2013/09/17

DataDirect™ NETWORKS
INFORMATION IN MOTION™

# A New Quality of Service (QoS) Policy for Lustre Utilizing the Lustre Network Request Scheduler (NRS) Framework

**Shuichi Ihara**

DataDirect Networks Japan

# Background: Why QoS?

- Lustre throughput and metadata performance scales very well with the number of OSTs/OSSs and—with DNE—the number of MDTs/MDSs
- Lustre performance is well balanced
- But, as of today, Lustre does not offer the option to “manage” performance or to “limit” performance
- Lustre is increasingly entering application areas outside the mainstream HPC with its large parallel application (“file-per-process”) use cases
- Some of these use cases require system administrators to manage (limit, increase, prioritize) performance
- Eventually, approaches to deal with these use cases will also benefit mainstream HPC centers!

# Quality of Service (QoS)

- Quality of Service (QoS) is a mechanism to ensure a "guaranteed" performance
- QoS was developed mostly in the network world, and especially, on TCP/IP networks, which pose specific QoS challenges
- QoS features are available on many network hardware or network management software products
- QoS is somewhat less common in the storage world, although some (expensive) enterprise storage products claim QoS or QoS-like features

# LQS: A QoS Policy for Lustre

- We have developed a policy layer called “Lustre” QoS (LQS) that can provide QoS by controlling the number of RPCs handled on the Lustre servers
- LQS runs as a policy of the Network Request Scheduler (NRS)
- Eventually, LQS limits Lustre performance by limiting the number of bandwidth and/or metadata operations

# The Network Request Scheduler (NRS)

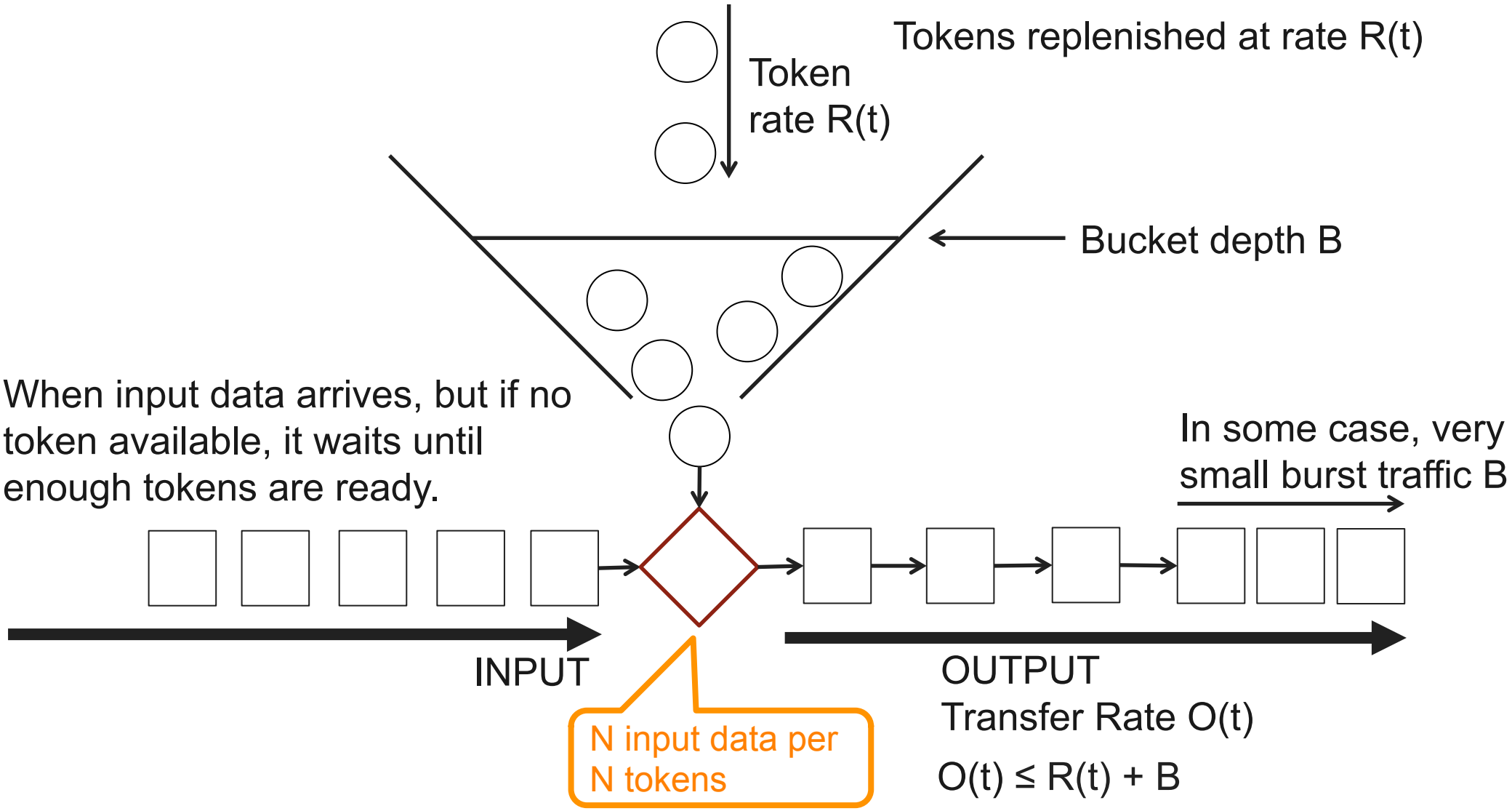
DataDirect NETWORKS
INFORMATION IN MOTION

