Storage Quality of Service for Realtime clouds

A Tutorial by Ganesan Umanesan, Xyratex

10th IRMOS GA, Oslo
September 2010

Data Storage Issues in Real Time Clouds

- "Last Mile" for Data Access in Realtime clouds
  - Adherence to ultimate application QoS through QoS methodologies in Storage
- Scalability to accommodate highly distributed applications
- Highly sharable storage architecture to provide shared storage
- Leveraging Commodity storage subsystems

Existing Enterprise Storage

- Provisioned for Worst case performance, and hence significantly suboptimal
  - Very high bandwidths are assumed even before applications are deployed
  - Very high capacities are assumed
  - Unsuitable for a service oriented delivery model
- Highly inflexible for dealing with changing application requirements
- There are absolute limits on Scalability

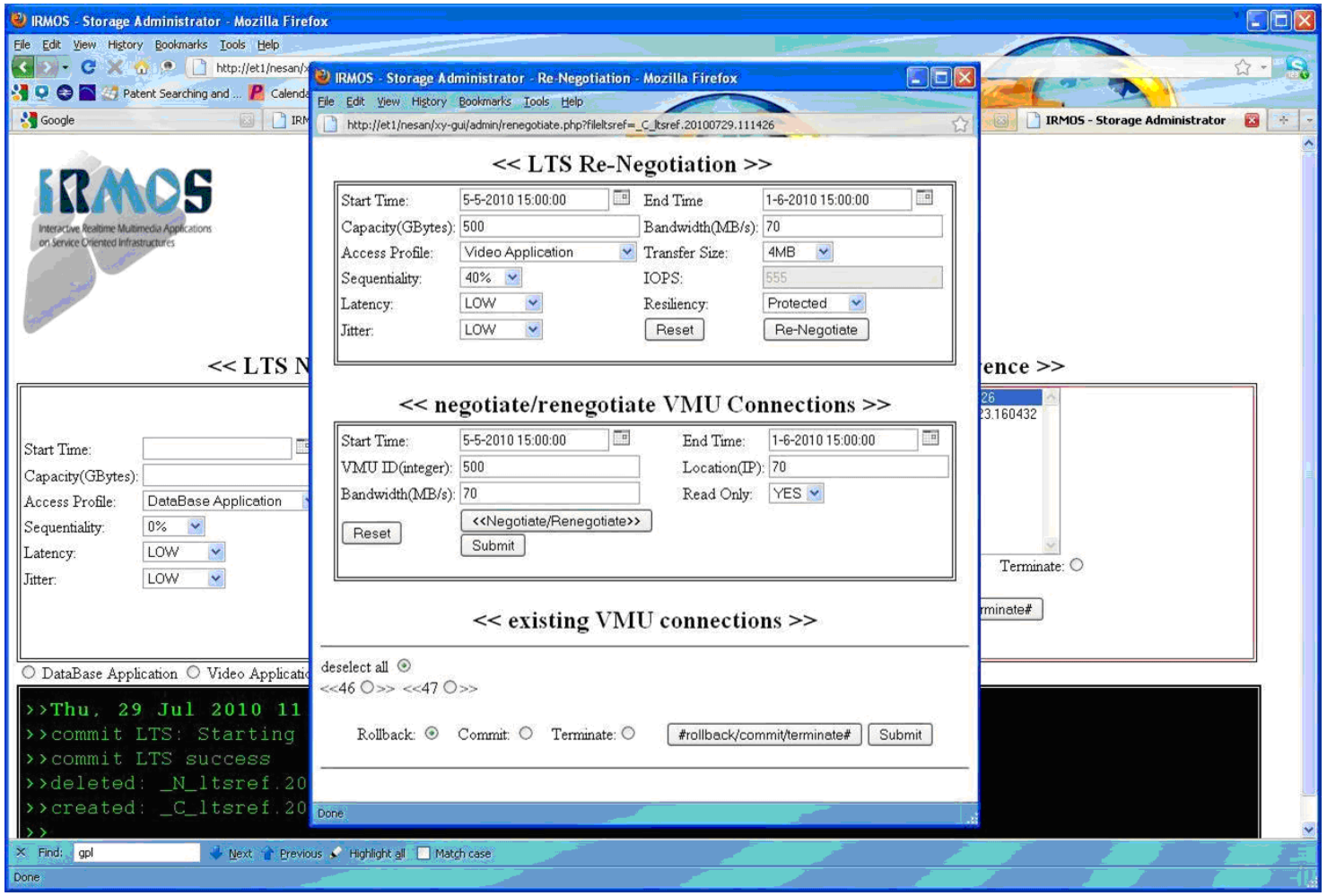
Existing Cloud Storage

- Amazon S3, Nirvanix SDP, EMC Mozy, Dropbox, Yahoo Zumo Drive etc
  - Very early stage dealing with a limited set of non QoS intensive apps.
  - Very coarse grained QoS features
    - Average bandwidth usage over a month etc
  - Highly Unsuitable for real time applications
    - Fine grain QoS features needed like near- instantaneous bandwidth and latency
  - Are not targeted for highly I/O intensive storage workloads such as
    - Scientific Simulation environments
    - Digital and 3D Film industry

