Models and algorithms for burst buffers in HPC Systems

Lionel Eyraud-Dubois, Olivier Beaumont, Guillaume Aupy

With inputs from JT Acquaviva (DDN), G. Goret (ATOS/Bull)

IO congestion in HPC systems:

- HPC applications are generating lots of data for PFS.
- Idea is to use a buffer when the I/O bandwidth is fully occupied
- The buffer can be emptied at a later time.

Note: there are other uses of Burst-buffers

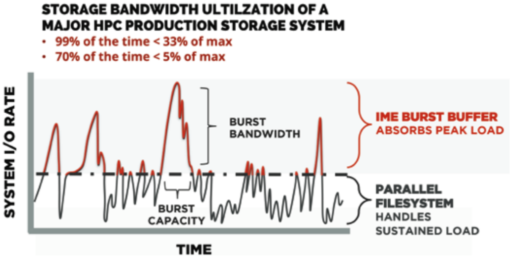


Figure: Burst-buffers to absorb IO peaks (DDN material)

Possible usage of Burst-Buffers

Historically, Burst-Buffers were attached to IONodes (ION), used as buffers when the I/O Bandwidth was not enough (Gordon@SDSC).

But many other possible uses:

- For temporary data that may not be needed (e.g. fault-tolerance)
- For intermediate data (e.g. BigData on HPC machine, In-situ/In-transit)
- For other uses?

Possible usage of Burst-Buffers

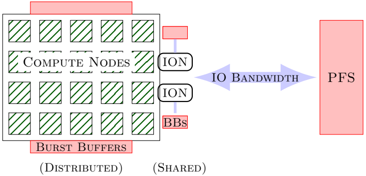
Historically, Burst-Buffers were attached to IONodes (ION), used as buffers when the I/O Bandwidth was not enough (Gordon@SDSC).

But many other possible uses:

- For temporary data that may not be needed (e.g. fault-tolerance)
- For intermediate data (e.g. BigData on HPC machine, In-situ/In-transit)
- For other uses?

How do we *design* and *dimension* our Burst-Buffer architecture depending on usage?

High level topics



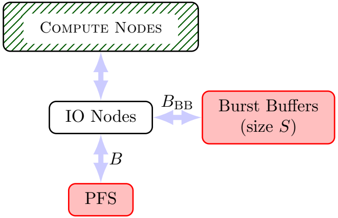
Application Modeling:

- Compute and I/O behavior, buffer needs?
- Performance model of application?
- For both HPC and BigData applications

Algorithm design:

- I/O Scheduling, Data placement, Buffer sharing
- Dimensioning: what size / bandwidth / parameters?
- Explore different designs: Distributed vs Shared, Static vs Dynamic, ...

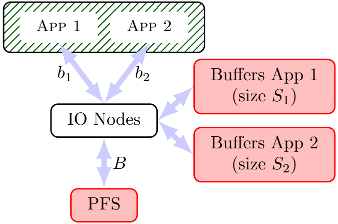
Examples of current work



Shared buffers for I/O management:

- What bandwidth B_{BB}?
- What size S?
- What filling/emptying policy?

Aupy, Beaumont, Eyraud-Dubois, `What size should your Buffers to Disks be?`, IPDPS'18

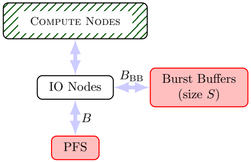


Static versus dynamic buffer sharing

- What buffer size to hide congestion?
- What overhead of static allocations?

Random burst model

We consider a unit time characteristic of the system.



Machine is characterized by:

- The Burst Buffer size S
- Its expected IO load: $\text{ExpectedLoad} = \sum_i p_i b_i$;
- Its bandwidth to PFS: B

Applications: At any time unit, application $\mathcal{A}_i$ sends data:

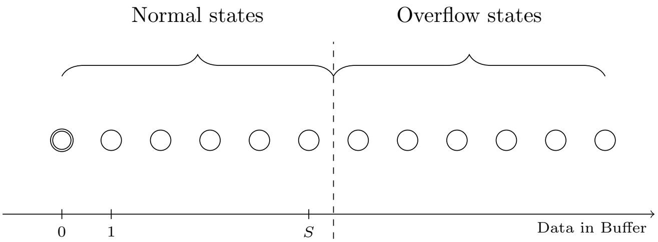
- with probability p_i
- at bandwidth b_i.

