Quality of Service in Lustre a hands-on approach

Etienne AUJAMES
eaujames@ddn.com

QoS: Quality of Services

Definition from ITU (International Telecommunication Union)

*"Totality of characteristics of a telecommunications service that bear on its **ability to satisfy** stated and implied needs of the **user** of the **service**."*
Rec. ITU-T E.800 (09/2008)

The following communication aspects are commonly considered :

- Reliability
- Transmission delay
- Throughput
- Availability
- Etc. …

Why QoS in Lustre

Lustre has many services distributed on several servers:

- Metadata services (e.g: mdt, ost)
- IO services (e.g: ost_io, mdt_io)
- Management services (e.g: mgs)

Lustre has many "users" of different types:

- Network nodes
- Jobs
- applications
- Unix users/groups

How do we maintain an acceptable storage performances for all those users?

For what purpose

- Prevent crazy applications that congest the storage
- Improve user experience, e.g. intolerable delay of ‘ls’
- Assure workloads of reliable bandwidth
- Prioritize critical administration application, e.g. space balancing when an OST pool is full
- Protect server hardware against crazy load, e.g. during RAID array reconstruction

Type of QoS in Lustre

Interconnect:

- e.g: Infiniband QoS (with the concept Virtual Lane and Service Lane).

LNet:

- Multi-Rail health algorithm: use to depreciate the usage of a local or remote interface if it return a lot of error.
- Multi-Rail User Defined Selection Policy (UDSP): allow policies for local/remote interface prioritization by NID.

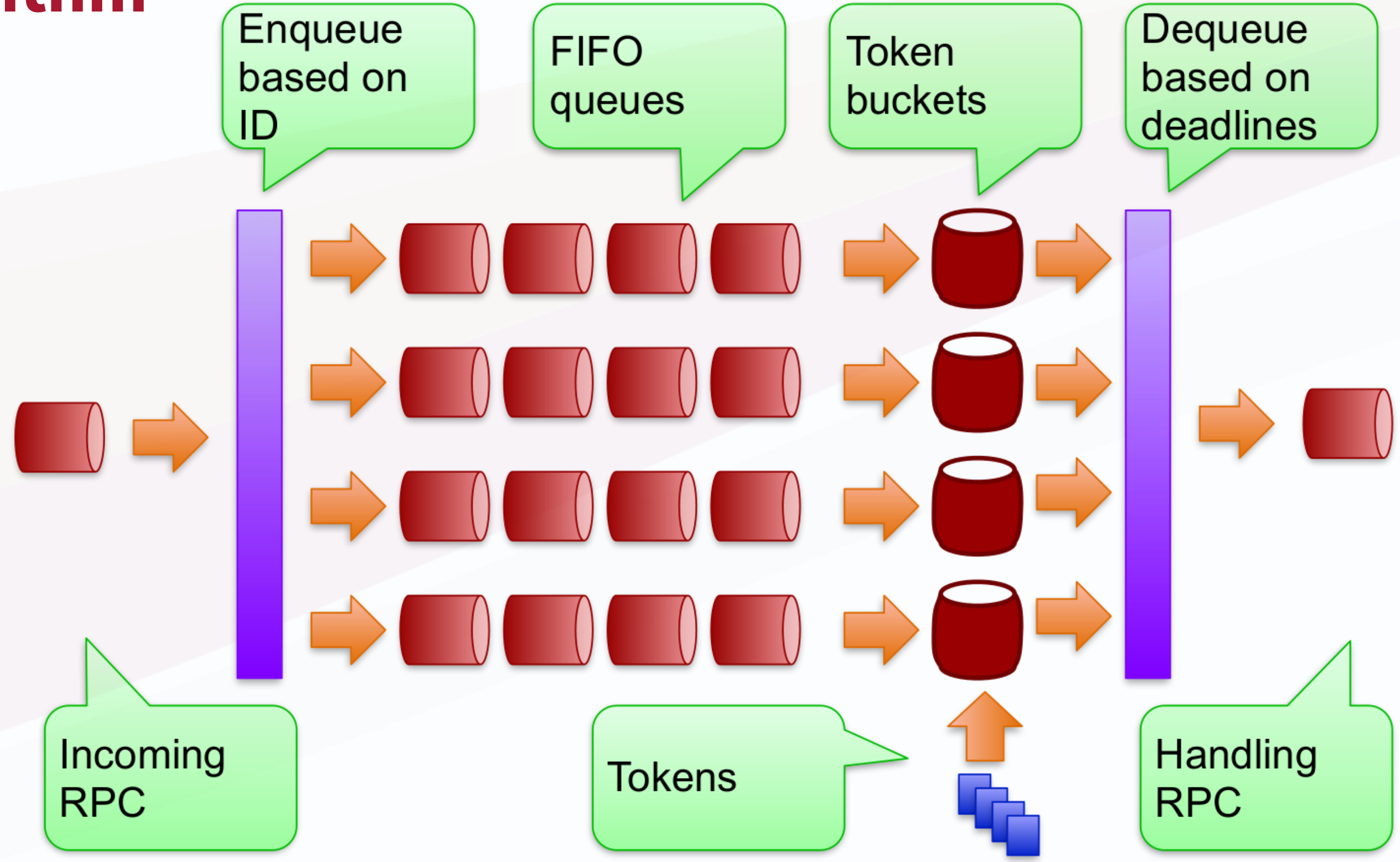
Token bucket filter (TBF):