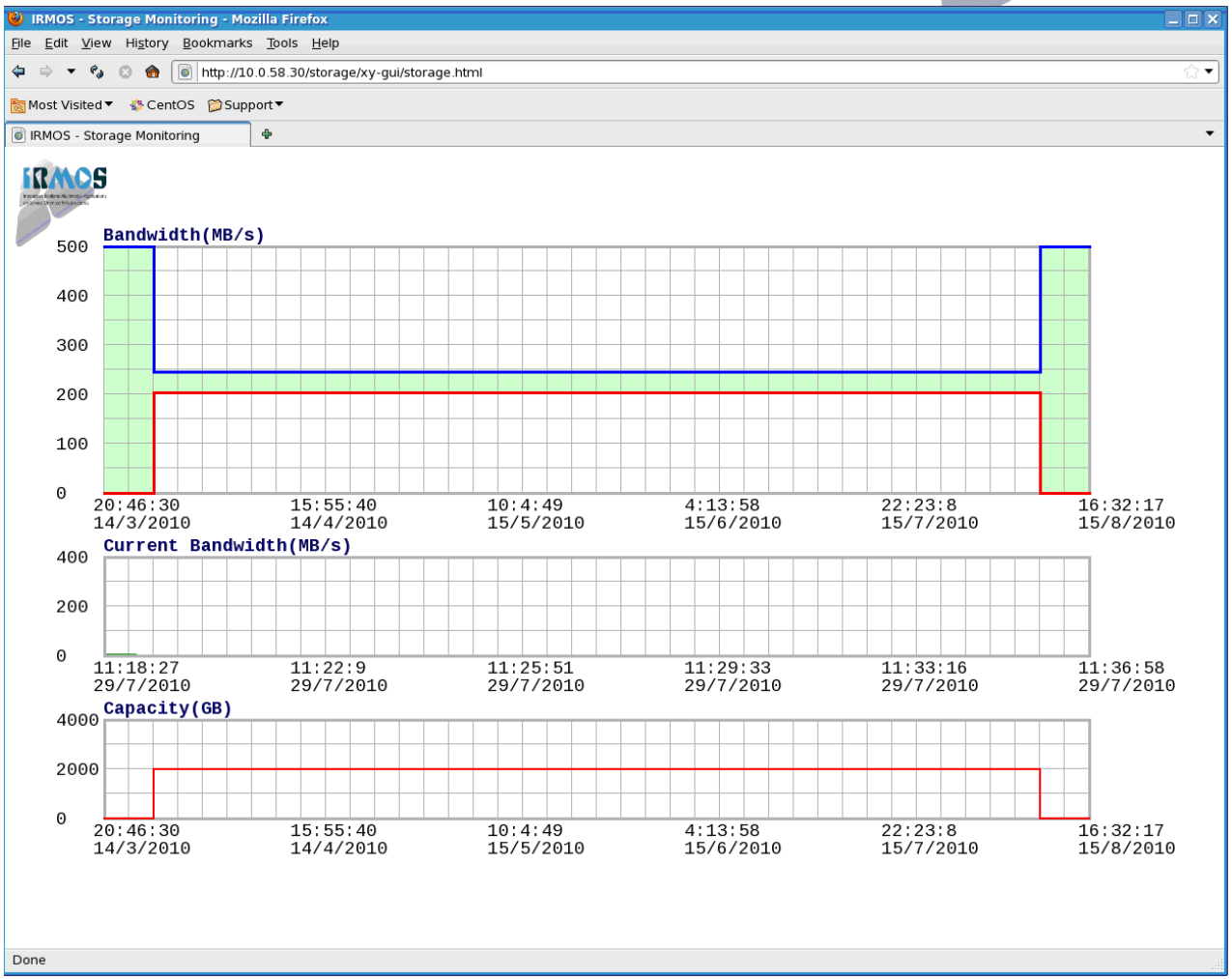
Key aspects of IRMOS storage

- Application SLAs mapped to Technical SLAs, which contain storage related requirements
  - Capacity
  - Bandwidth
  - Latency
