2015s1

COMP9321

E-commerce Infrastructure Implementation

Sample Questions Answer

By null

Notice: these answers may not be the standard answer and only from my point of view and experience. I only give these answers for possible reference. There is **NOT** any guarantee of the correctness of the contents within this document.

Sample Questions: Web Fundamentals

Short Answer Questions

1.How does HTTP version 1.1 support persistent connections?

Answer: Keep-alive

2.Why is the POST method preferred for implementing forms for gathering user authentication input (e.g. login)?

Answer: Input will not be presented in the address bar of web browser when being submitted to the server by POST method. If by GET method, the input can be revealed in the query string part of URL in the address bar.

3.A user is trying to log into a web application which has its authentication information stored in a database. Unfortunately, the database is momentarily disconnected at the time of the request and the user's authentication fails. What is the HTTP error code that should be returned to the user and why?

Answer: 500 Internal Server Error. Database is located at the server side, The disconnection of database is regarded as a server software internal failure.

Medium Length Answer Questions

1.List and discuss the basic building blocks of the web:

Answer: URL (Uniform Resource Locator to address/locate resources)
HTTP (Hyper-Text Transfer Protocol, request-response, based on TCP, connections to transfer resources)
HTML (Hyper-Text Mark Language, documents to present resources)

2.Using an example, describe the Uniform Resource Locator scheme as applied to Hypertext Transfer Protocol

Answer:

http://www.example.com:8080/document.html?userToken=1v4Tug82Pl7af&type=noPassword

[Scheme] http://
[Domain name of the host] www.example.com
[Port] :8080
[Path] /document.html
[Query string {key1=value1&key2=value2}]
?userToken=1v4Tug82Pl7af&type=noPassword

3.What are the steps in a typical HTTP interaction when the client is making a request to the server for the first time?

Answer:

→ Client sends a request that contains target URL to DNS server (first time)
→ DNS server queries URL and returns IP address and return to client
→ Set up a TCP connection between client and server (3-way handshake)
→ Client sends an HTTP request
→ Server receives the request and sends an HTTP response, and then closes the connection after that.

4.What are the disadvantages of the HTTP scheme for fetching Web resources?

Answer:

a. HTTP request can only fetch one resource at a time in one connection.
b. HTTP Headers are repeated for every request and sent without compression.

c. HTTP/1.1 is unidirectional, which means it can only be initiated by the client side. Even if the server knows that a client is lacking a particular resource, it has to wait for the client's request, otherwise it can do nothing.

5.Explain how input is passed from the client to the server for a GET request.

Answer: By putting the data into the query string part of a URL of a GET request in a form of key-value pair connected by ampersand (&) between each other, we can pass the input data from the client to the server. Server can just retrieve the data by identifying the content of query string of the URL.

Long Answer Questions

1. Consider the form provided below:

```
   <p>Please register with our site.Type in your contact email and password
(4 characters long).</p>
   <FORM action="registerServlet" method="POST">
      <INPUT TYPE="text" NAME="email">
      <INPUT TYPE="password" NAME="password">
      <INPUT TYPE="hidden" NAME="isNew" VALUE="true">
     <INPUT TYPE="submit" VALUE="Register">
   </FORM>
```

Assuming that the user filled in the form and pressed the Register button, explain how the input from the user is passed to the server-side using the POST request method. Write the code for processing the request in registerServlet.

Answer: Client sends the input as a block of data to the server by POST method to server. → Server gets the POST request and get the parameters by their names.

```
// though I prefer use RegisterServlet here…
class registerServlet extends HttpServlet {
   protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
      doPost(request, response); // only use POST method for this
   }

   protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws IOException {
      String emailString = request.getParameter("email");
      String passwordString = request.getParameter("password");
      String isNewString = request.getParameter("isNew");

      // do the other business logic…
      // response with the result of registration…
   }
}
```

2. Discuss in detail the 3-tier model for architecting Web applications.

Answer: Presentation Tier:

Present the resources to the client by JSPs and other pages based on HTML. Expose the functions of the application to clients.

Business Logic Tier:

Process user input. Performs logical evaluations & calculations, and Decides which view to expose to users.

Data Access Tier:

Interfaces with the database, stores, updates and maintains persistence within the application.