- Allow the administrator to define rules to enforce the RPC rate limit on it.

Token bucket filter (TBF), a NRS policy

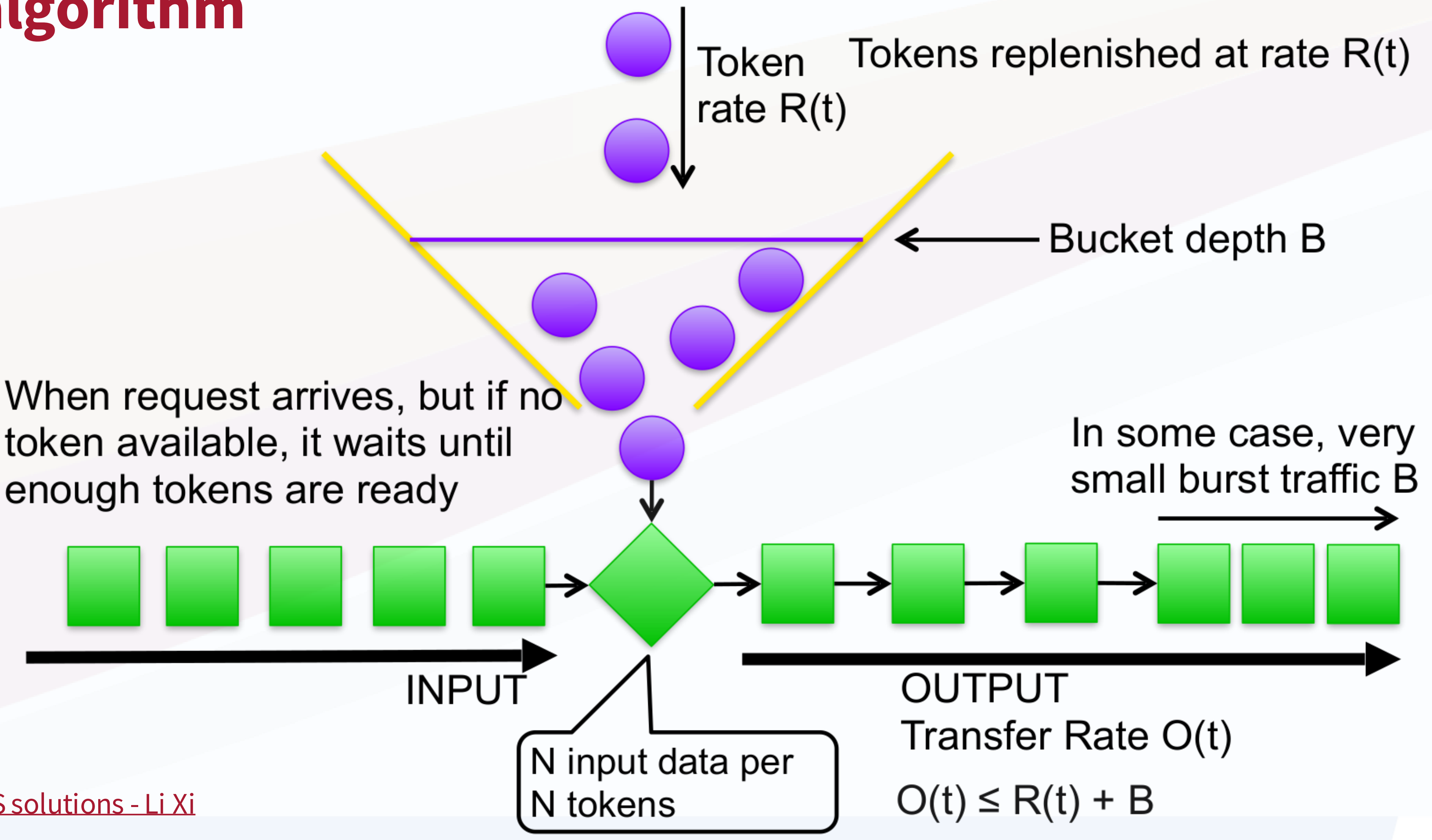
- **NRS** (Network Request Scheduler) is able to reschedule/resort/throttle the RPCs before forwarding them to the handling threads on MDS/OSS
- **TBF** (Token Bucket Filter) is the policy that enables NRS to throttle RPC rates by user defined rules
- **TBF rule** defines an RPC filter to enforce a rate limit on the matching requests. This can filter requests based on:
  - *JobID*: Lustre's job id. Simple wildcard can be used (e.g: "jobid={io*.10* *.500}").
  - *NIDs*: Lustre network id (e.g: "nid={192.168.*.*@tcp}")
  - *UID/GID*: Unix UID/GID (e.g: "uid={500}")
  - *Opcode*: RPC operation code to filter specific request (e.g: "opcode={ost_read}")
  - *Combination*: mix several type of rule in one (e.g: "opcode={ost_write}&jobid={dd.0}")