- A new component of the PTLRPC service
- NRS works on the server side and allows handling of incoming RPCs before passing them to the OSS/MDS threads for the backend file system
- This framework, together with a few policy options, was merged into the Lustre mainstream and has been available since Lustre 2.4.0
- The NRS Framework is very flexible and it is fairly easy and straightforward to add new policies
- Policies can manage RPCs based on NID and UID/GID/ JOBID, etc..

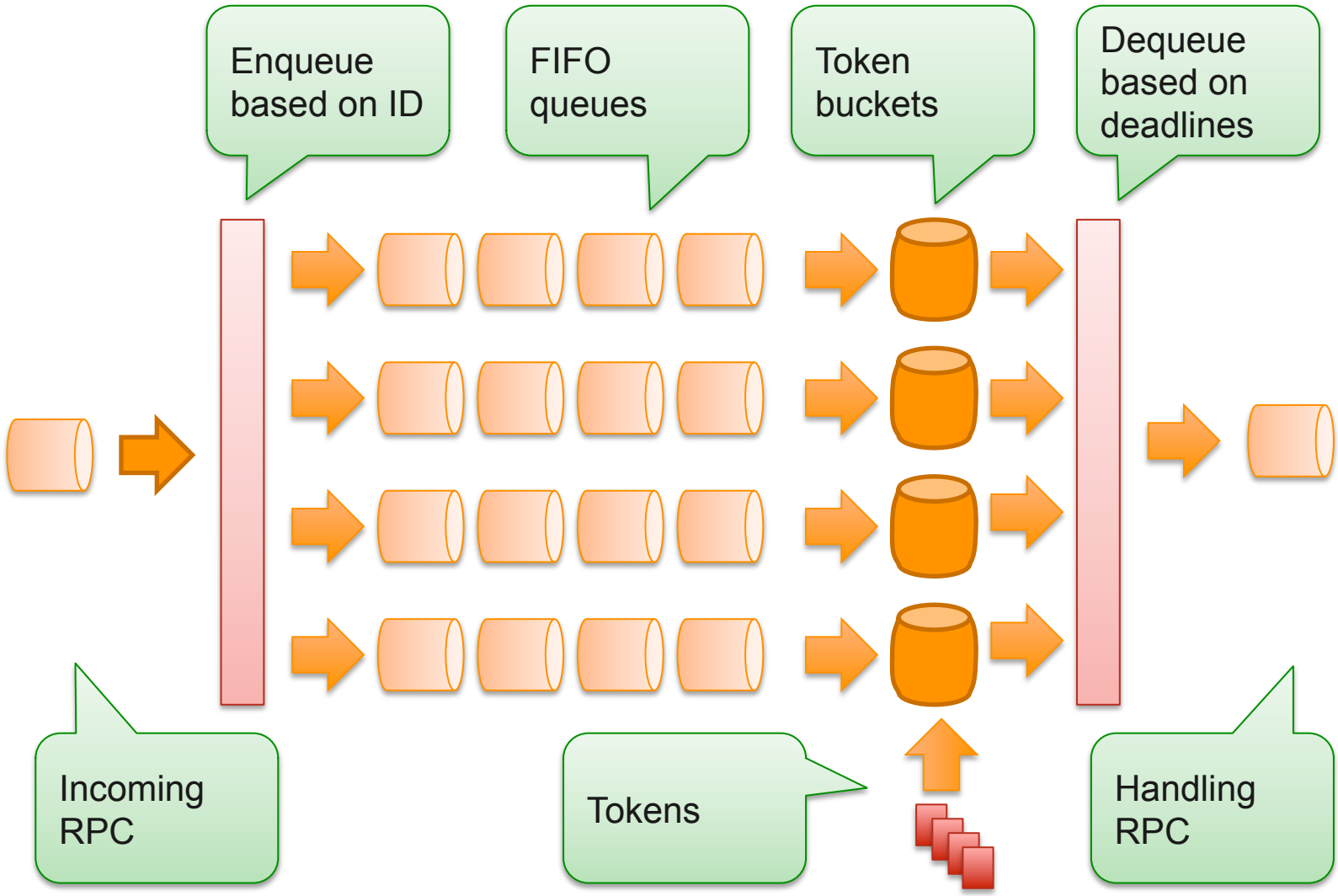
# QoS Algorithms

- Many types of QoS algorithms have been developed over the past few decades
- The Token Bucket Filter (TBF) is a major algorithm used in general network systems
  - It's simple and easy to implement
  - Many Ethernet switches and routers use TBF to enable QoS features
  - TBF can accommodate very small burst traffic, but is also OK for long-term data transmission

