QoS for Cloud Computing

PIREGRID THEMATIC DAY
May, 10thn 2011
University of Pau

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France
What is Quality of Service?

- Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance.

- QoS criteria are numerous and is highly dependant of the application...
  - Throughput, Delay, jitter, Loss rate

- ... Or of the end-user
  - Image resolution, sound quality, appropriate language, ...
Common Service Specification

- **Loss**: probability that a flow’s data is lost
- **Delay**: time it takes a packet’s flow to get from source to destination
- **Delay jitter**: maximum difference between the delays experienced by two packets of the flow
- **Bandwidth**: maximum rate at which the source can send data
- **QoS spectrum**:

Best Effort

Guaranteed
QoS for Cloud

The shared cloud assumption

Many users, with various profile and different needs!

Concept: MWD Advisors, www.mwdadvisors.com
**Application’s Profile**

- **Traditional scientific computing applications**
- **Business/financial applications & simulations**
- **Interactive applications with user feedback**
- **Urgent computing for public safety, disaster relief**
- **Environmental and data processing/analysis**

**Time scale**
- Bounded in time
- Sporadic
- Continuous

**Data injection rate**
- Assuming that received data must be processed
- Defines the minimum data injection rate that is required by the application

**Time scale**
- Bounded in time
- Sporadic
- Continuous

**Data injection rate**
- Assuming that received data must be processed
- Defines the minimum data injection rate that is required by the application
Urgent disaster relief!

Time scale
Continuous
Data injection rate
Min=25 fps
How to provide QoS?

Many ways to provide QoS
- Scheduling, admission control, traffic control, dynamic resource provisioning, ...

- How to take into account the various application’s profiles?
- How to protect users from misbehaving applications?
- How to handle urgent demands?

Regulate (adapt & control) the data injection rate into the computing resources
QoS: general picture

Network bandwidth is assumed to NOT BEING the bottleneck
Traffic and Service Characterization

- Definitions
  - Cloud Infrastructure (CI)
  - Processing Units (PU)

- To quantify a service one has two know
  - Flow’s traffic arrival
  - Service provided by the CI, i.e., resources reserved at PU

- Regulation will be done by an envelope process, borrowed & adapted from the network community

- Ideas is to
  - Bound the data injection rate to...
  - ...isolate users from each others and...
  - ...to provide QoS enforcement at flow level
Traffic Envelope (Arrival Curve)

- Maximum amount of service that a flow can request during an interval of time $t$

$$b(t) = \text{Envelope}$$

- Slope = max average rate
- Slope = peak rate

"Burstiness Constraint"
Traffic Envelope (Traffic Shaping)

Traffic are variable by nature, must take into account burstiness constraints.

Use an envelope process to bound the data injection rate while allowing for variable, bursty traffic.

Arrival curve
**Ex: Token Bucket (1)**

- Characterized by three parameters: $(b, R, C)$
  - $b$ – Token depth
  - $R$ – Average arrival rate
  - $C$ – Maximum data injection rate

- A bit is transmitted only when there is an available token
- When a bit is transmitted exactly one token is consumed

![Diagram showing token bucket regulation](image)

- Rate: $R$ tokens per second
- Tokens: $b$ tokens
- Capacity: $\leq C$ bps
- $b \cdot C / (C - R)$

**Graphical Representation**

- **bits** vs. **time**
- **slope C**
- **slope R**
Token Bucket (2)

Example

- $B = 4000$ bits, $R = 1$ Mbps, $C = 10$ Mbps
- Packet length = 1000 bits
- Assume the bucket is initially full and a “large” burst of packets arrives

![Diagram of token bucket with variables B, R, C = 10 Mbps, and a burst of packets arriving.](istoica@cs.cmu.edu)
Token Bucket (3)

time = 0

\[ \text{time} = 0 \]

\[ \text{time} = 0.3 \text{ ms} \]

\[ \text{time} = 0.1 \text{ ms} \]

\[ \text{time} = 1 \text{ ms} \]

\[ \text{time} = 2 \text{ ms} \]

\[ \text{time} = 3 \text{ ms} \]

\[ \text{time} = 1 \text{ ms} \]

\[ \text{time} = 0.7 \text{ ms} \]

\[ \text{time} = 1 \text{ ms} \]

\[ \text{time} = 1 \text{ ms} \]
ARRIVAL CURVE

A(t) – number of bits received up to time t

istoica@cs.cmu.edu
Per-hop Reservation with Token Bucket

- Given b, r, R and per-hop delay d
- Allocate bandwidth $r_A$ and buffer space $B_A$ such that to guarantee $d$

![Diagram](image)
The QoS measures (delay, throughput, loss, cost) depend on offered traffic, and possibly other external processes.

A service model attempts to characterize the relationship between offered traffic, delivered traffic, and possibly other external processes.
Arrival and Departure Process

$R_{in}(t)$ = arrival process
= amount of data arriving up to time t

$R_{out}(t)$ = departure process
= amount of data departing up to time t

buffer
delay
Delay and Buffer Bounds

$E(t) = \text{Envelope}$

$S(t) = \text{service curve}$

Maximum delay

Maximum buffer
Token Bucket support in workflows (1)

Support of superscalar pipeline models

Work done with R. Tolosana, J. Banares and O. Rana
Token Bucket support in workflows (2)

TB QoS is introduced seamlessly into workflow specifications with the Renew tools.

Work done with R. Tolosana, J. Banares and O. Rana
Conclusions

- Clouds will be shared clouds driven by economical constraints.
- For some applications, availability of resources and isolation are of prime importance (urgent computing).
- QoS for clouds is already a necessary and hot topic in research community.
Perspectives

- Add more parameters to the TB model
  - Excess burst size
  - Advanced mark vs. drop policy
- Dynamic configuration of TB parameters at each stage of the processing path
- Take into account data inflation behaviors
- Generalized the usage of envelope processes, comparison,...