Clustering and Prefetching techniques for Network-based Storage Systems

By
Dhawal N. Thakker
Dr. Glenford Mapp
Dr. Orhan Gemikonakli

Networking Research Group

Middlesex University

Streaming applications

- Increased use of YouTube, BBC-IPlayer applications
- Sequential access
- Prefetching
  - Can Clustering help to reduce cost?
- Demand misses

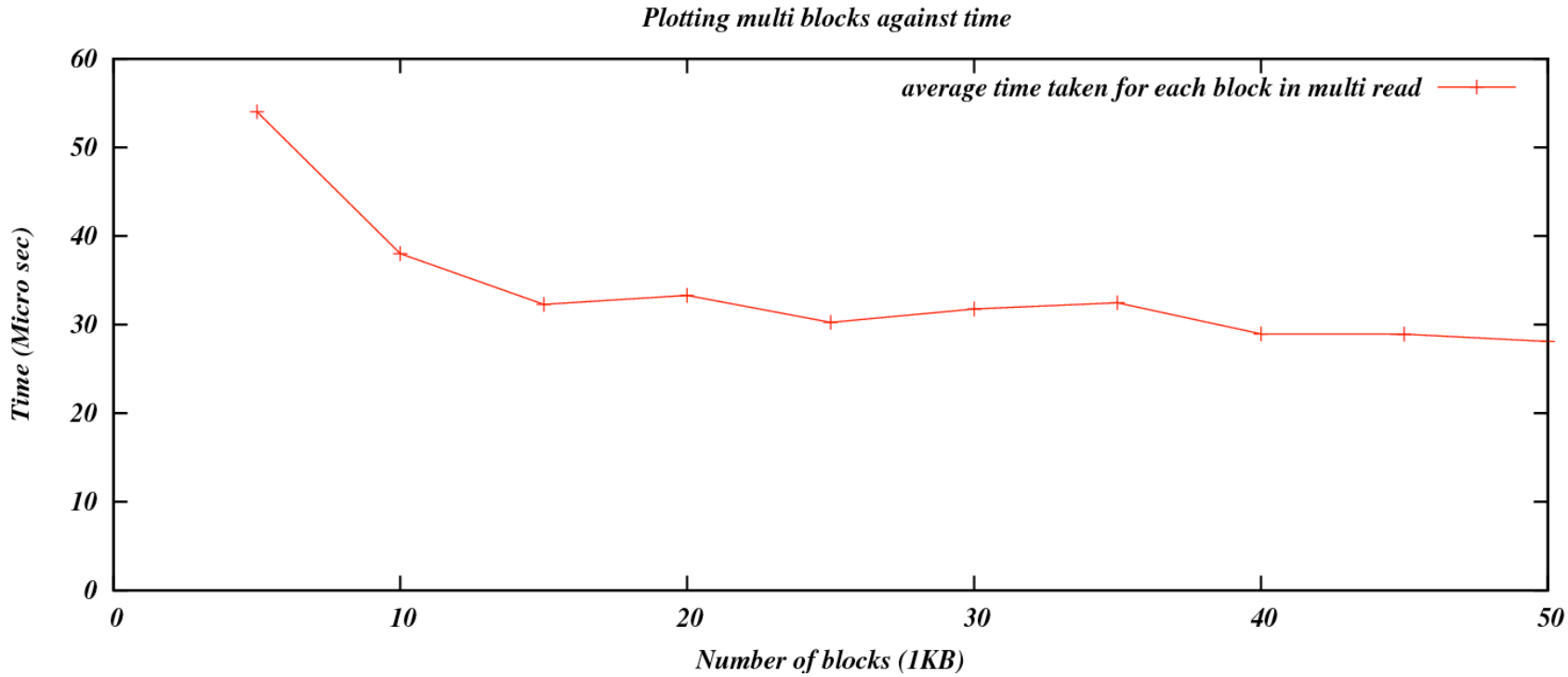
Clustering in the context of the networks

- What is Clustering?
- Why Clustering over the network?
- Network software not fast enough
- Aggregates reads and writes
- Better use of network bandwidth
- Prefetching can benefit.

Concept of a Network Memory Server

- Provides storage over the network
- Runs over Fast Commodity Hardware
- Block-based system, can request several blocks at a time
- Emulates the commercial environment for streaming applications
  - such as video-on-demand

Benefits of Clustering



- Time to Fetch p blocks = L + Cp

Prefetching Over a network: early experience with CTIP [D. Rochberg et al.]

- Showed performance improvement by reducing execution time (nearly 30%)
- Using NFS was the main drawback, no clustering
- Increased CPU cost while fetching one block at a time

Proposed Work

- Exploit clustering and prefetching over the network using NMS
- To guarantee the quality of service:
  - Streaming applications (once they are started) with no jitter
  - Demand requests, in reasonable time

Approach

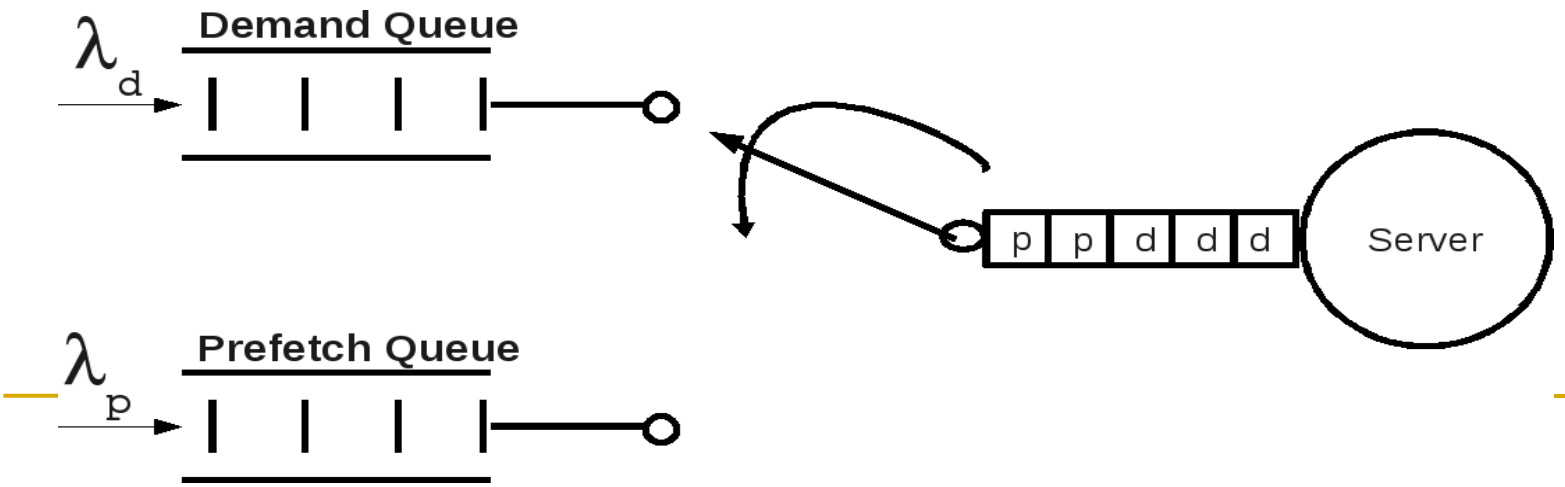
- T
  o allows streaming applications to run without jitter:
  - Time to fetch < Time to process
    - L + Cp < Tcpu * p
- Average waiting time experienced to satisfy Deman
  d
  misses < the average waiting time on disk (Tdisk)
    - L + (d * C) + Twait < Tdisk