  - Resiliency
  - Jitter
  - Lifetime
- Provision of guarantees for these parameters on behalf of application

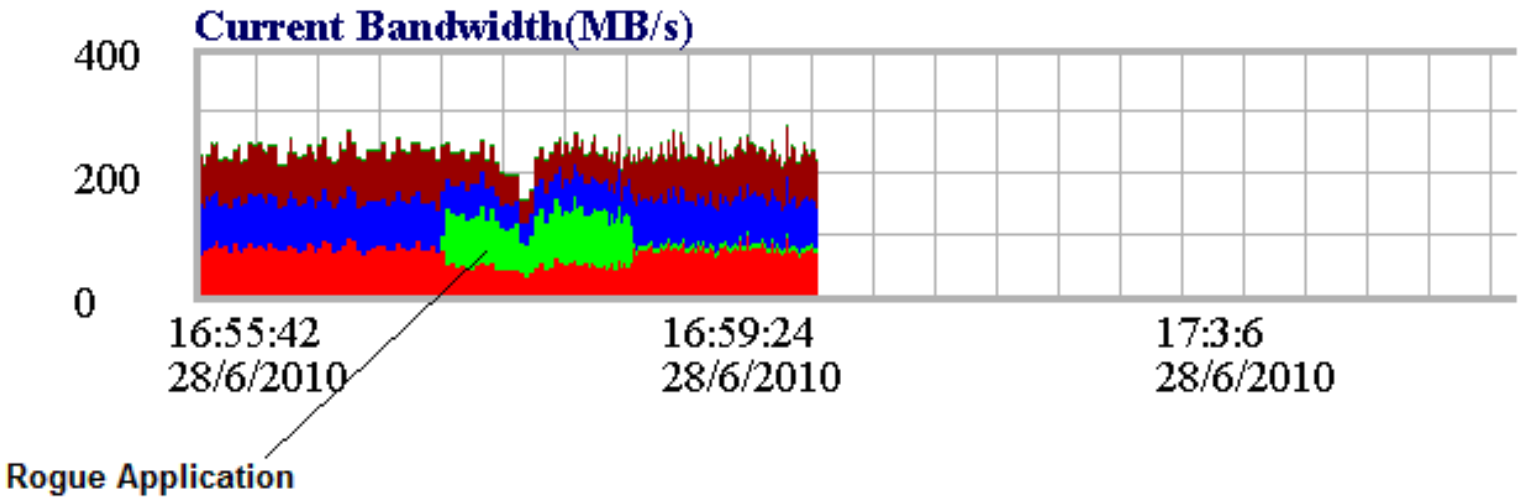
Interface to IRMOS QoS Aware Storage (Storage Manager)



