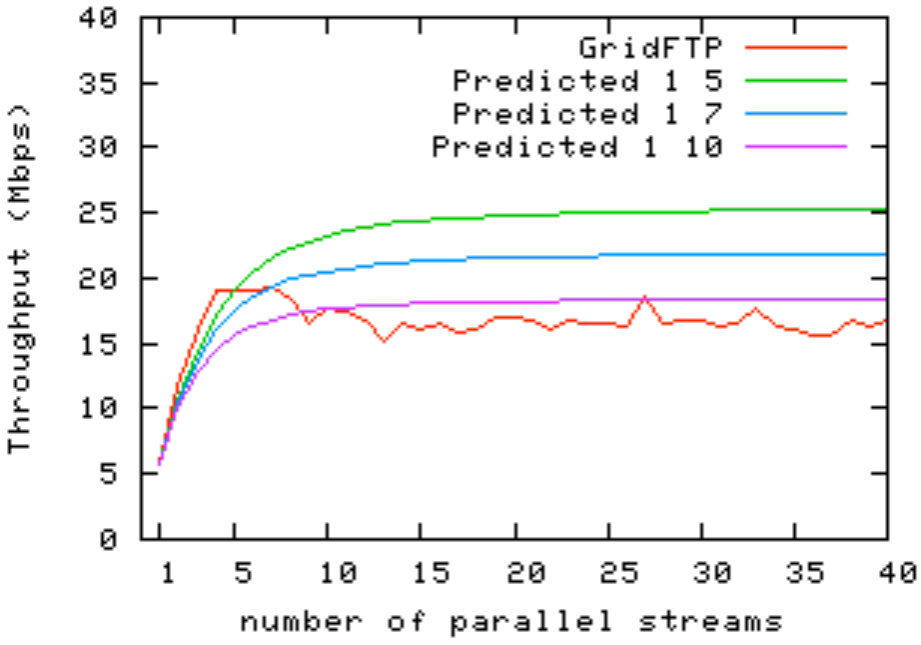
$$p_n = \frac{n^2}{Th'_n}$$

1 - Modeling Packet Loss

Changing the order of partial polynomial



Dinda et al Model



2 - Logarithmic Modeling

- We use an exponential function instead of a partial second order equation