Conservative Prefetching: PonD

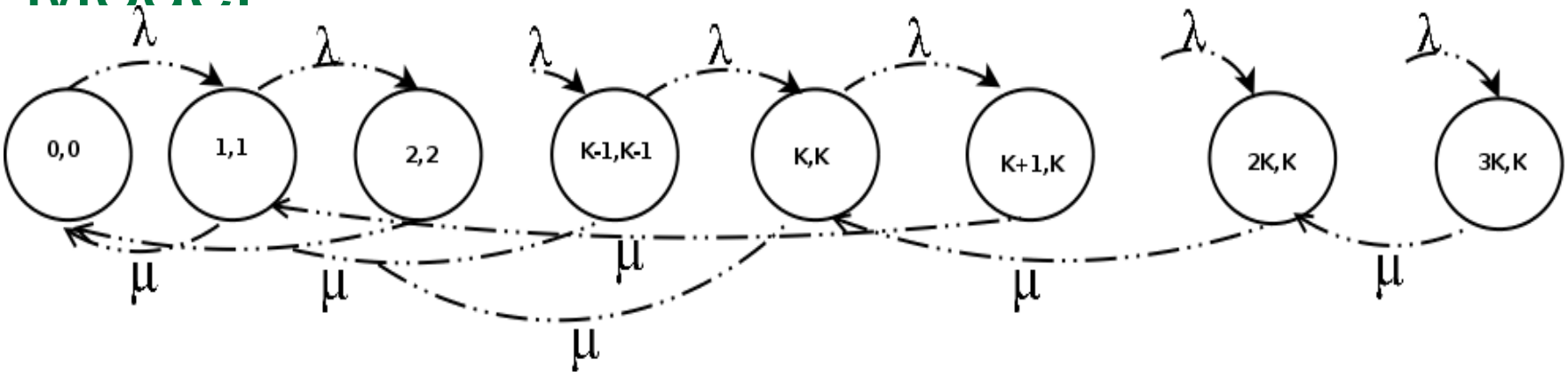
- First adopt the conservative approach taken by Pei Cao work
  - Prefetch when there is a demand miss i.e Prefetching on Demand (PonD)
- Therefore, using PonD:
  - L + (p+d) * C < (Tcpu * p)
- For demand misses:
  - Time to fetch + Twait < Tdisk
    - L + (p+d) * C + Twait < Tdisk

Towards an Analytical model

- Can control the prefetch rates for streaming applications
- Demand misses are totally random
- Need to analyse average waiting time on the demand queue