IRMOS QoS Aware Storage Monitoring

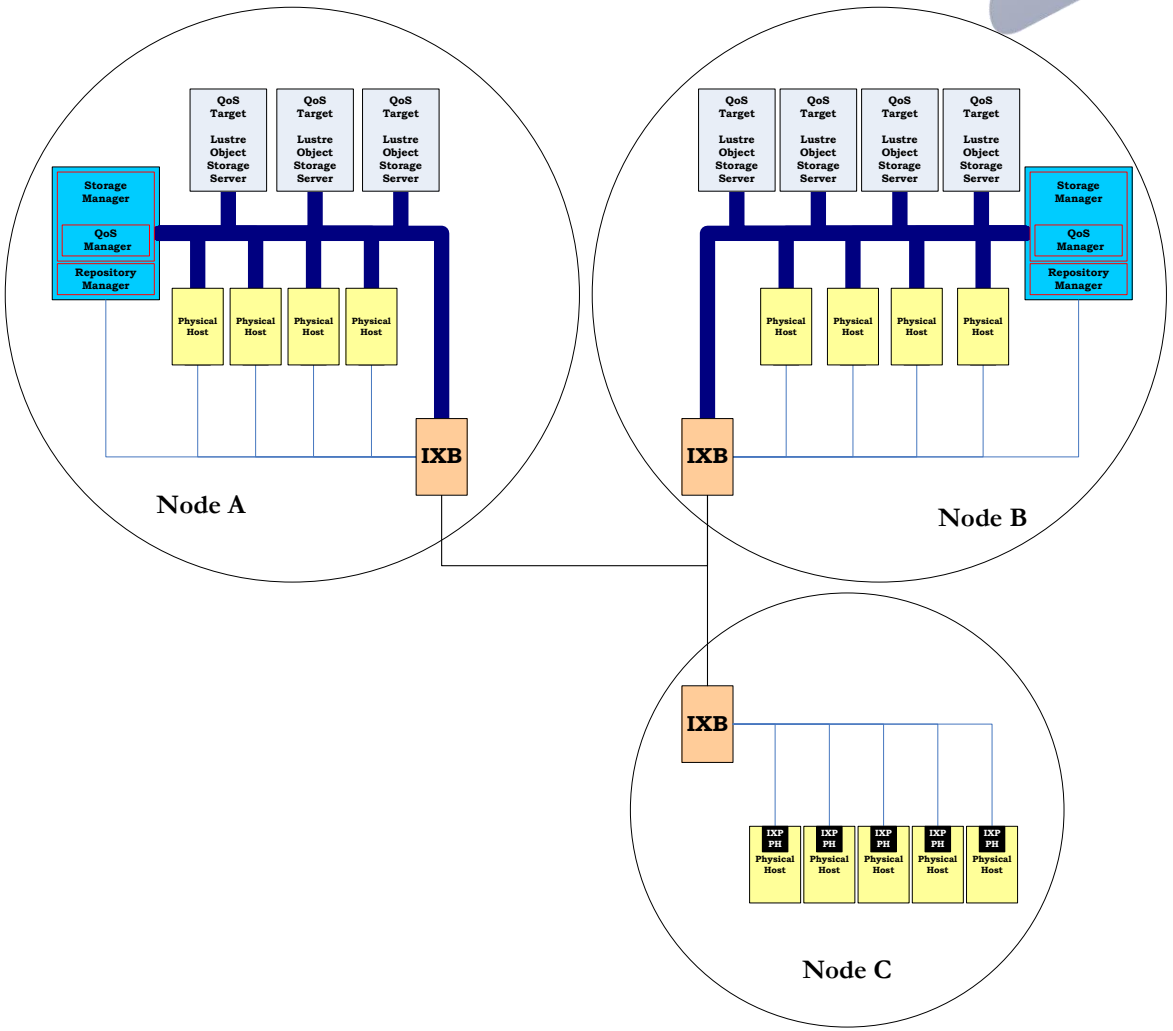


Example of QoS Guarantees on Bandwidth

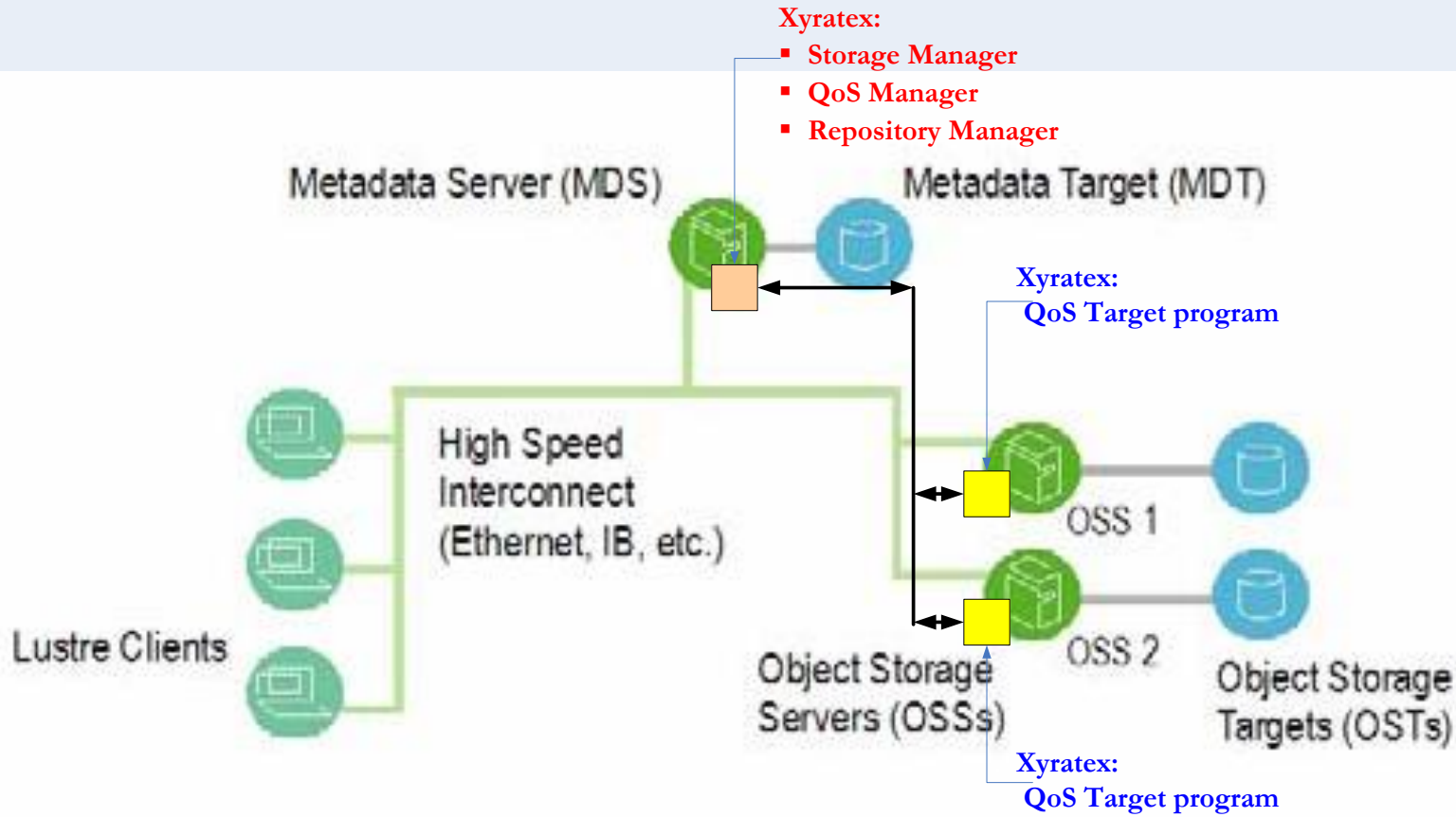


Architecture

Storage Components

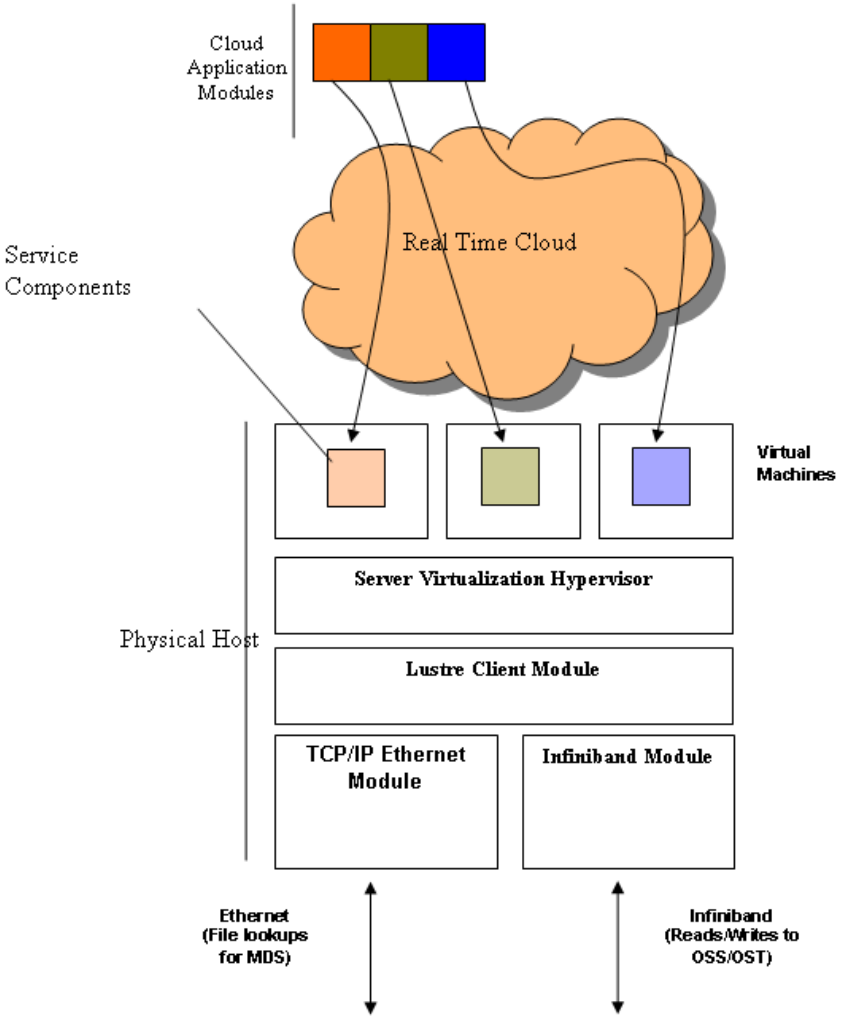


Lustre File System



Lustre™ 1.8 Operations Manual http://www.sun.com/software/products/lustre/

Lustre File System..



Lustre File System..

- Lustre Clients with VMUs can scale linearly and be distributed in a cloud
- Performance scales linearly with added number of OSS
- High speed infiniband provides very high upper bounds on bandwidth/latency performance
- However lacks QoS features

QoS in the Lustre Framework

Objectives

- Accurate allocation of storage resources (eg. User Bandwidth and System Capacity) to satisfy an SLA