$$p'_n = a' e^{b' n}$$

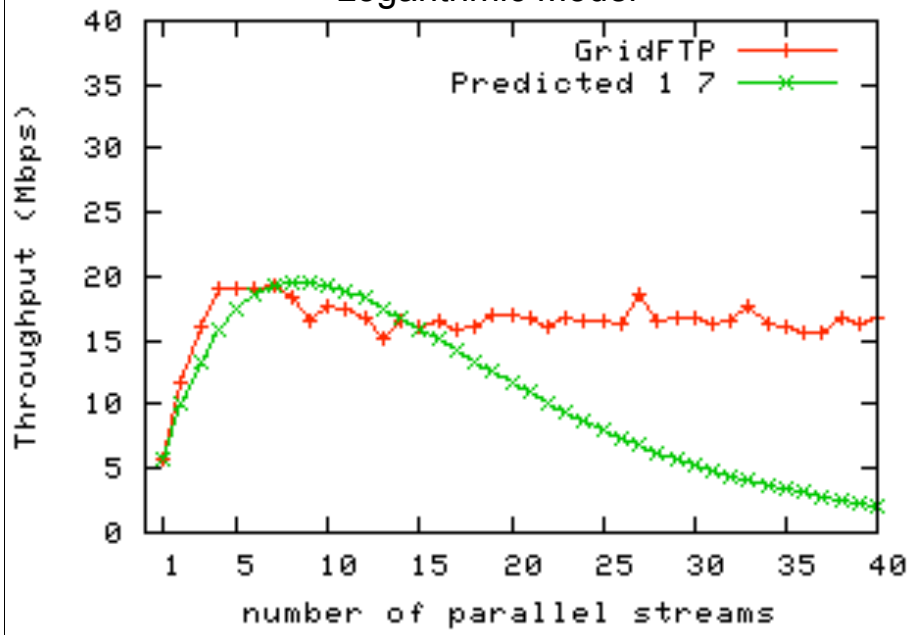
$$Th_n = \frac{n}{\sqrt{a' e^{b' n}}}$$

$$a' = \frac{n_1^2}{Th_{n_1}^2 \times e^{b n_1}}$$

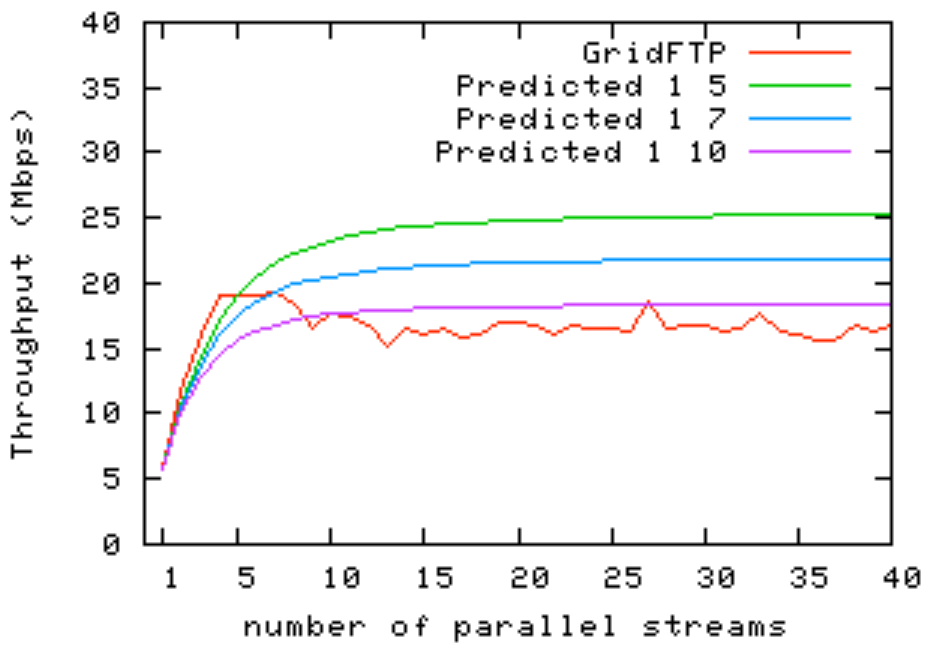
$$b' = \log_{e^{n_1 - n_2}} \frac{Th_{n_2}^2}{Th_{n_1}^2} \times \frac{n_1^2}{n_2^2}$$