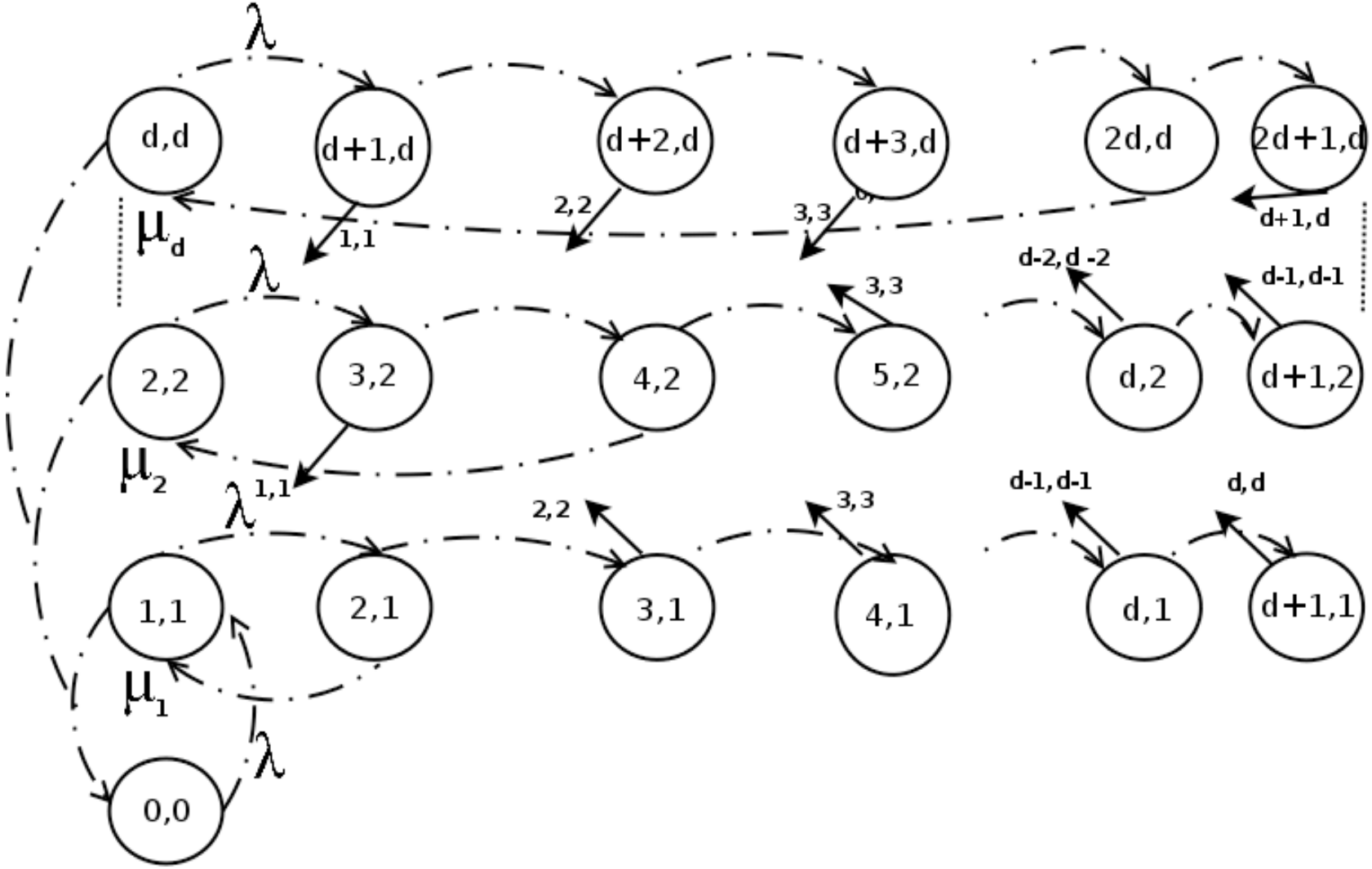


Standard Solution: Partial Bulk Service Model

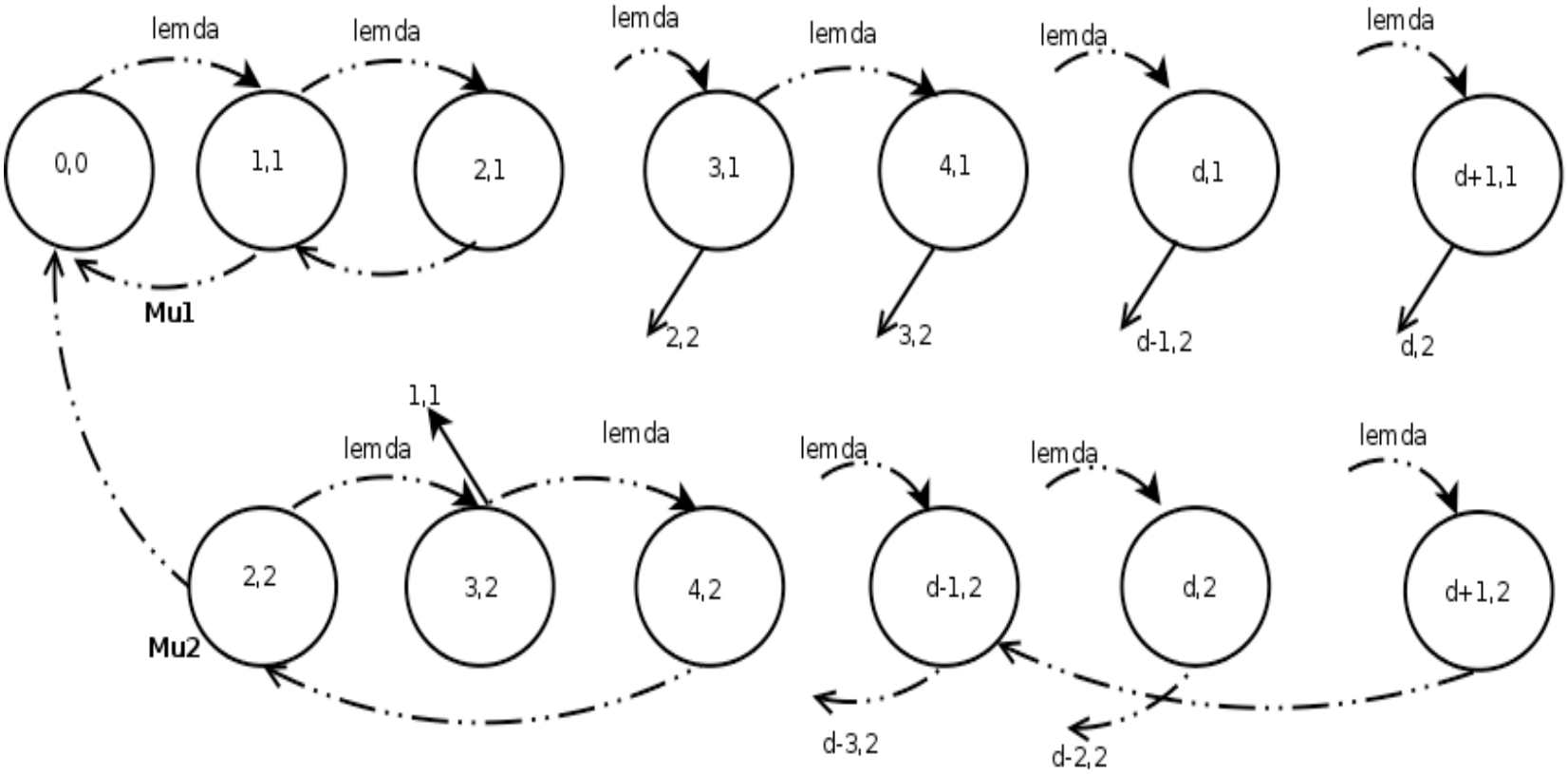


- Accurate at very high loads, but it is not suitable for low loads
  - Exhaustive-limited while our scenario is gate-limited
- Propose a new model

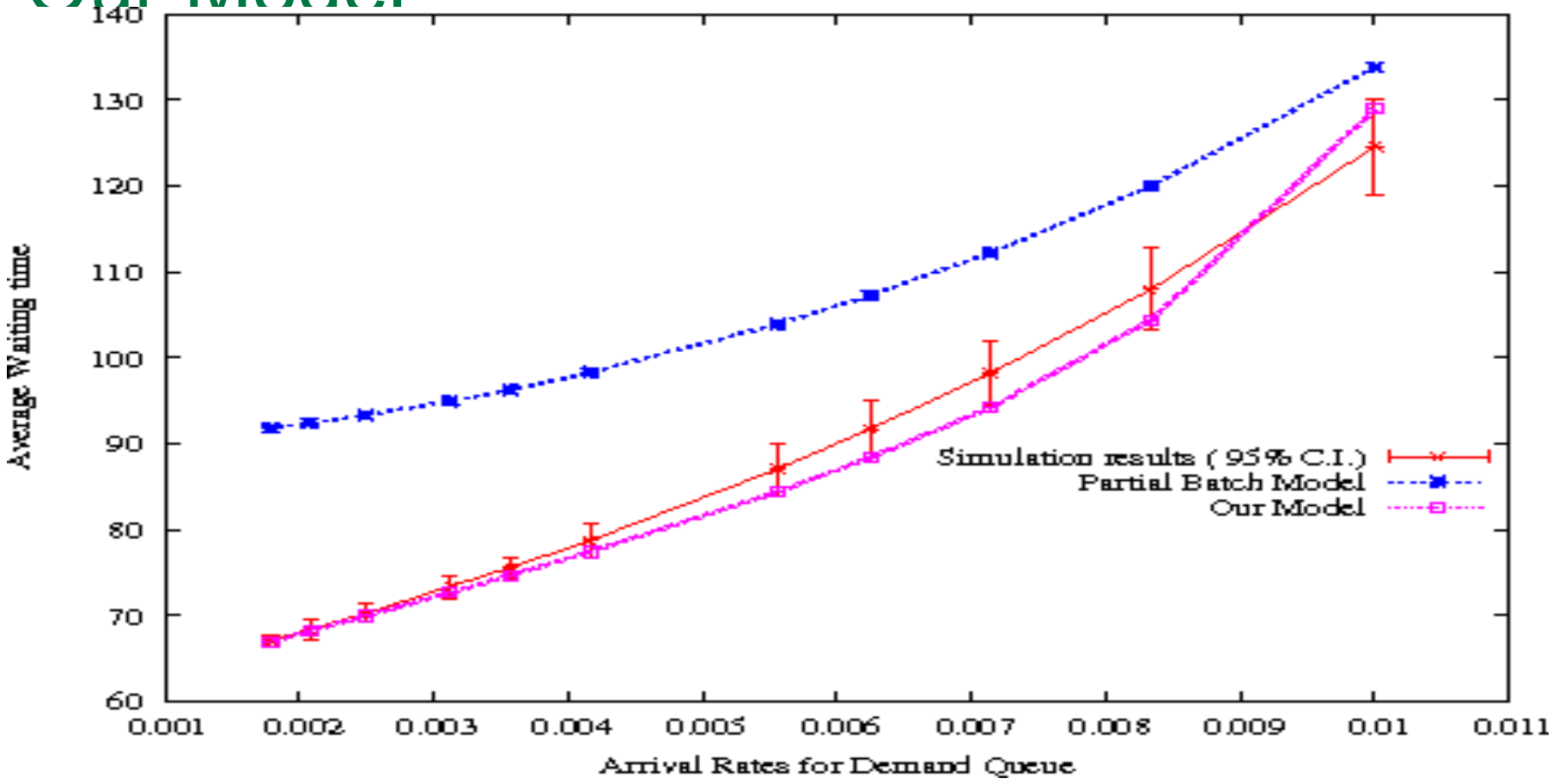
Our Model


λ
d,d
d+1,d
d+2,d
d+3,d
2d,d
2d+1,d
μd
1,1
2,2
3,3
d+1,d
λ
d-2,d -2
d-1,d-1
3,3
2,2
3,2
4,2
5,2
d,2
d+1,2
μ2
1,1
λ
d-1,d-1
d,d
2,2
3,3
1,1
2,1
3,1
4,1
d,1
d+1,1
μ1
λ
0,0