- An analytical model for system usage for the storage system as a whole which takes into consideration
  - Changes in User Block Size requests
  - Changes in User Sequentialities for Data access
  - Changes in Access profiles (Percentage of reads/writes)
  - Inclusion of new client requests , and hence the number of clients
- Ability to accept and reject connections based on available system capability for I/Os based on the model

Key Issues

- Can the storage system pre-specify, Guarantee (corresponding to an Incoming SLA) and control the bandwidth available to a user?
- When the storage system sees the request for an SLA corresponding to the above pre-specified (Threshold) bandwidth,
  - Can the SLA for the given bandwidth request be granted?
  - What are the implications of granting the request on available System Capability?

QoS Guaranteed Bandwidth

- Capability to throttle and control the bandwidth for clients to a fixed value (if permissible by the capability available in the system)
- Scales to multiple clients accessing Long Term Storage
- Different levels of throttling can be provided based on SLAs from the application

Bandwidth Throttling Implementation

- Bandwidth throttling is implemented by a credit based flow control mechanism implemented on the storage side
- Currently implemented on the Lustre OSS
  - Throttling can be turned “on” or “off”
- No changes required for Clients accessing the Lustre OSS

QoS Analytical Model

- Making decisions as to whether to allow an SLA requesting a given (throttled) bandwidth

- "System Capability"