# The Token Bucket Filter (TBF)



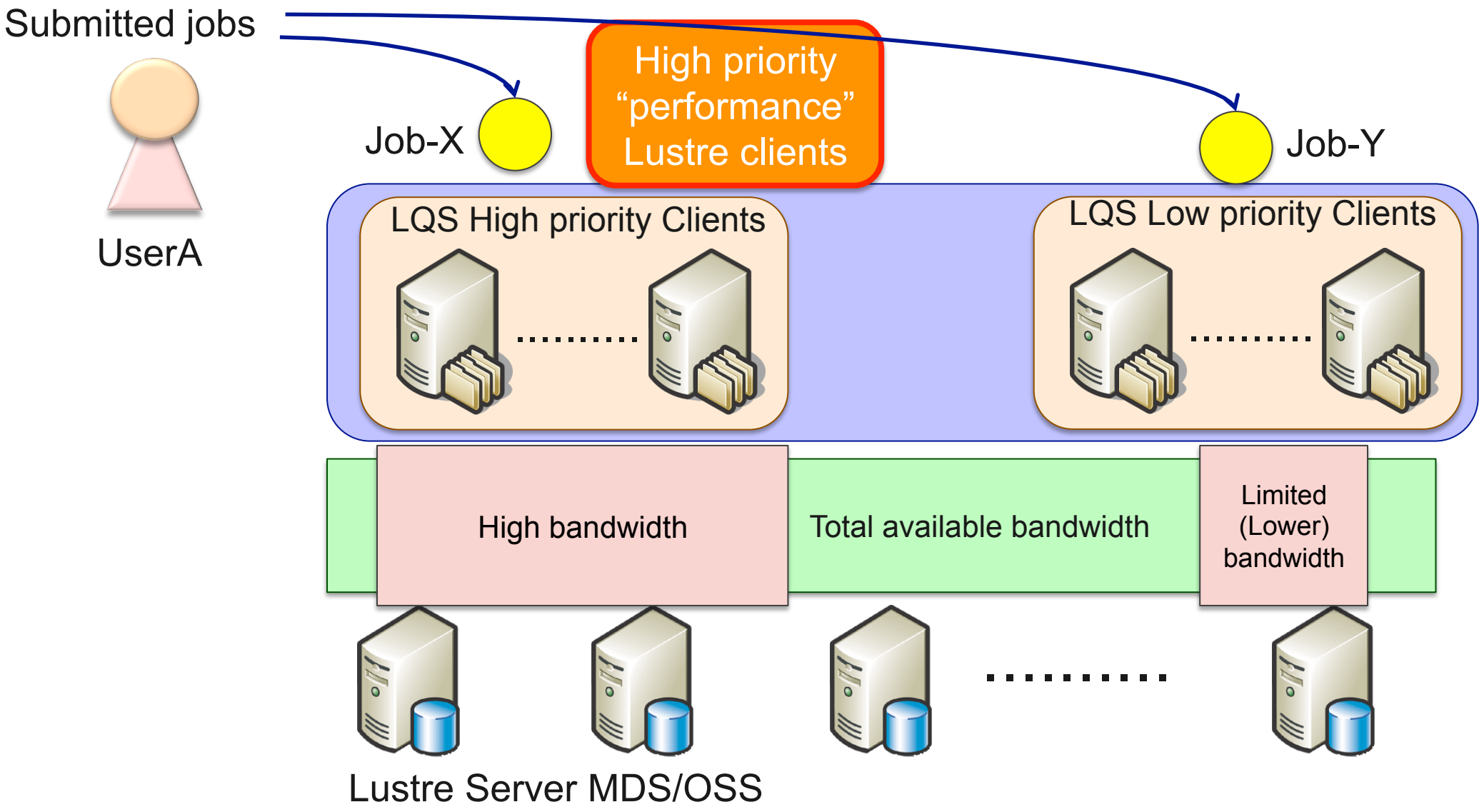
# TBF Implementation for Lustre



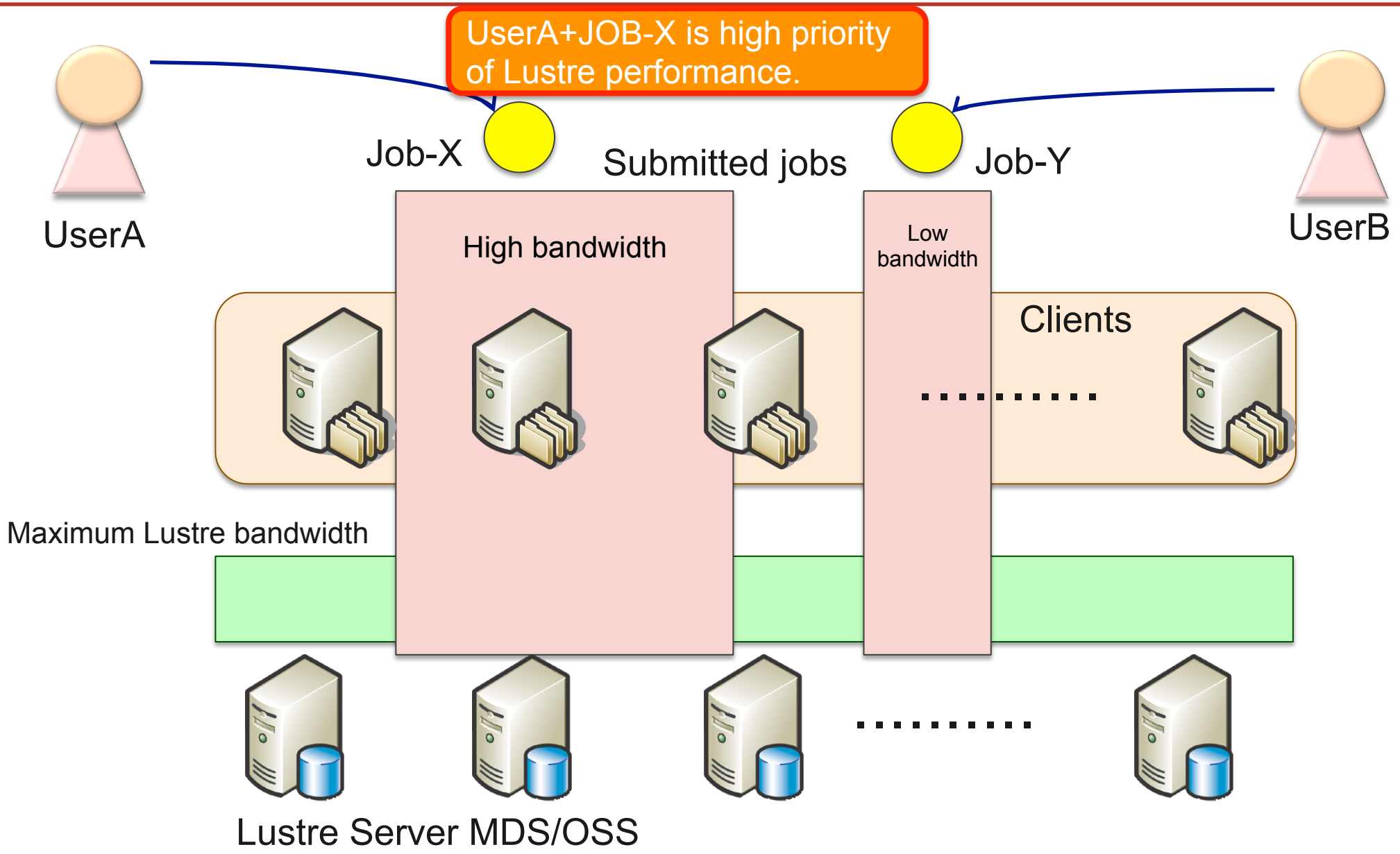
# TBF patches for Lustre

- LU-3558 ptlrpc: Add the NRS TBF policy
  Main TBF code for NRS-based policy
- LU-3495 ptlrpc: Add rate counter for request handling
  New counters in /proc to show request handing
- LU-3494 libcfs: Add relocation function to libcfs heap
  Added a function to efficiently change the rank of queue