3. Explain, with a diagram, the typical layers in a Web application architecture.

Answer: Presentation Layer – Business Logic Layer – Data Access Layer – Data Storage Layer (database)

(Difference between **tiers** and **layers**
http://stackoverflow.com/questions/120438/whats-the-difference-between-layers-and-tiers)

4. Explain the difference between HTTP GET and POST methods. Why shouldn't POST requests and responses be cached?

Answer:

Difference between GET and POST: (Chapter 2 slides)

POST can transfer a block of data, which GET cannot.

POST is generally used to submit data from client to server for processing, while GET to retrieve data from server by query strings.

POST cannot be bookmarked, use GET instead if necessary.

When dealing with sensitive data (e.g. user password, payment information, etc.), use POST instead of GET because GET will expose these data in the address bar of the browser.

GET can be cached, but POST is never cached.

Why POST cannot be cached: (Chapter 12 slides)

HTTP POST changes the state of a resource on the server and therefore should not be cached. Therefore, result of a POST should only be HTTP 200 or 204.

Details: http://www.w3schools.com/tags/ref_httpmethods.asp

Sample Questions: Java Servlets

Short Answer Questions

1.With reference to the control flow in a web application, describe the difference between using response.sendRedirect(URL) method and the forward() method of the Request Dispatcher interface.

Answer:

response.sendRedirect(URL): client URL U1 → to servlet A → A response with a new URL U2 with status code 301 Redirecting → client use U2 to access servlet B → servlet B handle this request and respond to client

requestDispatcher.forward(): client URL U → to servlet A → A generate a new request with U to servlet B → B handle the request and respond to client

[requestDispatcher.include(): client URL U → to servlet A → A generate a new request with U to servlet B → B handle this request and respond to **servlet A** (this is the main difference from forward) → servlet A uses this response to respond to client]

2.You have been asked to develop a web application using Java Servlets. One of the requirements is to add an HTML block at the bottom of every page. This HTML block resides in a file called WEB-INF/footer.html. Using code, describe how you would meet this requirement.

Answer:

```
Public class SomeServlet extends HttpServlet {
   Protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
      doPost(request, response);
   }

   protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws IOException {
      RequestDispatcher dispatcher =
request.getRequestDispatcher("footer.html");
      dispatcher.include(request, response); // include footer.html
   }
}
```

Or, in JSP way:

```
<jsp:include page="/footer.html">
```

3.Explain the difference between include() and forward()?

Answer: Please see Short Question 1.

4.Using an example, explain the need for the forward() method?

Answer: E.g. in a typical MVC web application, there may be a controller which only dispatches requests to different business logic servlets. In this case, the dispatcher controller servlet only use forward() method to forward requests to business logics.

Medium Length Answer Questions

1.If the client's browser disallows setting of cookies, then what are the possible options to perform session tracking?

Answer:

URL rewriting by appending JSESSIONID to the URL.

Use HTML hidden fields to store the session information so that the information will not display in the page directly.

2.What are the advantages of using cookies over URL Rewriting and hidden fields in a form?

Answer:

Cookies can be kept in the browser's memory, or can be written to a file for future references, while others cannot.

The contents of cookies cannot be seen directly from HTML documents or address bars of browsers, while others can.

Long Answer Questions

1.Why is the stateless nature of HTTP a problem in Web application development? You must illustrate your answer with an example.

Answer: E.g. If we want to implement a function of user login, without any session or other things' support, after we send a login request from the browser to the login servlet, we can get a result of login success/failed. If we login successfully and we go to the login page again, it still gives a login page without any sign of already logged in, because after the login procedure, the HTTP request and response of login no longer exist. We cannot know if we have logged in because of statelessness of HTTP. (Basically in this case we can use a session to keep the record of login state of a user.)

2.Discuss the lifecycle of a Servlet in the Web application container. When does a Servlet gain control over HttpServletRequest and HttpServletResponse objects?

Answer: load servlet class → instantiate servlet → servlet init() method to initialize servlet (only once in lifecycle of servlet) → servlet service() method (doGet, doPost, doXX…, used to handle requests) → servlet destroy() method (only once in lifecycle of servlet to destroy the servlet when the web application container shuts down).

When a client send an HTTP request, the container creates HttpServletRequest and HttpServletResponse objects and pass the control to the requested servlet.

(http://stackoverflow.com/questions/3106452/how-do-servlets-work-instantiation-session-variables-and-multithreading)

3.You have been asked to develop a web application for selling books. One of the features requested of this application is to have a wish list associated with each customer. The wish list contains books that the customer is interested in but may not be ready to buy this time. Clients can add and remove books from the wish list at any time. After the client exits from the Web site, the wish list is still associated with the client. When the client comes to visit the Web site again, the wish list with its contents from her last visit will be presented to the client again. The wish list feature should work without requiring the customers to log in.