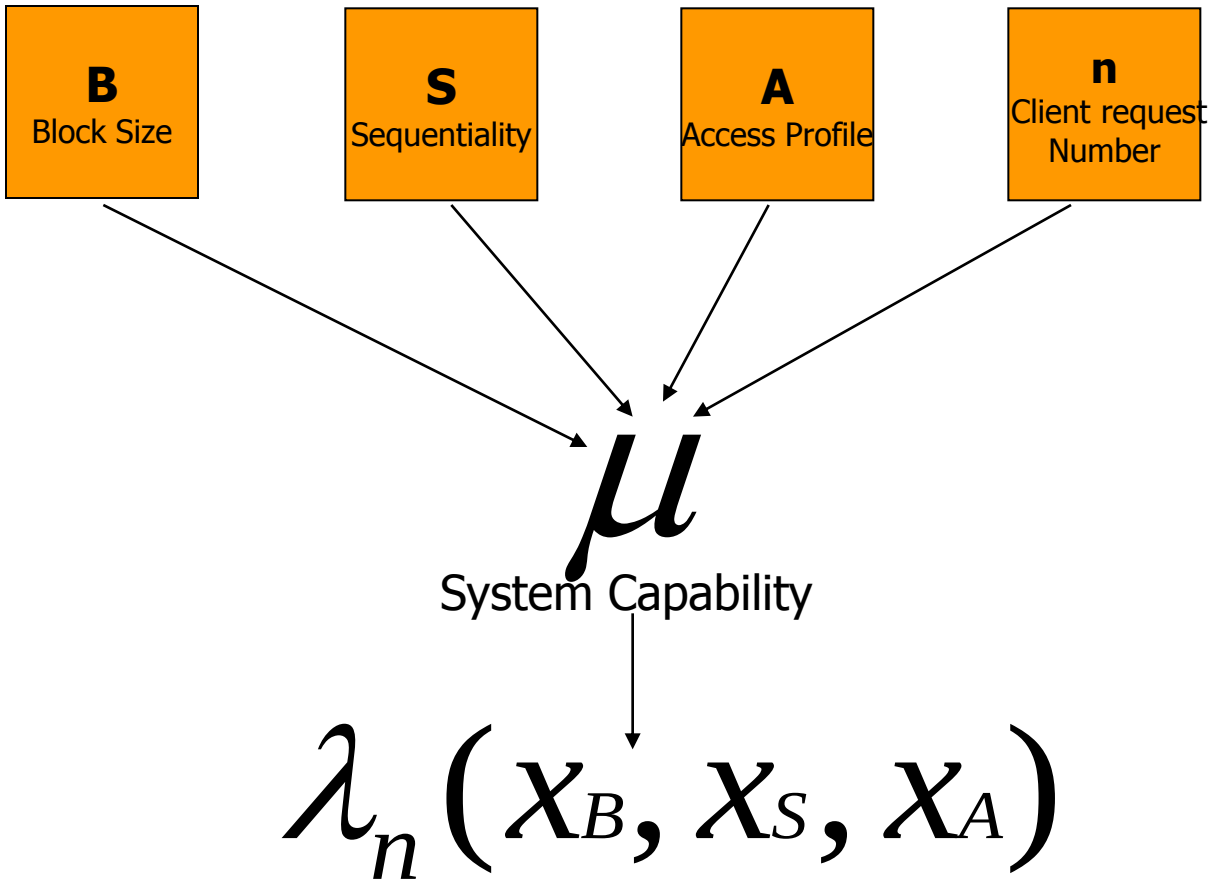
  - Maximum Theoretical bandwidth that can be accommodated by the system

  - Metric Incorporates the entire storage subsystem(OS, Back-end disks, etc) rather than "piece-meal" capabilities of the individual components
    - The Disks
    - The Disk Caches
    - RAID
    - Disk/RAID drivers
    - Back-end file system
    - The Lustre OSS software layers
    - The networking subsystem to connect to clients
    - Operating system memory memory/paging subsystem
    - Operating system caches
    - Hardware limitations (Bus bandwidths, front side bus bandwidth etc)
    - …And more!

