Management and Content Delivery for Smart Networks: Algorithms and Modeling

L2: Simulating a content distribution system

The objective of this laboratory is to practice simulation methodologies. You will execute the essential steps to remove simulation warm-up effects and estimate confidence intervals for simulated quantities.

Summary

You will simulate a content distribution system. Assume you are responsible for a service that is very popular around the world.

You have two design choices:

- Deploy a *fixed number of servers* worldwide, which remain always on-line. Management of the infra-structure is easy, but it may cost too much if servers are over-provisioned, or force users to reach a far away datacenter, thus increasing service delay;
- *Commission/decommission servers* nearby users following increases/decreases in workloads. The goal in this case is to increase users' satisfaction while minimizing costs. The former is obtained by reducing the service delay, while the latter is obtained by deploying a minimum number of servers to handle the workload.

The simulator considers the aspects described in the following.

Clients

- Clients arrive to the system and make a sequence of *requests*. Requests come back-to-back – i.e., as soon as a request is fully served, the client fires the next one, until the client leaves the system.
- Each client's session is composed of K requests, with K being a discrete random variable uniformly distributed between 10 and 100.
- Clients come from around the world. To make it simple, assume that, on average, a small percentage of users from the top-n countries in terms of Internet users[1] visit the service on a daily basis. The more Internet users in the country, the more arrivals from there you have.
- Clients arrive according to a Poisson process with rate defined as above. However, the arrival changes throughout the day, following typical Internet patterns: It grows substantially during the day and reduces substantially during the night. Suggestion: Divide the day in some slots, changing the arrival rate for each slot.[2]
- Clients are routed to the geographically closest server that is on-line in the arrival instant. If multiple servers are available in the same location, create a reasonable policy to allocate the new client.

Servers

- Each server is connected to the network through a single link of limited capacity, e.g., 10 Gbps. Capacity is shared among all active *requests*, thus the capacity allocated to serve each active request is updated every time a new request arrives or a request is fully served.

[1] `https://bit.ly/1MRwlzK`
[2] Traffic at Polito during a couple of days: `https://bit.ly/2YmcAwa`

- Each client's requests results in the download of a variable number of bytes. The size is described by a random variable. Get a reasonable distribution for it (e.g., object sizes of popular websites).
- The server can only handle a maximum number of concurrent requests (MaxReq). Requests that cannot be served are rejected. Clients retry immediately after a fail. If a request is rejected, the client keeps retrying until it succeeds.
- Servers can be deployed in some few locations. Having a server online costs per time unit. Get realistic parameters from cloud providers.[3]
- When simulating the dynamic server allocation alternative, whenever a server's workload (number of active requests) is bellow a threshold, the server stops receiving new requests, completes the pending ones and goes to sleep. A minimum number of servers is however always on-line.
- Again in that scenario, if a server workload gets close to MaxReq, it triggers the wake-up of a new server, which becomes available immediately to handle new requests. It is your task to decide where to deploy the new server.

Network

The network model will be extremely simplified to keep the simulation simple. The time to complete a request is the sum of:

- A small server latency, coming from a discrete random variable uniformly distributed between 1 ms and 10 ms.
- A round-trip-time (RTT) from the client to the server. Assume that RTT is equal to the time the light takes to travel from the capital city of the country where the client is located to the city where server is hosted. Since these times are constant, make a lookup table for the simulation.
- Transfer delay, which is determined by the size of the response divided by the capacity allocated to the request in the server. Note that the transfer delay changes with the number of active requests sharing the server.

Nothing else influences the download time, i.e, no packet loss, no cross-traffic in the network etc.

Your task

Evaluate the proposed alternatives by comparing the time to satisfy all requests of each client (client session time) and the cost of servers per day in different simulation scenarios.

Try to parameterize the simulator as realistically as possible. Show confidences intervals and pay attention to the simulation warm-up.

Groups and Final Reporting

You are expected to work on groups of up to three students. Each group is required to prepare a **single** report describing results of **all labs in the course**. This report must not exceed 10 pages.

You need to delivery both the written report and your source code before the end of exam session in September.

References

[1] SimPy in 10 Minutes. `https://simpy.readthedocs.io/en/latest/simpy_intro/`
[2] matplotlib. `http://matplotlib.org/`

[3] For example: `https://aws.amazon.com/about-aws/global-infrastructure/`