X_i: random variable indicating whether $\mathcal{A}_i$ is sending I/Os.
$\rightarrow X_i = 1$ with proba p_i and 0 with $1 - p_i$.

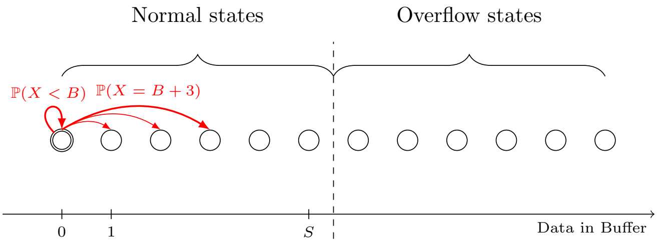
$$\text{Instant bandwidth } X = \sum_i b_i X_i$$

Modeling with Markov chains

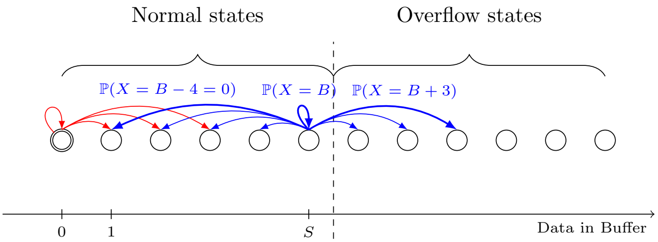
Platform model: when buffer full, stall all applications for one time unit



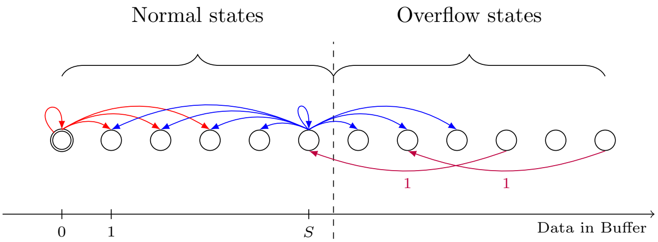
Platform model: when buffer full, stall all applications for one time unit



Platform model: when buffer full, stall all applications for one time unit

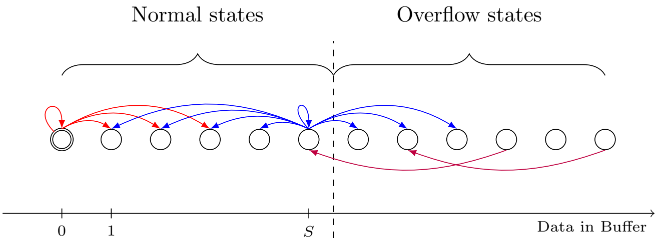


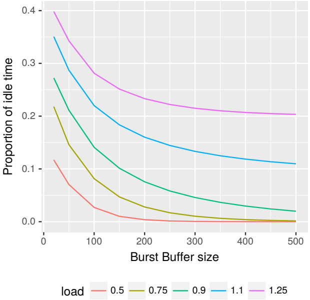
Platform model: when buffer full, stall all applications for one time unit



