19/10/2015
2015

# Pac1

SISTEMES DISTRIBUITS AULA 2

MANUEL MARTÍNEZ DÍAZ

1. Explain how web architectures solve the problems of scalability, concurrency and failure handling.

The major problem of scalability is to avoid performance bottleneck, through buying more physical resources to replicate data and also using decentralized algorithms. Another problem is when the system is running out of software resources, for example the IP address available is running out, and is needed to change the IP protocol to have more IP addresses.

The other problem is concurrency, the access a data by more than one user, is necessary to synchronize the access to the data to maintain its consistency, using techniques such as locking data until the transactions is finished or aborted, another technique is adding a timestamp on each transaction and synchronized all the other servers that contains the same replicated data or also an optimistic concurrency control that a distributed transaction is validated by an independent servers.

And the third problem is Failure handling that is managed by the next ways;

- Detecting failures using timeouts, but only indicates that is not responding during a time, may it have crashed or may be slow.
- Masking failures hidden errors, like when data fails to arrive or send a corrupt one then automatically the data is retransmitted. The corrupt data is detected using the checksum and even, in some cases, used to reconstruct to original data.
- Tolerating failures, when a server is not reachable the web browser informs you that an error occurs and leave up to you if you want to try again or leave it.
- With redundancy of routes between two routers to achieve a destination or the replication of data for example an entire server or the DNS table that are replicated in at least two servers.

2. Present real examples of thin client and fat client distributed system models.

An example of thin client is application and desktop virtualization like Citrix XenDesktop, where the computer only has an operative system and the applications and data remains on the server.

As a fat client distributed system model we have for example peer-to-peer computing like torrent or any other where each computer can collect and serve files.

3. Explain the differences between two tiered and three tiered architectures.

In a two tiered architectures, the presentation, the application and the data logic is partitioned into two processes, the client and the server.

In a three tiered architectures, each one of the parts mentioned above are in a different server.

The two tiered architectures have less latency than a three tiered architecture, only exchanging one message to invoke an operation. Also the three architectures is more difficult to manage because has one more server and more network traffic between them and more latency because the extra messages necessary to invoke and operation.

4. Identify the major concepts of chapters 1 and 2 (Coulouris book) presented in the following article:

   Above the Clouds: A Berkeley View of Cloud Computing. http://ftp.cs.duke.edu/courses/cps296.4/compsci590.4/fall13/838-CloudPapers/AboveTheClouds.pdf

**Cloud computing,** a term to define a set of Internet-based applications, storage and computing services. This services use physical resources and software services.

**Scalability,** the system will remain effective when there is an important increase. The cloud can add more resources to avoid performance bottleneck.

- Control cost of physical resources, a big datacenter can purchase hardware, network bandwidth and power with big discounts for the quantity.

**Software services (SaaS),** are the applications delivered as a services in a cloud computing.

Explain when it is worth to use a public cloud with formula (1).

$$\text{UserHours}_{\text{cloud}} \times (\text{revenue} - \text{Cost}_{\text{cloud}}) \geq \text{UserHours}_{\text{datacenter}} \times (\text{revenue} - \frac{\text{Cost}_{\text{datacenter}}}{\text{Utilization}}) \quad (1)$$

It depends directly by the Utilization, when the value is 1 the utilization is 100% and the only have to compare is the cloud cost with the datacenter cost.

Without using real prices in the formula, it is obvious to conclude that the use of a public cloud is worth it when the utilization varies in time, for example a web site that has a spike of visits only on holidays.

5. Explain the relevance of Content Delivery Networtks (CDN) for the scalability of Internet.

**Relevance of CDN for scalability,** the scalability in a system is the ability to handle any amount of data request without any perceptible performance loss. CDN is a system of computers networked together across the Internet to cooperate transparently for delivering content by replicating the content on-demand when users request for it or replicated beforehand, by replicating to the surrogate servers (mirrored servers).

CDN replicating the data only when is needed gives us a way to handle loads peaks and reduce the cost compared to an over-provisioning solution.

Describe the life cycle of a typical request passing through a CDN. See:

http://www.cloudbus.org/reports/CDN-taxonomy.pdf

**Life cycle of a request Life,**

1. A user sends a request by specifying the URL of content provider's web site in a web browser.
2. The origin server of the content provider that receive the client request returns the basic index page to the user and also redirects it to the CDN provider.
3. The CDN provider manage the client request using an adaptive or a non-adaptive algorithm using metrics such as network proximity, client perceived latency, distance, and replica server load to route the request to the best surrogate server.
4. The client is informed the surrogate server trough a request-routing mechanism as Global Server Load Balancing (GSLB) , DNS-based request-routing, HTTP redirection, URL rewriting, anycasting and CDN peering.
5. The selected surrogate server serves the requested data.