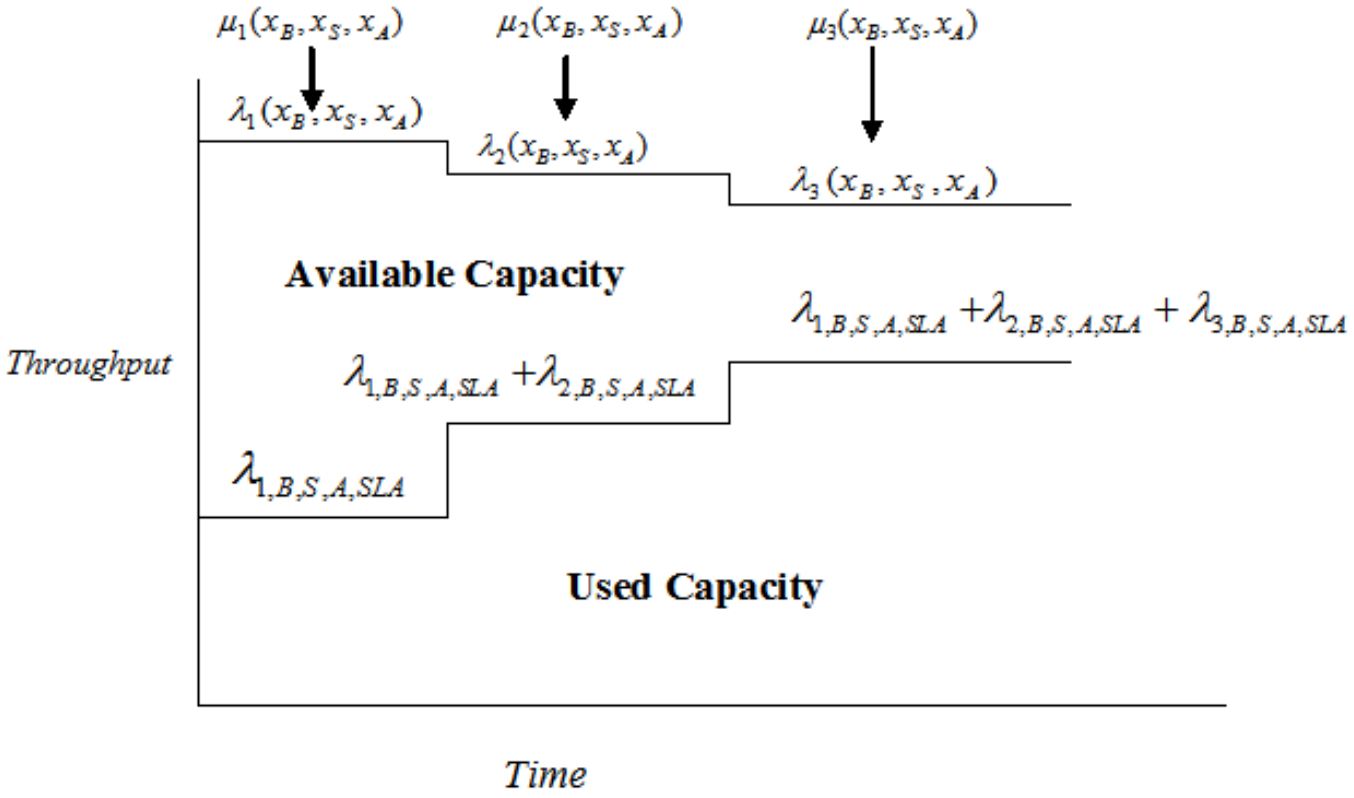
System Capability

- Motivation for System Capability
  - Existing approaches characterize capabilities for the different individual components of the storage subsystem
  - The net system capability available for applications are NOT simple mathematical function of the existing hardware components
- Need probabilistic models to accurately characterize the net capability available to individual applications
- We need to understand effects such as reducing maximum net bandwidth available to applications under loaded conditions (such as multiple clients)

System Capability and Max Achievable Bandwidth



Max Achievable Bandwidth Example



System Capability Derivation Methodology

- Understanding the responses of the system to a given set of inputs (changing Block Sizes, Sequentialities, Access Profiles and Number of Clients)
  - Bandwidth responses
  - Latency responses
- Application of Probabilistic Models and Queuing Models

Questions?