Model for d =2

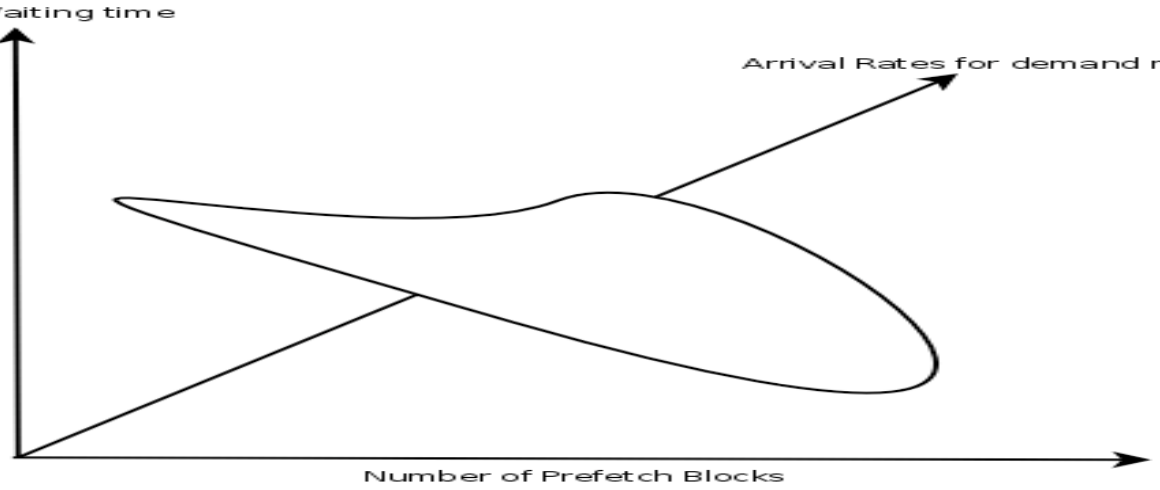

lemda
lemda
lemda
lemda
lemda
lemda
0,0
1,1
2,1
3,1
4,1
d,1
d+1,1
Mu1
2,2
3,2
d-1,2
d,2
1,1
lemda
lemda
lemda
lemda
lemda
2,2
3,2
4,2
d-1,2
d,2
d+1,2
Mu2
d-3,2
d-2,2