How would you implement the wish list feature? Describe your solution in as much detail as possible, but you do not need to write actual code.

Answer: (only provide a sketched one) The wish list can be a serializable object created by a servlet and stored in both the database at the server side and a cookie at the client side. After a user modify the contents of a wish list, the changes will be written into the above places. For performance reasons we can save the wish list only in two situations, when the user logs out of the site or closes current window of the browser. By the above implementation, on the one hand we can save the wish list in not only client side and also server side so that the client can check its wish list without logging in (by keep the cookies of the browser enabled), on the other hand we keep a copy of user's wish list at server side so that in case the user disables cookies, the application can also get the correct wish list for the user after it logs in.

4.Explain the lifecycle of a Session object inside a Servlet container, starting from the moment the user's request reaches the container. How does the container associate a Session object with a user?

Answer: For each user, the container creates an HttpSession object to be associated with that user. An HttpSession object relies on a cookie or URL rewriting to send a token to the client. The token is usually a unique number called the session identifier (JSESSIONID). This session identifier is used to associate a user with a Session object in the server. The session will be terminated in the following three cases: 1) user logged out (which can be implemented by calling invalidate() to destroy the session immediately) 2) time out 3) the application goes down.

Sample Questions: Java Server Pages

Short Answer Questions

Discuss why JSP scriptlets should NOT be used in a web application.

Answer: Scriptlets will mix up the presentation layer and business logic of a web application. If we put too many scriptlets in JSP, when we want to modify the business logic we must change JSPs as well, which is not a good structure of a web application.

Medium Length Answer Questions

1.Describe the requirements for defining an object as a JavaBean. Explain the role played by JavaBeans in the 3-tier architecture for Java Web applications.

Answer: A JavaBean requires:

a) To be defined as an implementation of Java interface Serializable;

b) All member variables private, each of which has a pair of getter and setter methods.

2.Using examples, discuss how the JSP Expression Language exploits the notion of "Convention over Configuration".

Answer:

Convention over Configuration (also called “coding by convention”) means that when we define JavaBeans, we use the same or similar names to access these beans at different layers of a web application. By doing so we can focus more on the data content itself instead of thinking about names to use to access the data. E.g. in EL of JSP, we can use ${BeanName.variableName} or ${BeanName[“variableName”]} to get a variable of a JavaBean, which is implemented in JavaBean as a variable name with a pair of getVariable() and setVariable() methods.

Answer:

Long Answer Questions

(Nothing here)

Sample Questions: XML

Short Answer Questions

1.Consider the following portions of an XML document.

```
<freeText> This web site services both <italic>ITB262</italic> and
<italic>ITN262</italic>. The material covered will be the same for
<bold>both</bold> units.</freeText>
```

Describe a DTD for this XML fragment.

Answer:

```
<!ELEMENT freeText(#PCDATA|italic|bold)*>
```

2.Using examples, discuss the need for namespaces in XML documents.

Answer: E.g.

```
<books>
   <book>
      <name>Hans Mong</name>
   </book>
   <author>
      <name>Bibliography of Hans Mong</name>
   </author>
</books>
```

When we need to distinguish these two name element between book and author, we use a namespace for one of them to specify that this “name” is a name of a book (if it works on element book) or an author (if it works on element author).

Medium Length Answer Questions

1.Consider the following XML file describing items in a computer hardware store:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE item SYSTEM "item.dtd" >
<list>
   <item>
     <name>Keyboard</name>
     <description>Microsoft Comfort Curve Keyboard</description>
     <price currency="USD">20.00</price>
   </item>
   <item>
     <name>Mouse</name>
     <description>Logitech Mouse</description>
     <price currency="AUD">10.00</price>
   </item>
</list>
```

There has to be at least one <item> element in the list to begin with. Also, "currency" is a required attribute for the <price> element.

Write a DTD for this XML file.

Answer:

```
<?xml version=”1.0”?>
<!DOCUMENT list [
<!ELEMENT list (item+)>
<!ELEMENT item (name, description, price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ATTLIST price currency (USD|AUD) #REQUIRED>
]>
```

2.Given an XML document with the following DTD.