# UseCase #1
# Lustre QoS based on NIDs (Clients)



# UseCase #2
# Lustre QoS based on JOBID



# How to use Lustre QoS

- ***Change NRS policy to TBF with NID***

***# lctl set_param ost.OSS.ost_io.nrs_policies="<NRS policy> <TBF argument>"***

# lctl set_param ost.OSS.ost_io.nrs_policies="tbf nid"

- ***Set rule with classification and number of token rate***

***# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="start <TBF's rule name> {NID} <rate>"***

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="start rule_client1 {192.168.1.1@o2ib} 1"

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="start rule_clients {192.168.1.[2-16]@o2ib} 10"

- ***Change number of token rate***

***# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="change <TBF's rule name> <new rate>"***

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="change rule_client1 100"

- ***Stop a rule (delete)***

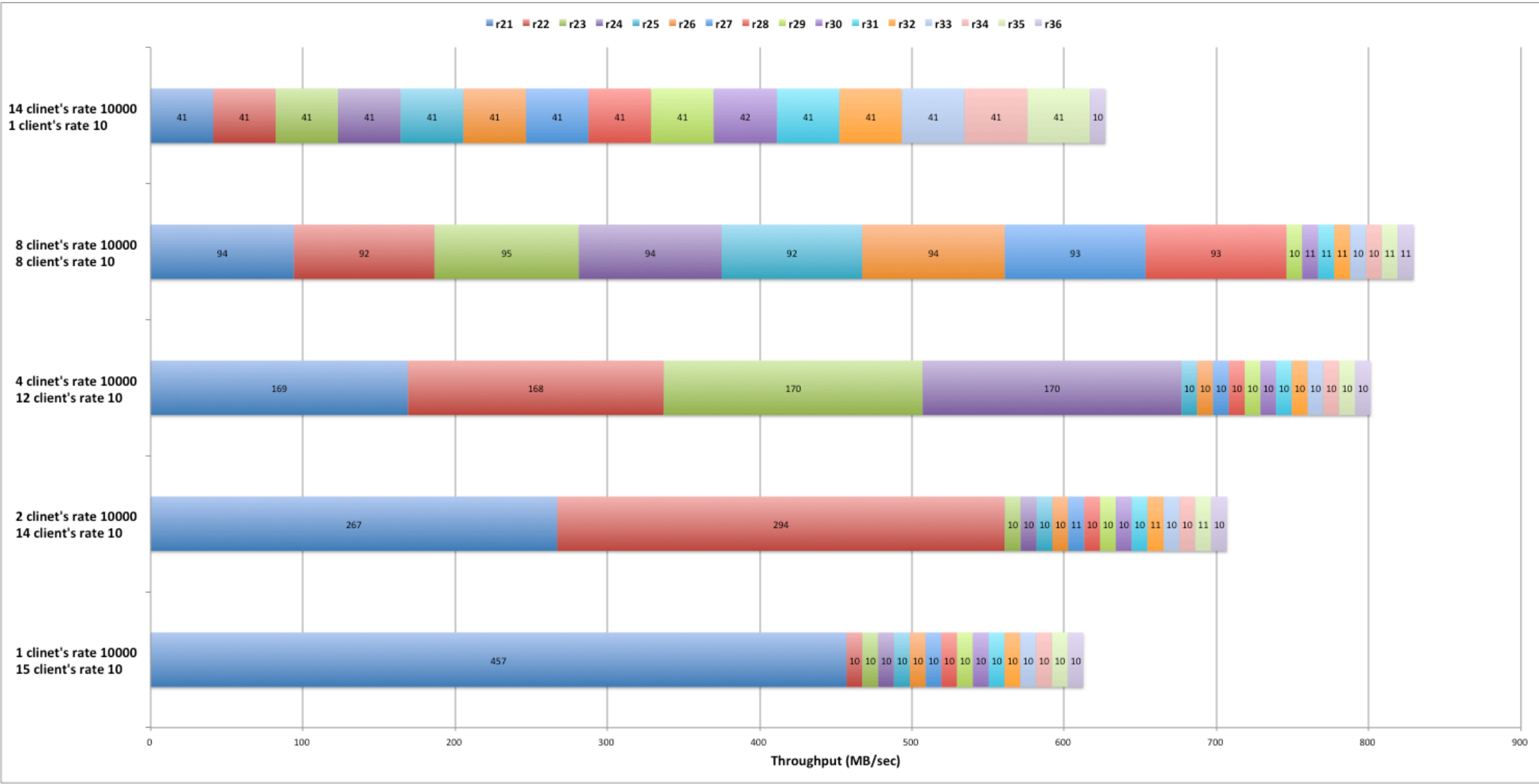
***# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="stop <TBF's rule name>"***