Results: Varifying results from simulation with Bulk service model and Our Model



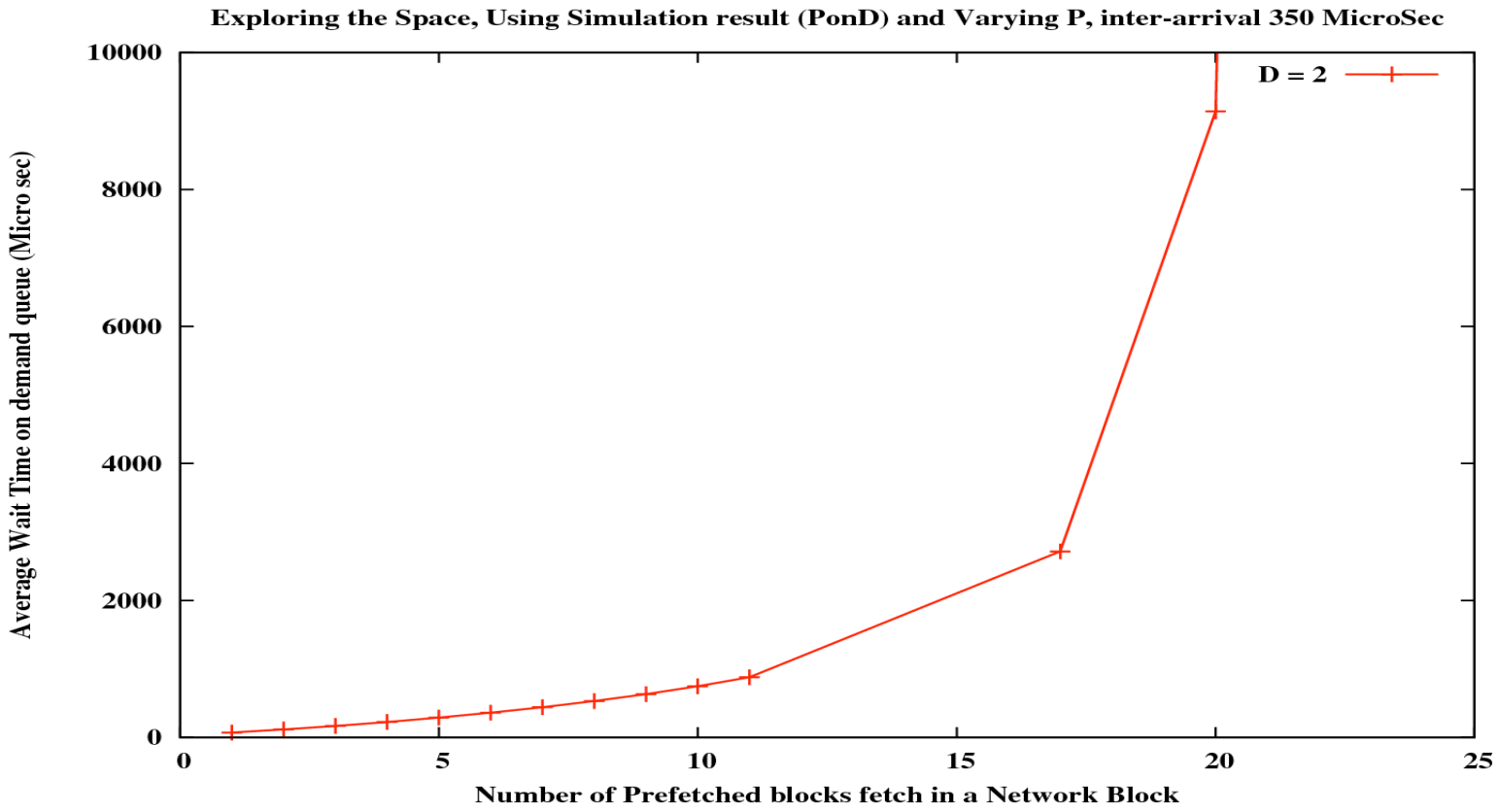
Defining an Operational Space

- Define an operational space consisting of three axes

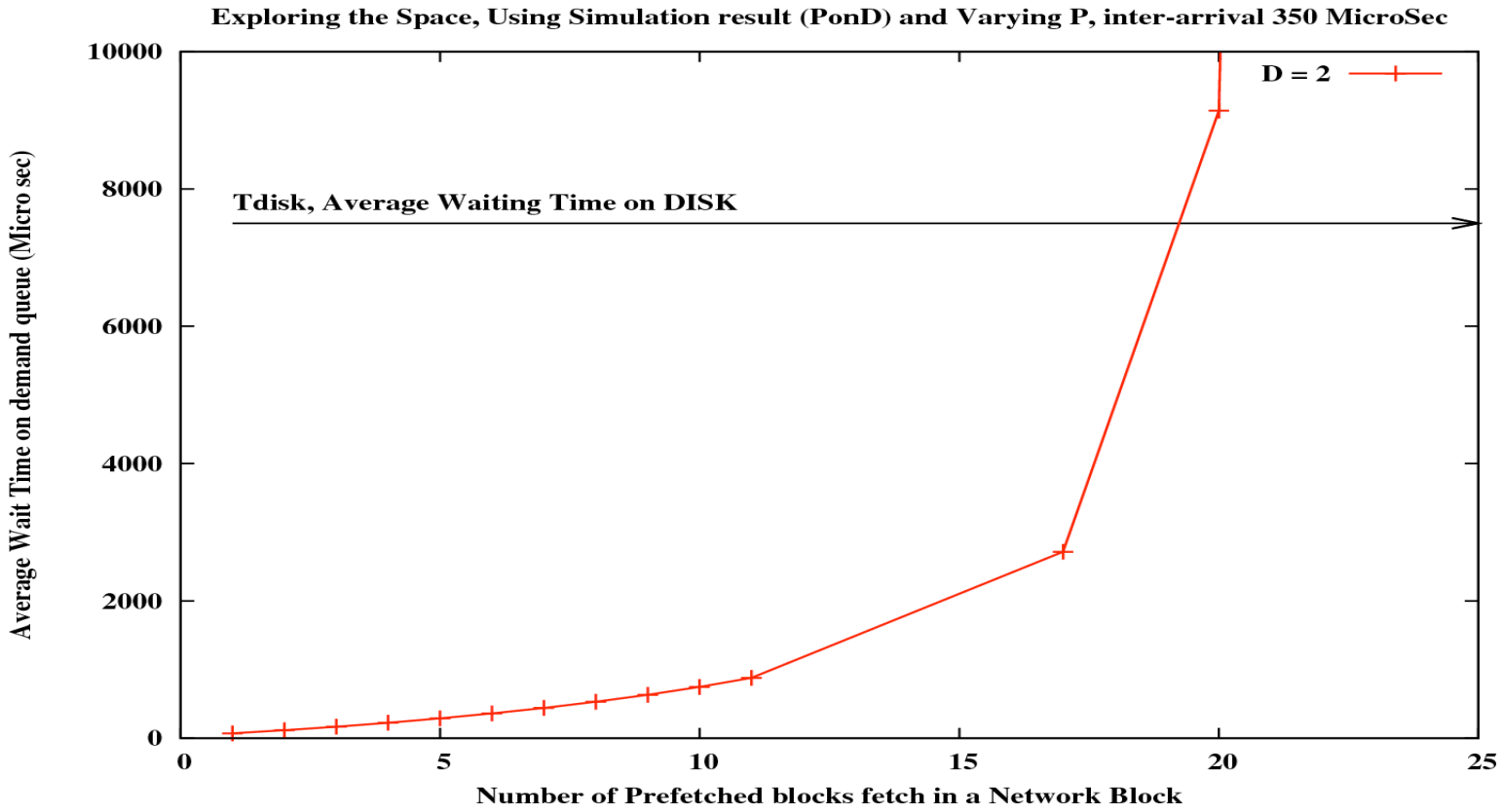


- Tavg, demand misses < Tdisk
- L + (p+d) C < P* Tcpu

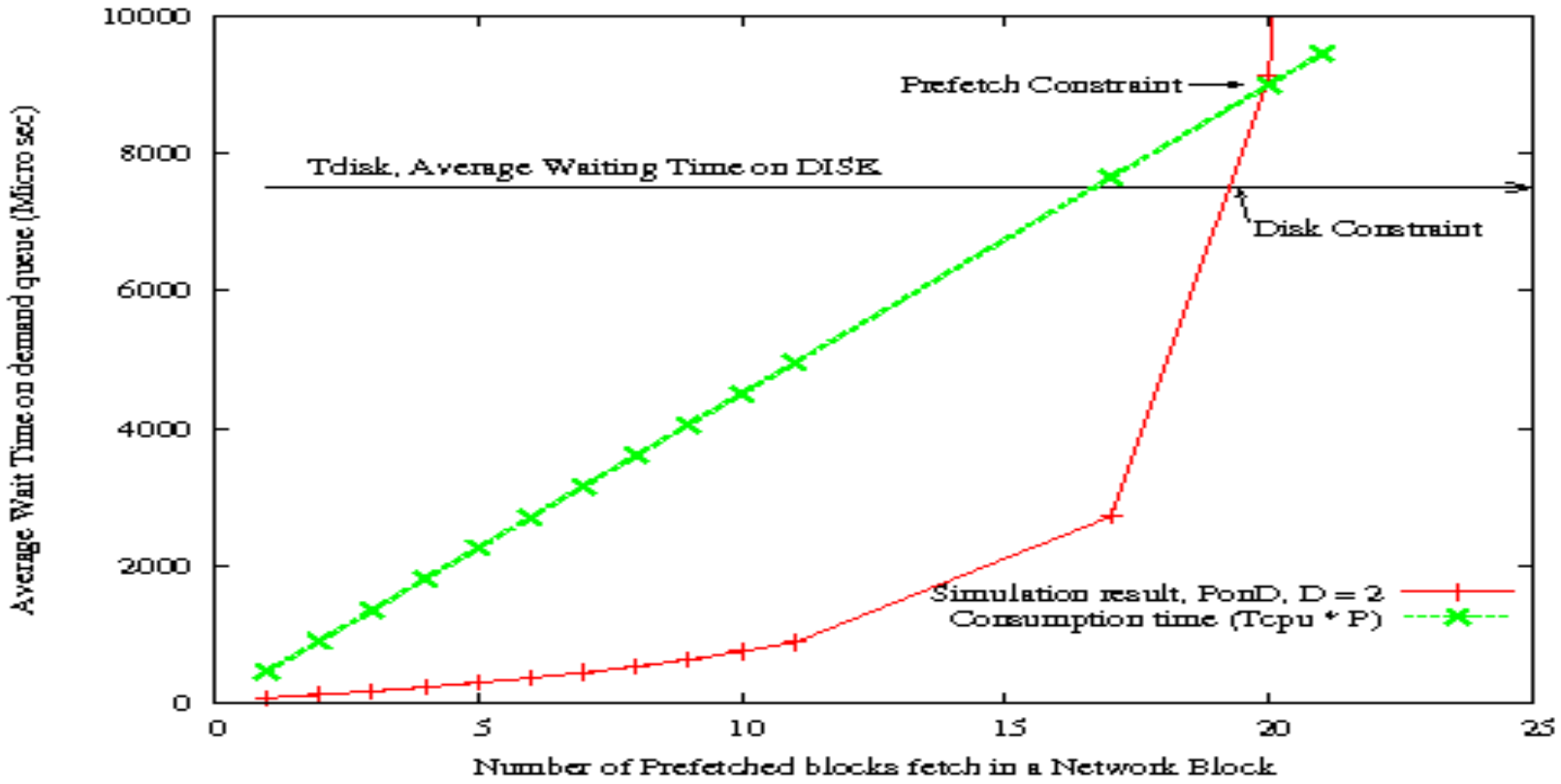
Exploring the space for a given demand miss inter arrival time, 350 microsec