Modeling with Markov chains

Platform model: when buffer full, stall all applications for one time unit

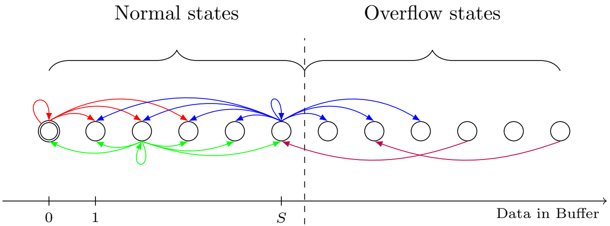




Results

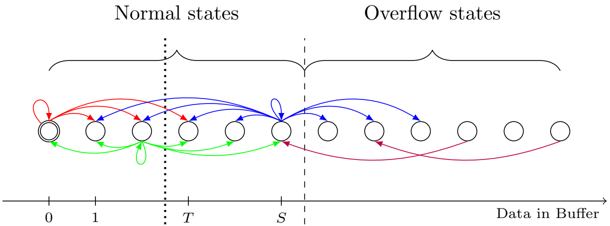
Can compute steady-state idle time for a given buffer size S

Lazy Emptying [Cluster 2017]:
Only empty the burst buffer when its load reaches a threshold T .



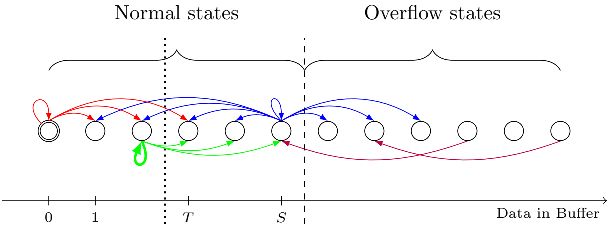
Extension to lazy emptying

Lazy Emptying [Cluster 2017]:
Only empty the burst buffer when its load reaches a threshold T .



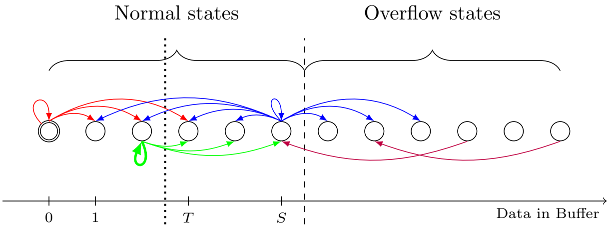
Extension to lazy emptying

Lazy Emptying [Cluster 2017]:
Only empty the burst buffer when its load reaches a threshold T .

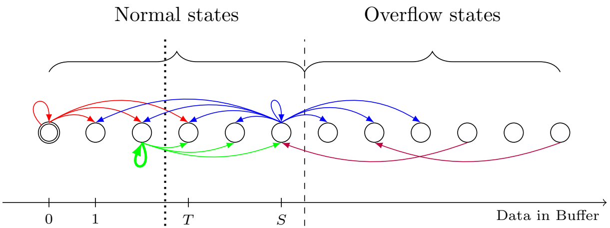


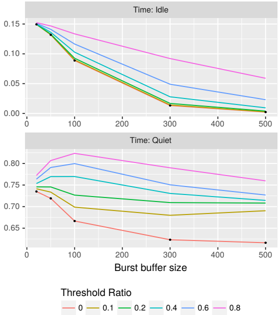
Extension to lazy emptying

Lazy Emptying [Cluster 2017]:
Only empty the burst buffer when its load reaches a threshold T .



Lazy Emptying [Cluster 2017]:
Only empty the burst buffer when its load reaches a threshold T .

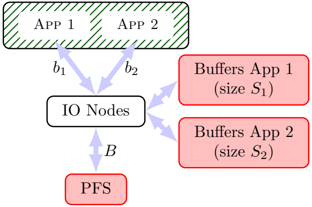




Results

Threshold ratio around 20-40% seems reasonable

Offline application model



Application model:

- divided into read-compute-write phases
- offline model: all data known in advance
- release dates

Machine model:

- Applications run independently, share the bandwidth B
- Each application communicates with bandwidth b_i
- Burst buffer is statically allocated

Questions: (solved with Linear Programming formulations)

- Buffer size to optimize an application by itself
- Additionnal buffer size to hide congestion

Topics of interest

- Burst buffer modeling and design
- Algorithms for dimensioning and/or scheduling

Critics and **suggestions** welcome!

- Interested in other people's view of Burst Buffers

Questions?