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="stop rule_client1"

# Lustre QoS for Clients
# Test results

DataDirect NETWORKS
INFORMATION IN MOTION

"dd" command to single OST from multiple clients with various QoS rules
(Write, 1MB IO, max_rpc_in_flright=32)



# QoS for UID with Jobstats
# Test result

***Start JOBstats and changed NRS policy to TBF with JOBID***

# lctl set_param jobid_var=procname_uid
# lctl set_param ost.OSS.ost_io.nrs_policies="tbf jobid"

***Set rule with classification and number of token***

***# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="start <TBF's rule name> {JOBID} <rate>"***

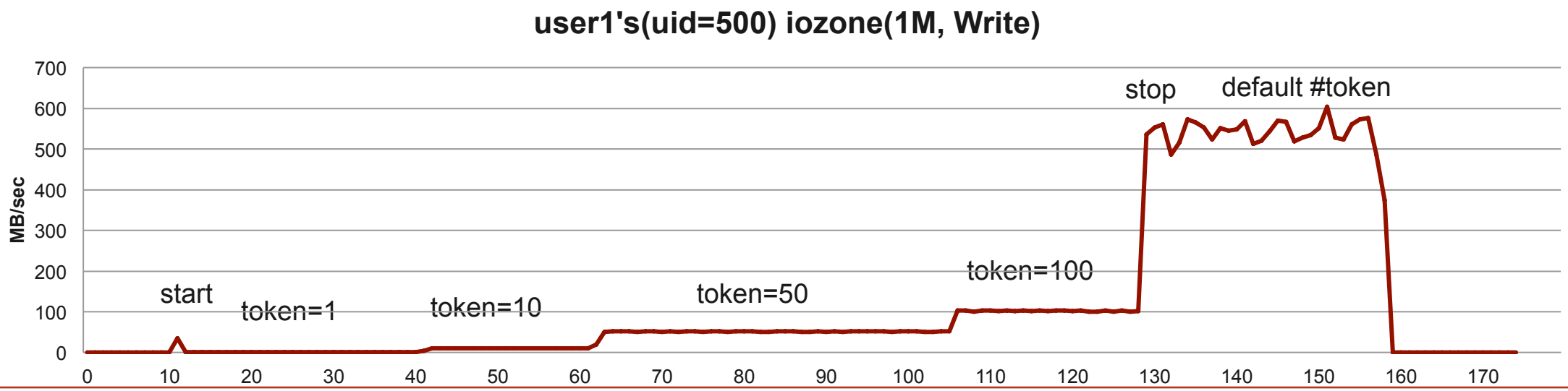
# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="start iozone_user1 {iozone.500} 1"

***Change number of token***

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="change iozone_user1 X" (change X to 10,50 and 100)

***Stop a rule (delete)***

# lctl set_param ost.OSS.ost_io.nrs_tbf_rule="stop iozone_user1"



# Conclusion

- We adapted a standard Token Bucket Filter (TBF) algorithm to Lustre and implemented LQS, a QoS policy based on the Lustre Network Request Scheduler (NRS) framework
- We demonstrated that it is possible to manage Lustre performance selectively and with high accuracy, discriminating by NID or JOBID. (We are still more testing!)
- As of today, this approach only supports simple QoS rules with only a single discriminator present at any time
  - A rule with multiple discriminators appears possible, but is still under investigation
  - Changes in the NRS framework itself may be necessary to implement more complex policies with multiple discriminators

DataDirect™
NETWORKS
INFORMATION IN MOTION™

# Thank you!