Using Control Theory to Reduce Disk Congestion Caused by Unpredictable I/O in Cloud Computing

Per3S Workshop - Thomas Collignon

28/05/2024

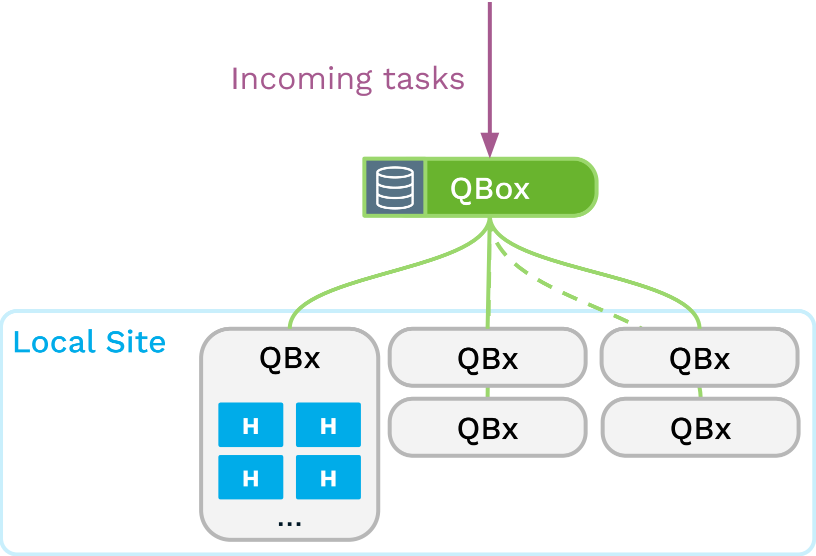
QARNOT

Inria

ADEME
AGENCE DE LA TRANSITION ÉCOLOGIQUE

PULSE
Pushing Low-carbon Services towards the Edge

Qarnot infrastructure



QBox

- Task distribution
- Download of tasks' input data
- Upload of results
- Shared storage for the tasks

I/O congestion on the QBox’s storage

Background tasks

- Data traffic
  - Downloads
  - Uploads
- Task Checkpointing
- Cache handling

Computing tasks

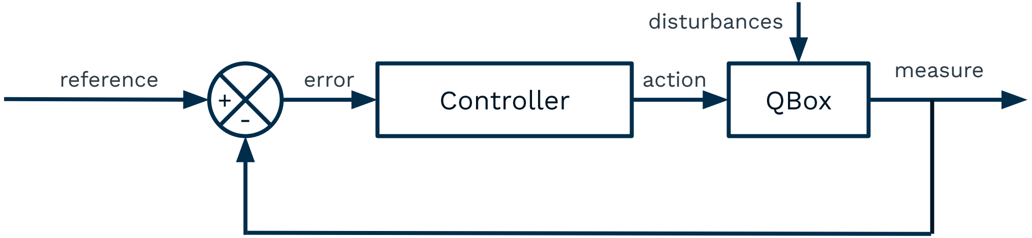
- Varied I/O profiles

The performance of Computing Tasks can be degraded by I/Os interferences with other tasks.

How to improve the performances of computing tasks by controlling disk I/Os ?

Control Theory

- Autonomic Computing
- Actions on the system at runtime
- Supports disturbances (new computing tasks ...)



The I/O problem is inherently hard to predict so Control Theory is a good candidate to solve it at runtime.

Control strategies

Actuators for the selected problems :

Data traffic

- Bandwidth
- Delay

Cache Handling

- Garbage collector
- Cache strategy

Checkpointing

- Delay, but user constraints

Actuator for all the tasks:

cgroups

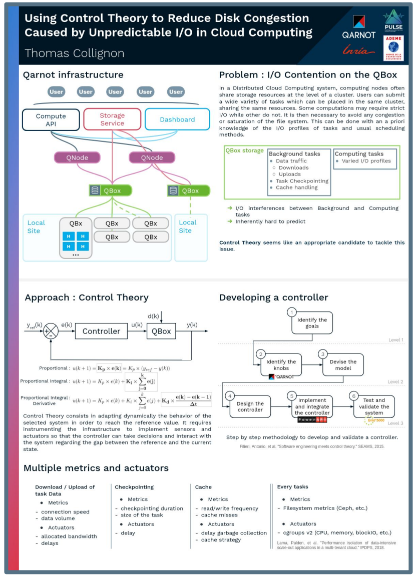
- Memory
- I/O bandwidth
- CPU

Using Control Theory to Reduce Disk Congestion Caused by Unpredictable I/O in Cloud Computing

Thomas Collignon

Thank you for your attention,

Let’s discuss at the poster session !



Using Control Theory to Reduce Disk Congestion Caused by Unpredictable I/O in Cloud Computing

Thomas Collignon

Qarnot infrastructure

Problem : I/O Contention on the QBox

In a Distributed Cloud Computing system, computing nodes often share storage resources at the level of a cluster. Users can submit a wide variety of tasks which can be placed in the same cluster, sharing the same resources. Some computations may require strict I/O while other do not. It is then necessary to avoid any congestion or saturation of the file system. This can be done with an a priori knowledge of the I/O profiles of tasks and usual scheduling methods.

QBox storage

Background tasks
- Data traffic
  - Downloads
  - Uploads
- Task Checkpointing
- Cache handling

Computing tasks
- Varied I/O profiles

→ I/O interferences between Background and Computing tasks

→ Inherently hard to predict

Control Theory seems like an appropriate candidate to tackle this issue.

Approach : Control Theory

Proportional : $u(k+1) = \boxed{\mathbf{K_p} \times \mathbf{e(k)}} = K_p \times (y_{ref} - y(k))$

Proportional Integral : $u(k+1) = K_p \times e(k) + \boxed{\mathbf{K_i} \times \sum_{\mathbf{j=0}}^{\mathbf{k}} \mathbf{e(j)}}$

Proportional Integral : Derivative $u(k+1) = K_p \times e(k) + K_i \times \sum_{j=0}^{k} e(j) + \boxed{\mathbf{K_d} \times \frac{\mathbf{e(k) - e(k-1)}}{\mathbf{\Delta t}}}$

Control Theory consists in adapting dynamically the behavior of the selected system in order to reach the reference value. It requires instrumenting the infrastructure to implement sensors and actuators so that the controller can take decisions and interact with the system regarding the gap between the reference and the current state.

Developing a controller

Step by step methodology to develop and validate a controller.

Filieri, Antonio, et al. "Software engineering meets control theory." SEAMS, 2015.

Multiple metrics and actuators

Download / Upload of task Data
- Metrics
- connection speed
- data volume
- Actuators
- allocated bandwidth
- delays

Checkpointing
- Metrics
- checkpointing duration
- size of the task
- Actuators
- delay

Cache
- Metrics
- read/write frequency
- cache misses
- Actuators
- delay garbage collection
- cache strategy

Every tasks
- Metrics
- Filesystem metrics (Ceph, etc.)
- Actuators
- cgroups v2 (CPU, memory, blockIO, etc.)

Lama, Palden, et al. "Performance isolation of data-intensive scale-out applications in a multi-tenant cloud." IPDPS, 2018.