```
<?xml version="1.0" ?>
<!DOCTYPE item [
<!ELEMENT item (name, description, price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ATTLIST price
   currency (USD|AUD) #REQUIRED >
]>
```

Write the equivalent XML Schema Definition (XSD).

Answer:

```
<simpleType name=”currencyString”>
   <restriction base=”string”>
      <enumeration value=”USD”/>
      <enumeration value=”AUD”/>
   </restriction>
</simpleType>

<complexType name=”priceType”>
   <element name=”price” type=”decimal”/>
   <simpleContent>
      <extension base=”string”>
         <attribute name=”currency” type=”currencyString”/>
      </extension>
   </simpleContent>
</complexType>

<complexType name=”itemType”>
   <sequence>
      <element name=”name” type=”string”/>
      <element name=”description” type=”string”/>
      <element name=”price” type=”priceType”/>
   </sequence>
</complexType>

<element name=”item” type=”itemType”/>
```

Long Answer Questions

1.For the example displaying an XML for a computer hardware store, write a program using SAX that will parse the given XML and produce the following output:

```
Keyboard USD 20.00
Mouse AUD 10.00
```

2.For the same example, write a program using DOM to produce a report on the total price (in AUD) of the items in the document.

3.Consider an XML file with the following DTD, that describes a list of books :

```
<!ELEMENT bibliography (book)+>
<!ELEMENT book (author, title, publisher, address, year, isbn, price)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

Assuming that you are using DOM API to parse the XML file, write a function to print the author name and the title text for each book. You only need to use pseudo-code.

4.In the previous example, imagine that the DTD describes the XML file that serves as the database for a bookstore Web application that is designed as per MVC pattern. For this application, write the Data Access Object implementation that returns a list of BookBean objects, each containing author name, title and price for that book.

Sample Questions: Design Patterns

Medium Length Answer Questions

1.What are the advantages of employing Dependency Injection in a Web application architecture.

Answer:

1)Loose coupling of different parts of a web application - avoid monolithic architecture.

2)Provide modularity wherein one service or a component can be replaced with another implementation at deployment time.

3)Provide strong typing for different services and reduce the amount of string-based lookups by replacing them with declarative Java annotations.

4)Provide a central location for modifying configuration data.

5)Enable unit testing of different components and easy mocking of different services.

2.Discuss how the Model2 architecture for Web applications enforces the principle of "Separation of Concerns".

Answer: (Model1 stands for JSP architectures without any controller, Model2 for MVC pattern)

Model2 (MVC) architecture:

1)Separate data from view and use a **controller** to handle application flow between components

2)View is used to present the data it is given Model represents data and some business logic

3)Model represents data and some business logic

4)Controller is used to handle client requests and invoke the appropriate business logic

3.Contrast (对比) the roles of Servlets and JSP under the Model-View-Controller design paradigm. When is it acceptable to use <jsp:forward> ?

Answer:

In MVC design paradigm, JSPs are mostly used to present the resources and interact with users as View parts, while servlets to either handle the input data by their business logics (Service) or manage the redirect and/or forward of different views after receiving a variety of inputs.

One possible situation for using <jsp:forward> action may occur in Model1 (JSP architecture), which some of JSPs are used to forward requests.

Long Answer Questions

1.In a Web application that uses the MVC pattern (e.g., think of your own project implementation), how do you guarantee that all client connections to your application must go through the controller Servlet, after the client has successfully logged in? That is, the client cannot directly go to any of the JSP pages by entering the location of a JSP page in the browser. Describe your solution to this problem. Note that you do not need to write code.

Answer:

1) Put all the JSP pages into WEB-INF folder under the root directory of

application. This prevents JSP from being accessed by URL typed in the browser.

2) Use front-controller design (command design pattern) and add all the possible jumps between pages into the controller with all the commands mapped to the pages.

3) Define servlet mapping in web.xml and give a specific servlet mapping URL pattern, so that all the invalid URL from any client will be responded a 404 not found status.

2.Explain the "Service Locator" design pattern with the example of abstracting data sources in web applications.

Answer:

In a pattern of service locator, we have all the data source access methods encapsulated and abstracted in several services and expose some interface to clients. There will also be some service factory classes which create the objects of services. A service locator is used to look up and get services for clients.

3.Consider a web application whose content can only be accessed after the user passes authentication. How do you guarantee that the client cannot directly go to any of the JSP pages in the application by entering the location of a JSP page in the browser ? Describe the solutions to this problem.

Answer: the same to Long Question 1.

4.Explain with an example, how the Command Design Pattern can be used to separate request processing in a web application from its core business logic.