TBF algorithm



Source: Lustre QoS solutions - Li Xi

TBF algorithm


Token
rate R(t)
Tokens replenished at rate R(t)
Bucket depth B
When request arrives, but if no token available, it waits until enough tokens are ready
In some case, very small burst traffic B
INPUT
N input data per N tokens
OUTPUT
Transfer Rate O(t)
O(t) ≤ R(t) + B

Source: Lustre QoS solutions - Li Xi

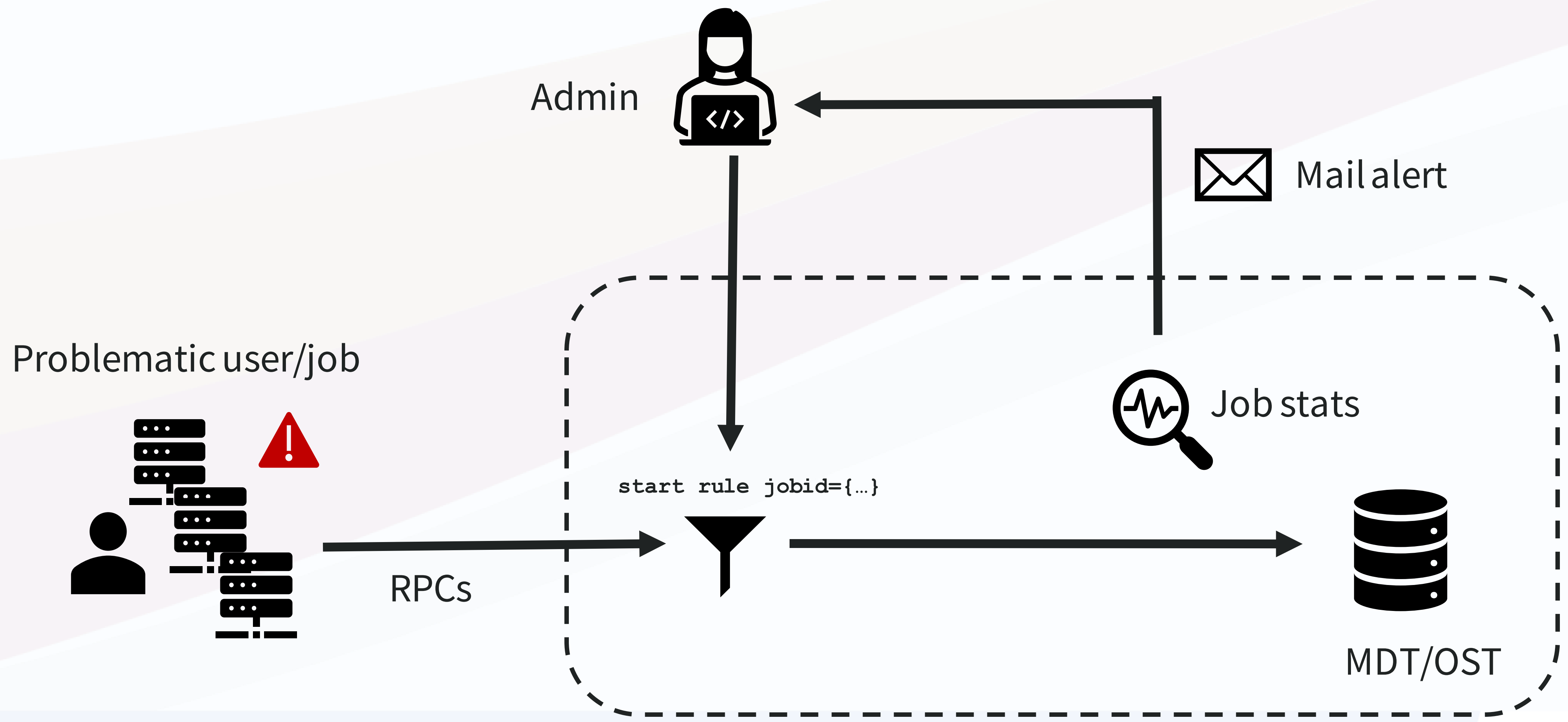
Examples

- **start** the NRS TBF policy for the service ost_io
  ```
  lctl set_param ost.OSS.ost_io.nrs_policies=tbf
  ```
- **start** rule for all the login nodes on the OSS
  ```
  lctl set_param ost.OSS.ost_io.nrs_tbf_rule=\
  "start loginnode nid={192.168.[1-4].1@tcp} rate=1000"
  ```
- **start** rule to limit the write rate of the "dd" root user jobs
  ```
  lctl set_param ost.OSS.ost_io.nrs_tbf_rule=\
  "start dd_rule opcode={ost_write}&jobid={dd.0} rate=100"
  ```
- **change** default bucket rate (from 10000 to 100000 RPC/s)
  ```
  lctl set_param ost.OSS.ost_io.nrs_tbf_rule="change default rate=100000"
  ```
- **show** the TBF rules
  ```
  lctl get_param ost.OSS.ost_io.nrs_tbf_rule
  ```
- **stop** the dd rule
  ```
  lctl set_param ost.OSS.ost_io.nrs_tbf_rule="stop dd_rule"
  ```

Limitations

- TBF enforces limits on RPC rate not directly on the bandwidth: it needs to be estimated with the RPC rate.
- Rate limitation is applied on the CPU partition of a server (CPT) for a specific service: global filesystem limitation can be difficult to estimate
- TBF does not implements guarantee minimum rate. For that purpose, rules have to be dynamically changed in function of the server loads (feedback loop).

CEA's TBF applications



Questions