2 - Logarithmic Modeling

Logarithmic Model



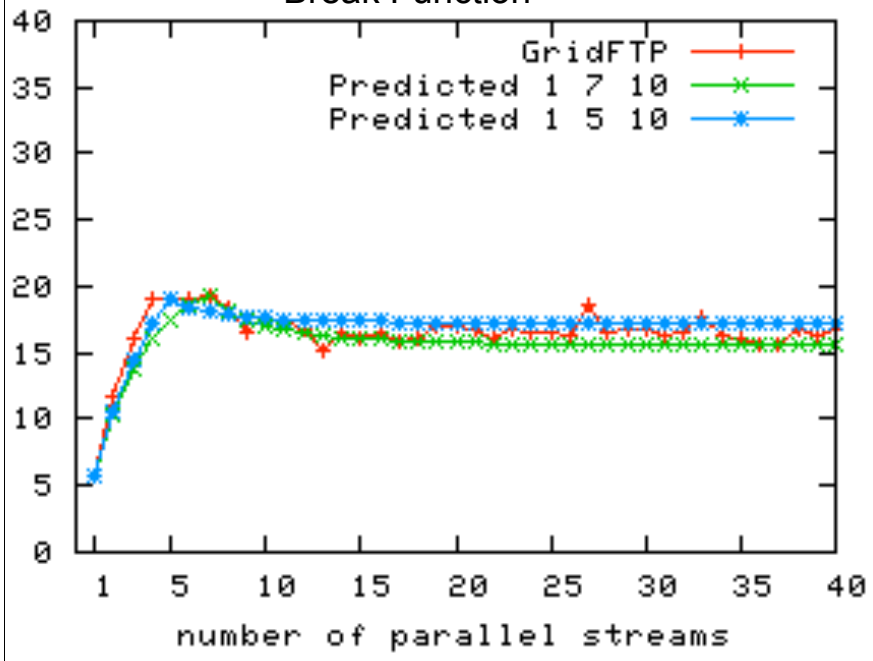
Dinda et al Model



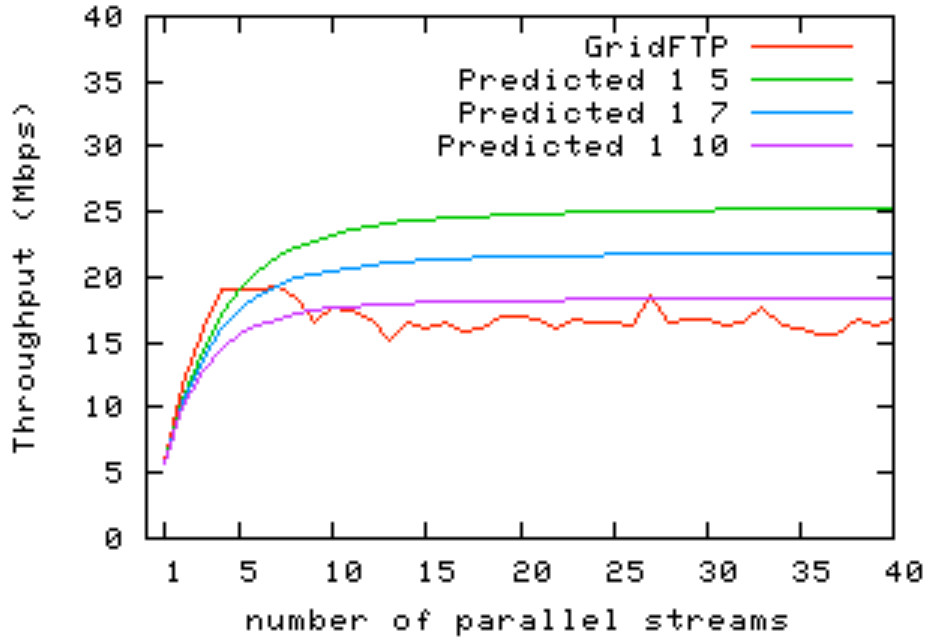
3 - Break Function

- Model the throughput curve as two separate functions
- Calculate *a'* and *b'* for increasing and decreasing part of the curve
- Based on n_1 and n_2
- Based on n_2 and n_3

Break Function



Dinda et al Model



4 - Newton's Method

•The order of the equation should be unknown and determined dynamically

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

•Three measurements are needed to solve the unknowns a', b' and c'

• It is difficult to solve as c' is the power in this equation

4 - Newton's Method

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

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

$$b' = \frac{n_1^2}{Th_{n_1}^2} - an_1^{c'}$$

- The value of a' and b' depends on c'
- How to solve c'?

