# _____Z-blog_____

O.R.Manoj Niranthaka. IT17048088

FRIDAY, MARCH 8, 2019

## Distributed systems

### "DISTRIBUTED SYSTEMS"

A distributed system consists of a collection of autonomous computers, connected through a network and distribution middle ware, whi computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integra facility.

### "DISTRIBUTED COMPUTING"

Distributed computing is where multiple computing units are connected to achieve a common task. The larger computing power enables tasks to be performed than in a single unit, and searches can be coordinated for efficiency. Successes usually give the finder credit.

Distributed computing projects include hunting large prime number, and analysing DNA codes.

---

### STANDALONE SYSTEM

*all the components are executed within a single device.
*Do not need a network.
*Usually one or tightly coupled set of technologies are used to develop
(JAVA,.NET)

### DISTRIBUTED SYSTEM

*The components are distributed and executed in multiple devices.
*Need a Network.
*Multiple and looselycoupled set of technologies are used to develop
(HTML+CSS+JS+PHP)

---

### ELEMENTS OF DISTRIBUTED SYSTEMS

- Processing components
- Data networksfor components to communicate
- Including the components who are dedicated for processing the communication, called connectors
- Data stores (data bases) and Data
- Theconfiguration of the above elements

---

DIFFERENT TYPES OF SERVICES

- Mail service (SMTP, POP3, IMAP)
- File transferring and sharing (FTP)
- Remote logging (telnet)
- Games and multimedia (RTP, SIP,H.26x)
- Web (HTTP)

---

BROWSER-BASED

Pluging Based
Standed (HTML+CSS+JS+PHP)

NON-BROWSER-BASED

Standard Desktop Applicaion Components
Mobile App
IoT Devices

---

There are mainly two types of web services.

1.SOAP web services.
2.RESTful web services.

SOAP (Simple Object Access Protocol)

SOAP is known as a transport-independent messaging protocol. SOAP is based on transferring XML data as SOAP Messages. Each messag something which is known as an XML document. Only the structure of the XML document follows a specific pattern, but not the content of Web services and SOAP is that its all sent via HTTP, which is the standard web protocol.

Here is what a SOAP message consists of

*Each SOAP document needs to have a root element known as the <Envelope> element. The root element is the first element in an XML
*The "envelope" is in turn divided into 2 parts. The first is the header, and the next is the body.
*The header contains the routing data which is basically the information which tells the XML document to which client it needs to be se
*The body will contain the actual message.
The diagram below shows a simple example of the communication via SOAP

---

## DIFFERENT ARCHITECTURES FOR DISTRIBUTED SYSTEMS

Client/server architecture

Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to the client. This type of architecture has one or more client computers connected to a central server over a network or internet connecti shares computing resources.

Client/server architecture is also known as a networking computing model or client/server network because all the requests and servic over a network.

3-tier architecture

A 3-tier architecture is a type of software architecture which is composed of three "tiers" or "layers" of logical computing. They are oft applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development envi modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allo update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other p system.

For example, the user interface of a web application could be redeveloped or modernized without affecting the underlying functional bu access logic underneath. This architectural system is often ideal for embedding and integrating 3rd party software into an existing app integration flexibility also makes it ideal for embedding analytics software into pre-existing applications and is often used by embedde vendors for this reason. 3-tier architectures are often used in cloud or on-premises based applications as well as in software-as-a-serv applications

N-tier architecture

N-tier architecture is a client-server architecture concept in software engineering where the presentation, processing and data manage are both logically and physically separated. These functions are each running on a separate machine or separate clusters so that each i the services at top capacity since there is no resource sharing. This separation makes managing each separately easier since doing worl not affect the others, isolating any problems that might occur.

Service-oriented architecture

Service-oriented architecture (SOA) is a software development model for distributed application components that incorporates discover data mapping and security features.
SOA has two major functions. The first is to create a broad architectural model that defines the goals of applications and the approache meet those goals. The second function is to define specific implementation specifications, usually linked to the formal Web Services Des Language (WSDL) and Simple Object Access Protocol (SOAP) specifications.

---

MVC for web-based systems and their strengths and weaknesses

Advantages of MVC

1) Faster development process: MVC supports rapid and parallel development. With MVC, one programmer can work on the view while ot the controller to create business logic of the web application. The application developed using MVC can be three times faster than appli developed using other development patterns.

2) Ability to provide multiple views: In the MVC Model, you can create multiple views for a model. Code duplication is very limited in MV separates data and business logic from the display.

3) Support for asynchronous technique: MVC also supports asynchronous technique, which helps developers to develop an application th fast.

4) Modification does not affect the entire model: Modification does not affect the entire model because model part does not depend on Therefore, any changes in the Model will not affect the entire architecture.

5) MVC model returns the data without formatting: MVC pattern returns data without applying any formatting so the same components and called for use with any interface.

6) SEO friendly Development platform: Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a s application.

Disadvantages of MVC

1) Increased complexity

2) Inefficiency of data access in view

3) Difficulty of using MVC with modern user interface.

4) Need multiple programmers

5) Knowledge on multiple technologies is required.

**6) Developer have knowledge of client side code and html code.**

---

Posted by ManojNiranthaka at 6:31 AM

No comments:

Post a Comment

Enter your comment...

Comment as: ManojNiranthal ▾

Publish Preview

## Home

Subscribe to: Post Comments (Atom)

ABOUT ME

ManojNiranthaka

View my complete profile

BLOG ARCHIVE

▼ 2019 (4)

▼ March (1)

Distributed systems

► February (3)