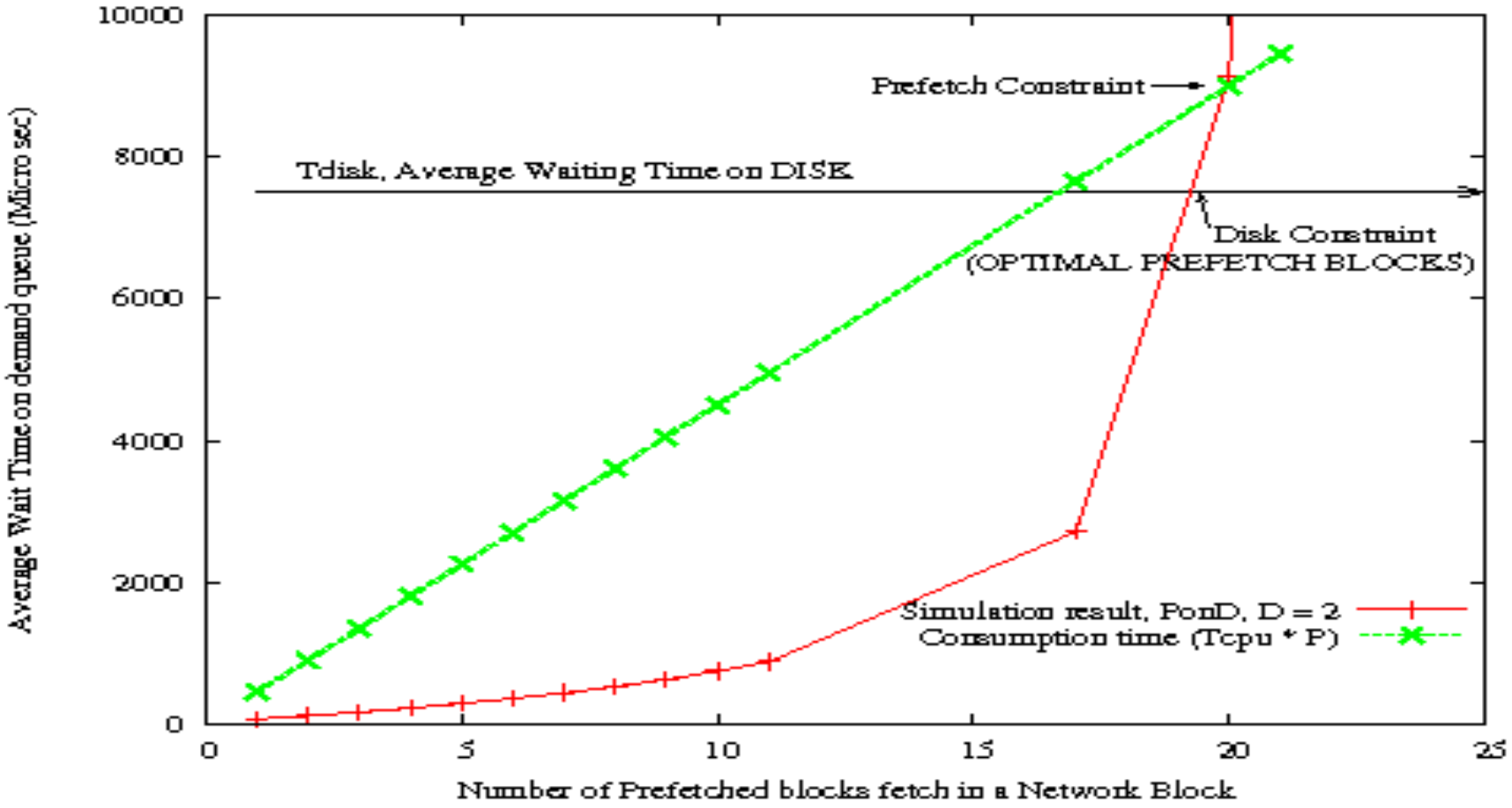
Exploring the Space, for a given demand miss inter arrival time, 350 microsec



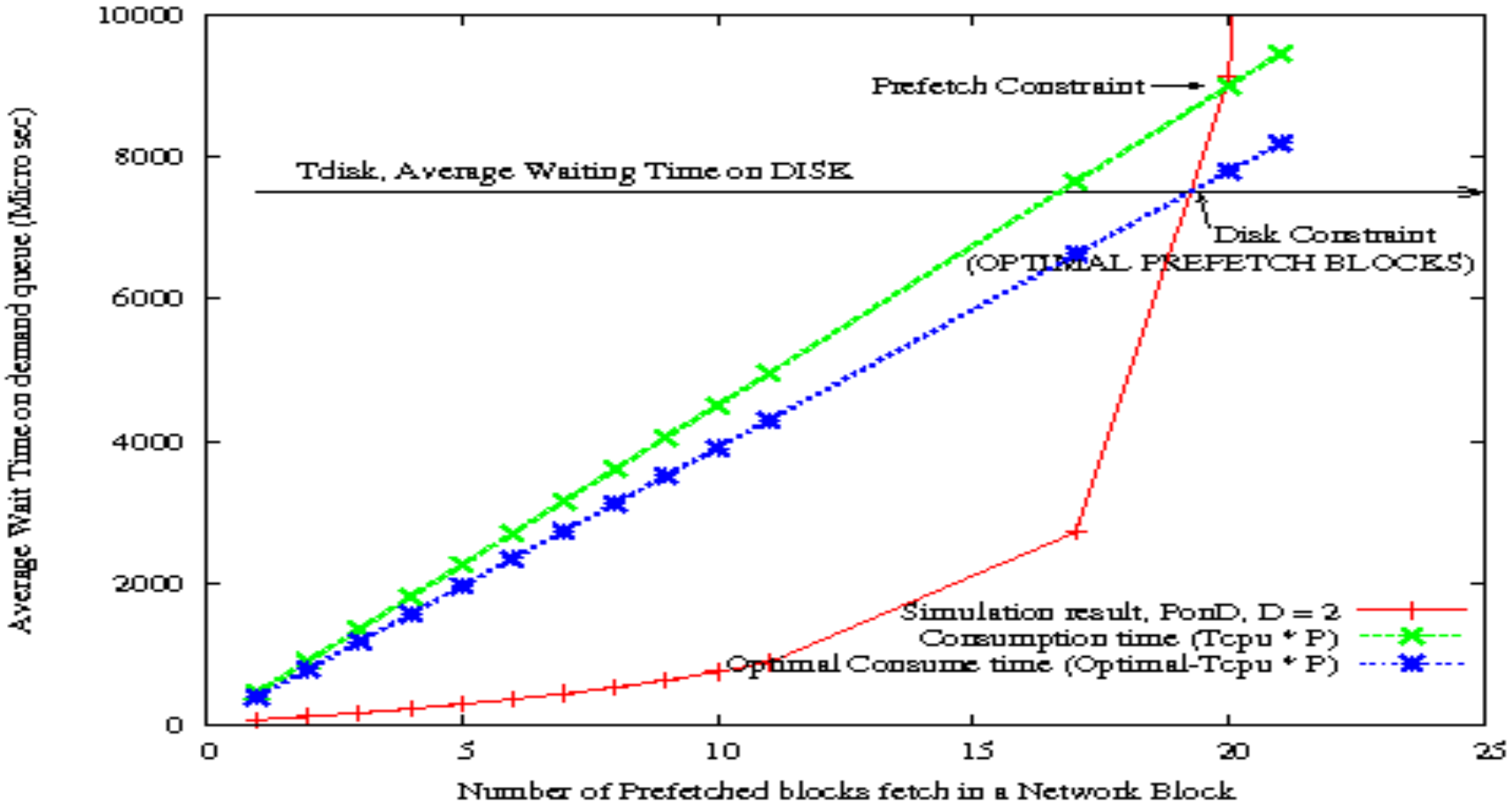
Exploring the Space, for a given demand miss inter arrival time, 350microsec