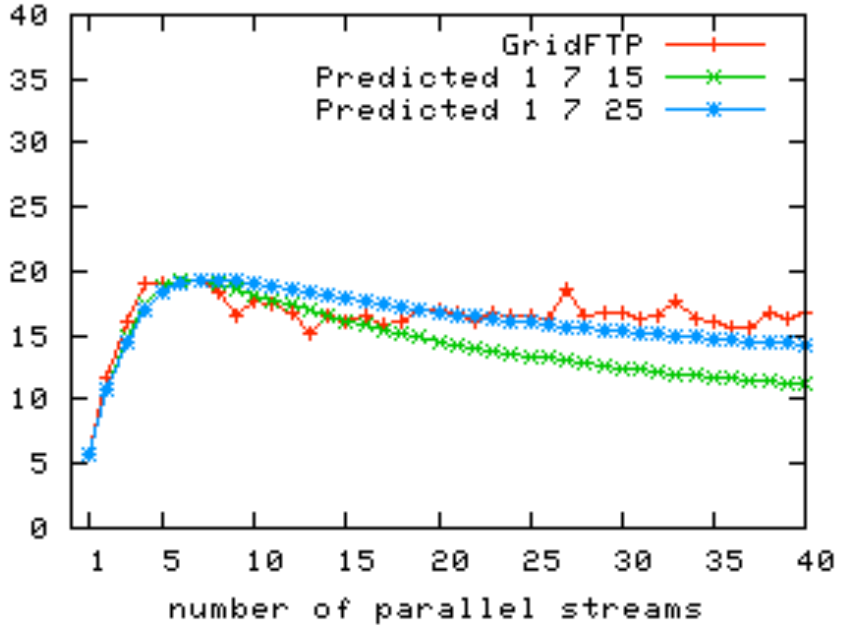
4 - Newton's Method

- We use a mathematical root finding method which gives approximate solutions

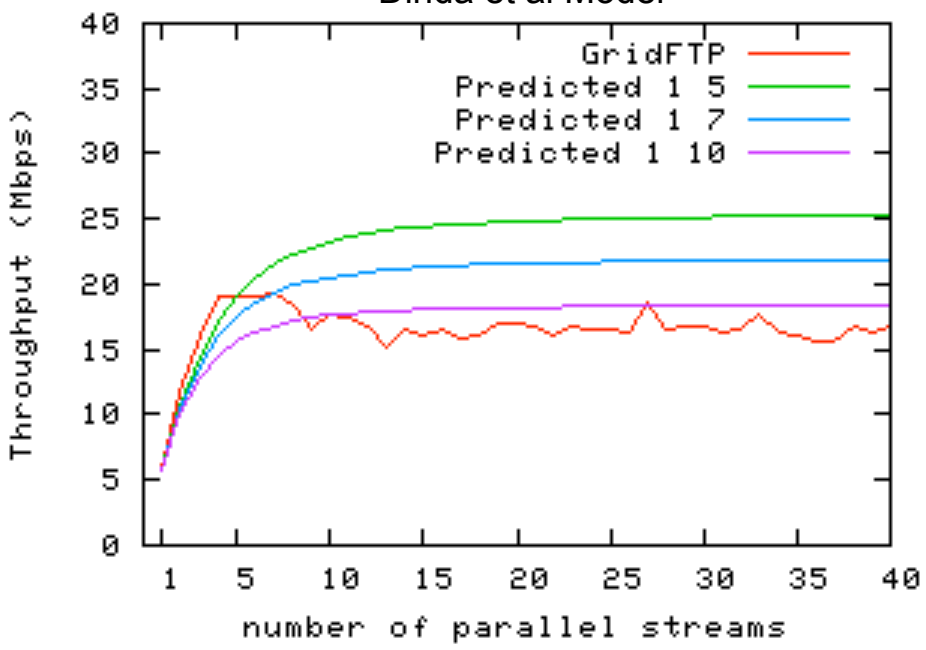
$$c'_{x+1} = c'_x - \frac{f(c'_x)}{f'(c'_x)}$$

- After x iterations a very approximate solution is found for c'

Newton's Method



Dinda et al Model



Comparison

Table 1: Distance between Actual and Predicted Throughput values up to 20 streams

Models	Distance
Dinda et al 1-7	283.0011927
Averaging 1-7-15	90.33315451
Break Function 1-7-15	25.97119405
Logarithmic 1-7	107.3807982
Newton's Method 1-7-25	44.64025274

