A Software-Defined Approach for QoS Control in High-Performance Computing Storage Systems

Neda Tavakoli\textsuperscript{1}, Dong Dai\textsuperscript{1}, John Jenkins\textsuperscript{2}, Philip Carns\textsuperscript{2}, Robert Ross\textsuperscript{2}, Yong Chen\textsuperscript{1}

\textsuperscript{1}Department of Computer Science, Texas Tech University,
\textsuperscript{2}Mathematic and Computer Science Division, Argonne National Lab

---

**Abstract**

- **Objectives**
  - To meet the Quality-of-Service (QoS) requirements for HPC platforms
  - To propose a flexible and effective storage system using software-defined approach to assure a certain level of resources per application

- **Background**
  - The ability to guarantee a certain level of performance is called QoS support
  - As larger storage systems are using, multiple applications which are sharing the same storage systems, will compete and interfere with each other
  - Application interference dramatically degrades the overall system performance

- **Contribution**
  - We proposed a flexible solution to achieve soft QoS storage guarantees using software-defined approach
  - We proposed borrow model and policies to meet QoS requirements

---

**Proposed Architecture**

- We propose to use software-defined technique as Figure 1 shows
- Two key software-defined components are added into HPC storage system to enable a flexible QoS provisioning
  - **Data plane**: It is running on each storage server for IO classification and bandwidth shaping
    - Data plane contains multiple queues, each of which buffers requests from a given application.
    - IO classification is done based on the IO header
  - **Control plane**: The control plane consists of the several components
    - **Token Rate Generator**: It communicates with the Desired QoS component in data plane to sync the requested bandwidth specification of each application and generates a token rate per application
    - **Virtual Token Buckets**: It learns the token rate from token rate generator to operate
    - **Traffic Shaper**: It communicates with the virtual token buckets to get information to shape the traffic
    - **Policy Enforcer**: It is used to deliver policies to meet the QoS requirements

---

**Proposed Solution**

- **Proposed Software-Defined Solution**
  - Based on the software-defined architecture, we propose several key improvements:
    - We introduce borrowing model which allows a queue of one application to borrow tokens from queues of the same application in other storage servers.
    - We extend the original M-LWDF algorithm in conjunction with the borrowing model. The extended M-LWDF is used to guarantee the fairness during degradation
    - We design a set of policies regarding the borrowing model, including whether the borrow can happen and when and how many tokens should be borrowed.

---

**References**