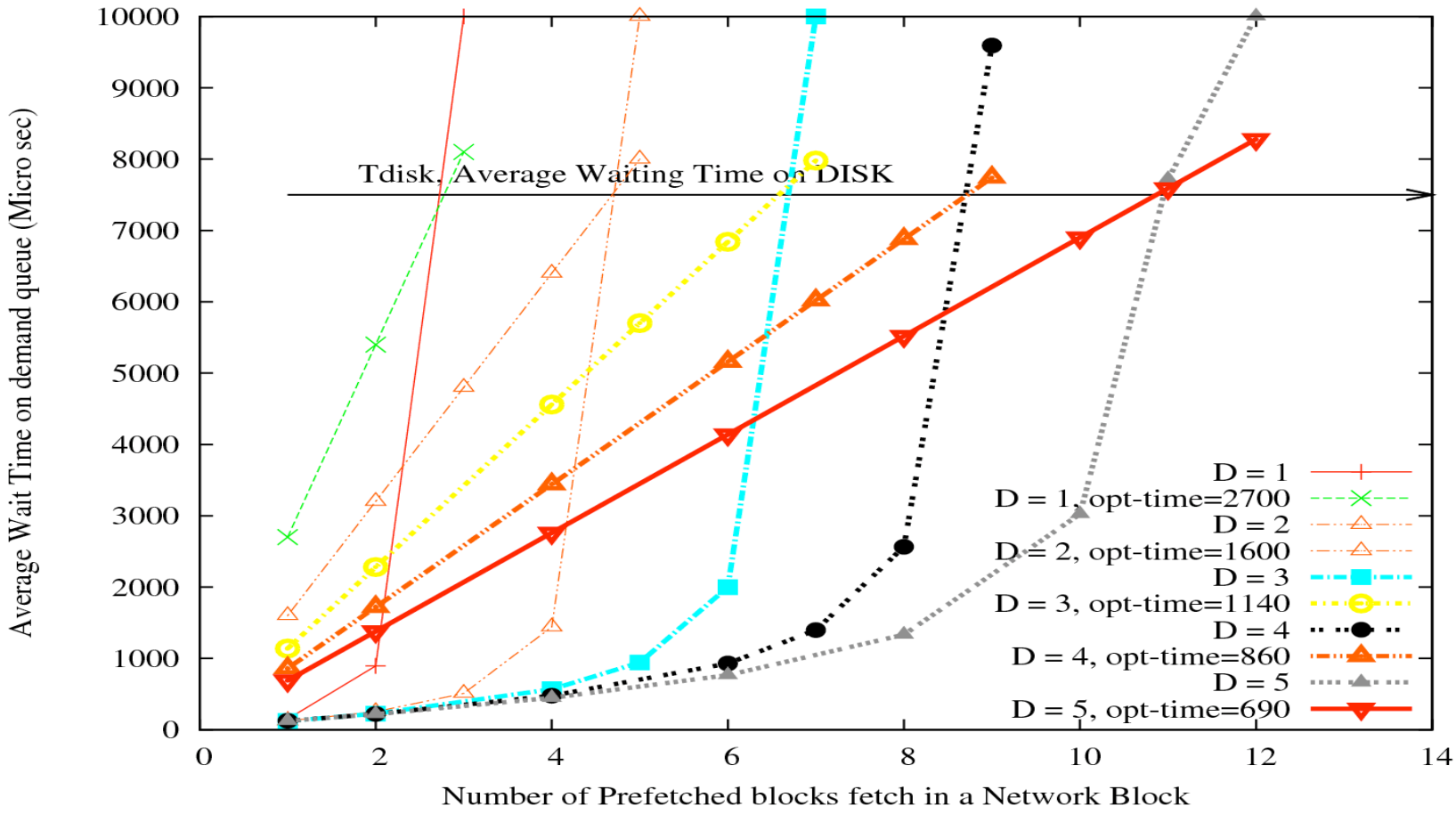
Exploring the Space, for a given demand miss inter arrival time, 350 microsec



Exploring the Space, for a given demand miss inter arrival time, 350 microsec

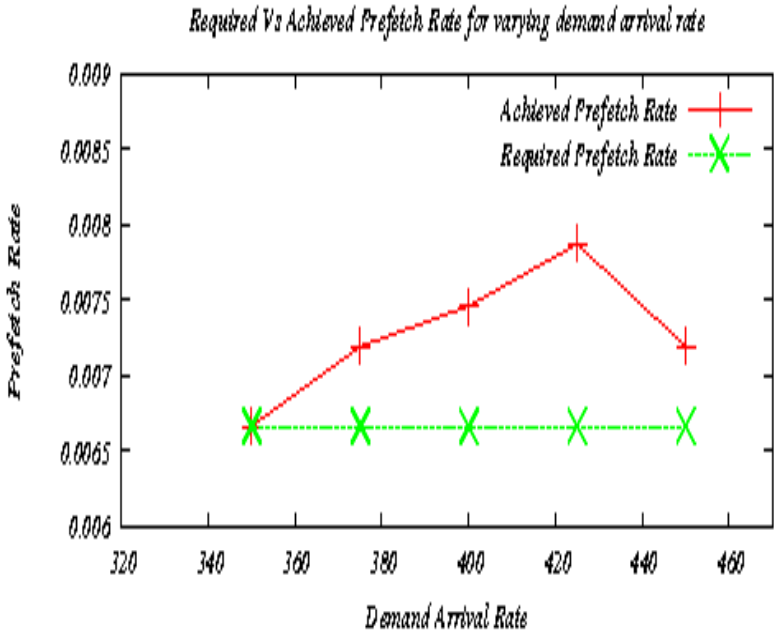


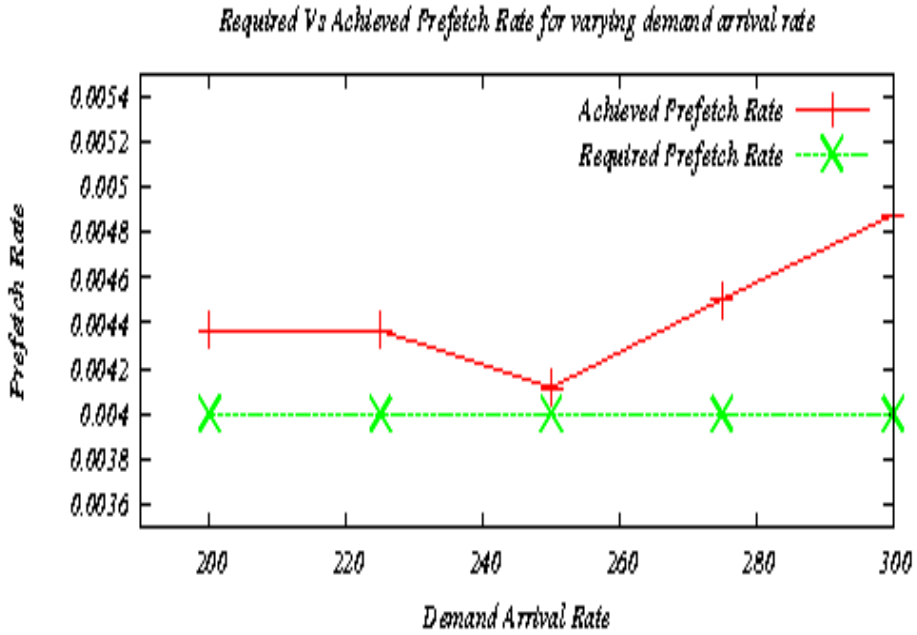
Exploring the Space, for a given demand miss inter arrival time, 100 microsec