$$Distance_n = \sum_{i=1}^{n} (p_i - a_i)^2$$

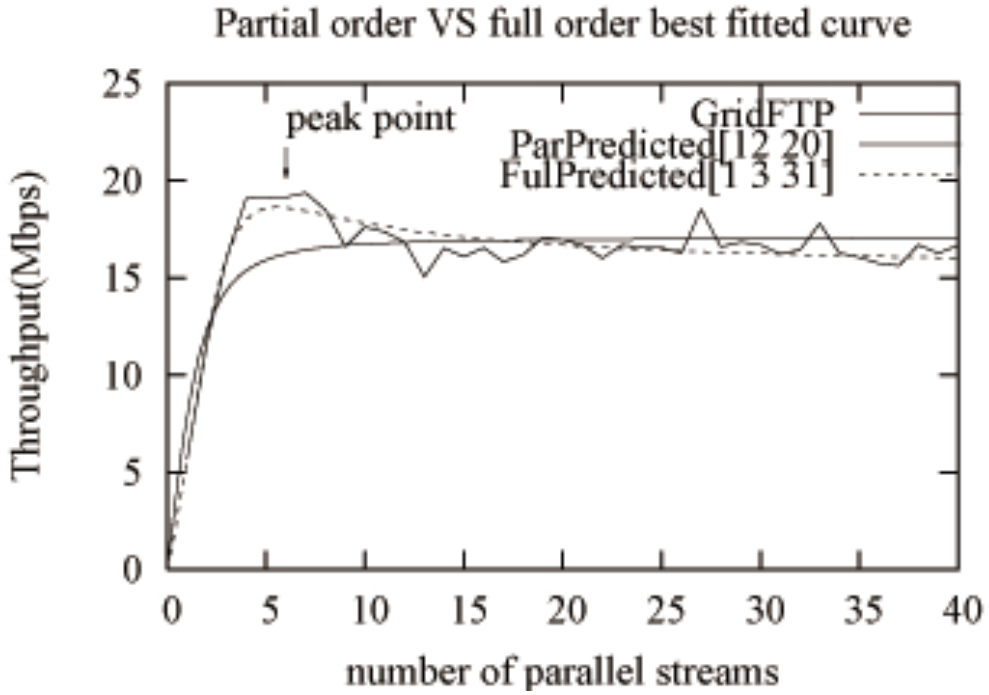
Table 2: Average distance between actual and predicted optimal number of streams

Models	Distance
Dinda et al 1-4..12	4.111
Averaging 1-4..12-15	2.444
Break Function 1-4..12-15	2.333
Logarithmic 1-4..12	2.333
Newton's Method 1-4..12-25	1.000

5 - Full Second Order

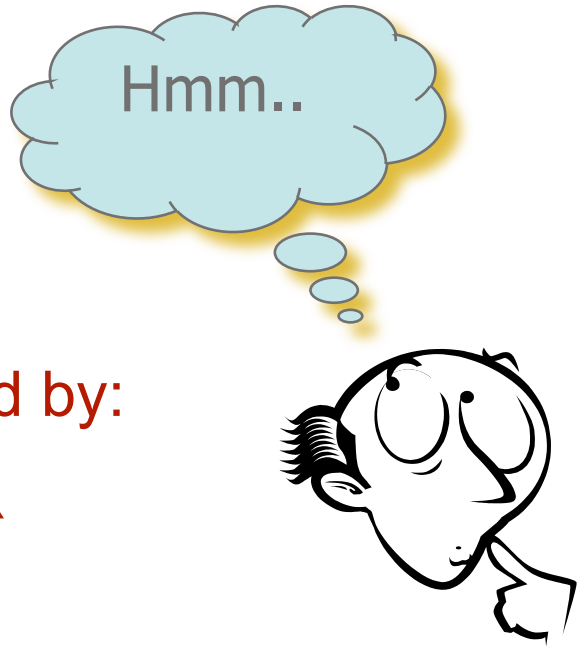
- Instead of a partial second order polynomial, use a full second order polynomial:

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



Conclusions

- Parallel streams increases the total achievable throughput for an application
- Opening too many connections reaches the network to a point of congestion
- Further increasing the number of streams results in a decrease in total throughput
- The optimal number of streams could be predicted with very little information
- Our proposed models could predict this number with a very close approximation



This work has been sponsored by:

NSF and LA BoR

For more information

Stork: http://www.storkproject.org

PetaShare:http://www.petashare.org