Using the explored Data

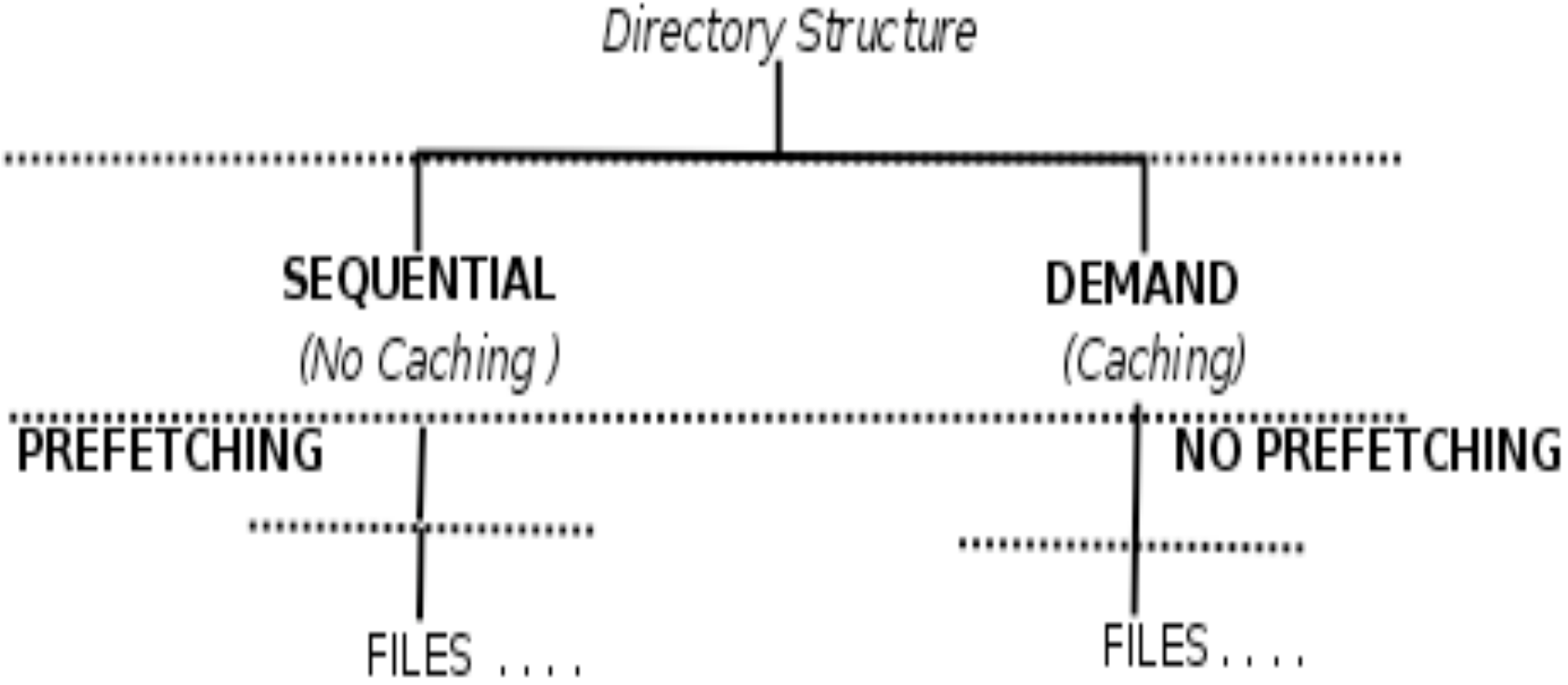
- Optimal points → database
- Database used in simulation
- Dynamically determine the value of P and D.





Development and future work

- Experimental File System (EFS)
  - To find out real time performance



Published Papers

- The Design of a Storage Architecture for Mobile Heterogeneous Devices icns, IEEE Computer Society, 2007, 0, 41
- Network Memory Servers: An idea whose time has come Multi-Service Networks (MSN), 2004
- Modelling and Performability Analysis of Network Memory Servers ANSS '06: Proceedings of the 39th annual Symposium on Simulation, IEEE Computer Society, 2006, 127-134
- Modelling Network Memory Servers with Parallel Processors, Break-downs and Repairs ANSS '07: Proceedings of the 40th Annual Simulation Symposium, IEEE Computer Society, 2007, 11-20

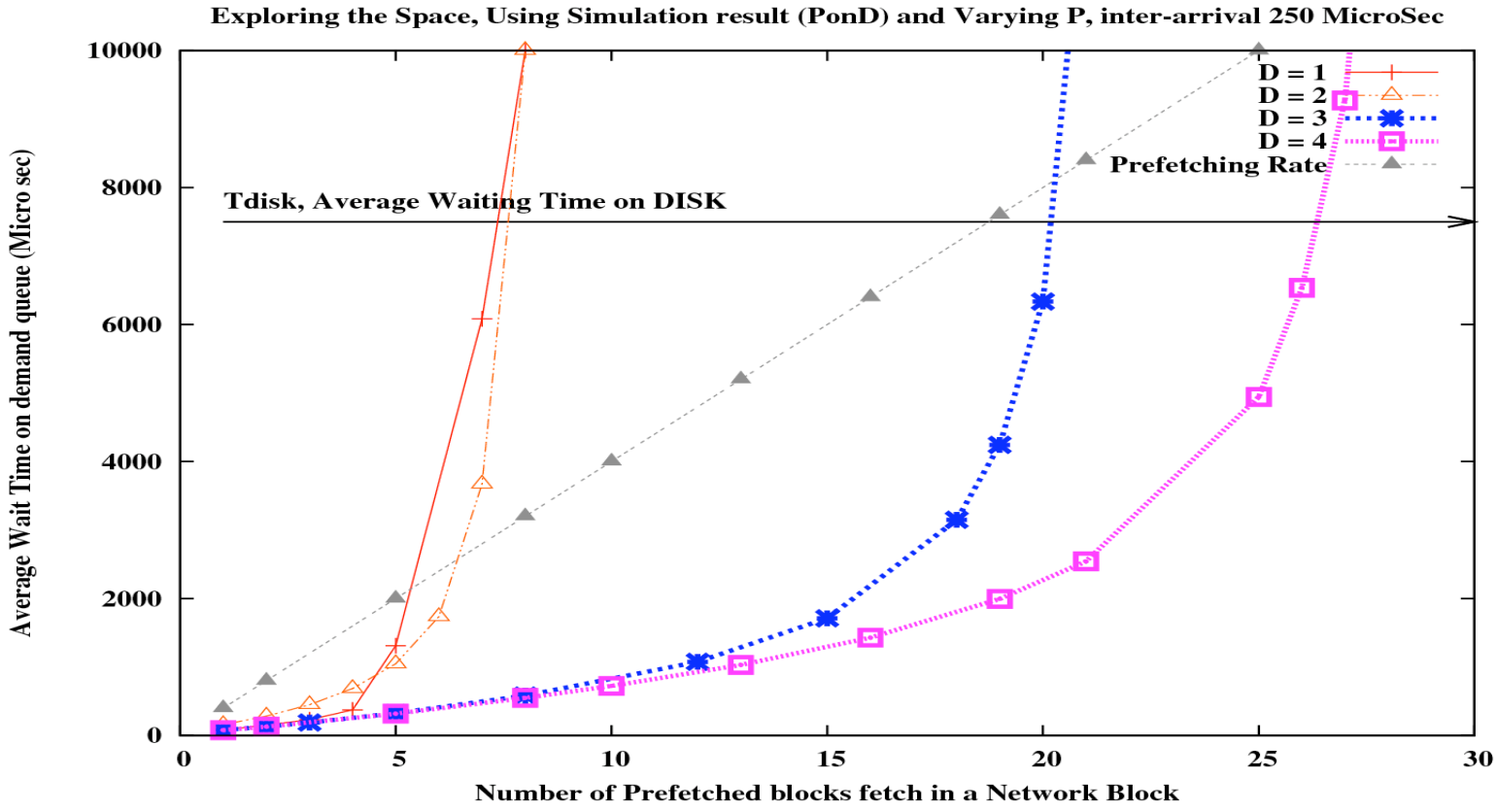
REFERENCES

- Cao, P.; Application-Controlled File Caching Policies, 1994
- Gemikonakli, O.; Mapp, G.; Thakker, D. & Ever, E. Modelling and Performability Analysis of Network Memory Servers, 2006
- D. Rochberg et al. Prefetching over a network: early experience with CTIP. SIGMETRICS Perform. Eval. Rev., 25(3):29–36, 1997
- R. H. Patterson et al. Informed Prefetching and Caching. In SOSP '95:Proceedings of the fifteenth ACM symposium on Operating systems principles, 1995. ACM
- C. Li, K. Shen et al., Competitive prefetching for concurrent sequential I/O. SIGOPS Oper. Syst., 2007
- A. E. Papathanasiou et al., Energy efficient prefetching and caching. In ATEC: USENIX Annual Technical Conference, 2004. USENIX Association.

Thank You

QUESTIONS?

Exploring the Space, 250microsec



Exploring the Space, 350microsec