Answer: This can be identified from the procedure of handling a client request under a Command Design Pattern application. When a client request with a specific URL comes, it first access the controller. The controller will determine the command to execute by the URL of the request, then execute that command. This specific command will call several business logic methods to handle the user's input and get results and the page to forward for the controller. Finally the controller forward to the page with those results. In this model, JSPs are only used to present data and interact with users. The controller only determine which command to execute to satisfy the request and which page to return. Commands perform the methods provided by the services and other business logic components. Whenever we need to change the business logic, we can achieve it by only modifying the code of services with the counterparts of controllers remained, which separates the request processing part from business logic part.

5.Discuss Model-View-Controller architecture pattern in detail. What are the advantages of using MVC over an architecture based solely on JSP pages ?

Answer:

1)At the center is the controller servlet (dispatcher), which is the single entry point of the application.

2)Presentation parts (JSP pages) are isolated from each other.

3)MVC separates content generation and content presentation.

4)Model 2 is more flexible and easier to maintain, and to extend, because views do not reference each other directly.

6.Describe methods for avoiding duplicate form submissions from a browser in a web application ?

Answer:

1) POST redirect GET. POST request to business logic → process and HTTP 301 redirect → generate a new request and use GET to access the redirect logic → do nothing but output the result page. In this case when the page refresh the client will generate a new GET request, which will be immediately handled by redirect logic.

2) Synchronizer token. Request comes → server compare token values with previous requests

if equal → treat as the same request → outgoing response

if not → process request → generate a token for this request → store the token in the persistence layer → prepare and outgoing response. In this case when a request comes with the same token value, it will be directly responded as the same request.

These two ways can prevent duplicate submission, which will not access to the business logics.

7.You are about to build a simple Web application that manages student records for a course using the J2EE framework. The data is stored in a Students table. The core functionality of the Web application is limited to

listing students,

adding a student,

deleting student(s), and

updating a student.

Using diagrams and pseudo-code, discuss the implementation of the Data Access Object pattern for this application.

Answer:

DAO: provides method for business logic to interact with database.

In this case interface StudentDAO contains at least the following methods: List<Student> findAll(); void insert(Student student); void delete(Student student); void update(Student student);

Its implementation is a class which implements StudentDAO (say StudentDAOImpl): public class StudentDAOImpl implements StudentDAO {…}

For this kind of DAO we also need a student JavaBean to transfer the data of corresponding database table:

```
Public class Student {
   // student information as private

   // getters and setters of above information
}
```

This JavaBean Student is implemented based on its corresponding table student, whose tuples (or records) have the same pattern of information as Student Bean.

8.For the previous example, discuss the implementation of a Front Controller for the application.

Answer:

In this case it is appropriate to use Command Design Pattern.

```
Public interface GeneralCommand {
   Public void execute(HttpServletRequest request, HttpServletResponse
response);
}
```

We create four classes AddStudentCmd, UpdateStudentCmd, DeleteStudentCmd and ListStudentCmd, whose execute() method calls StudentDAO.insert(), StudentDAO.update(), StudentDAO.delete(), StudentDAO.findAll().

Interface GeneralController is used to define a controller, which has the basic method executeCommand().

Class StudentController implements GeneralController. This class stores internally a HashMap of request URL patterns and their corresponding command names. It implements executeCommand by searching URL pattern in the HashMap above to find its Command class and performing Command.execute().

Sample Questions: Security

Short Answer Questions

1.Give an example of how third-party cookies can be used to violate a user's privacy ?

Answer: This can happen when a third party cookie stores login credentials of a user. The attacker can get access to the cookies by sending the user a URL which contains malicious request for a target site.

Medium Length Answer Questions

1.Explain the steps in setting up an HTTPS (HTTP Secure) connection between a browser and a Web server. How are HTTPS connections vulnerable ?

Answer:

… compute a session key when handshaking with HTTP.

HTTPS will be vulnerable when the private key used for encryption/decryption leak from the client side, or the public key is exposed from the server side.

Long Answer Questions

1.Using an example, discuss how lack of input validation can lead to a web application becoming vulnerable to Cross-Site Scripting (XSS) attacks. Does using POST over GET reduce this vulnerability ?

Answer: XSS happens when the web application lacks validation of front end input. Because it can happens at the front end, there is nothing can do by using POST over GET.

2.Explain SQL injection attacks with an example. How can they be prevented ?

Answer:

E.g. id=' or '1'='1.

```
String id = request.getParameter("id"); // ‘[space]or[space]’1’=’1
Statement stmt = con.createStatement("select * from TBL_USERS" + "where
id ='"+ id +’”);
```

In this example the SQL query will finally become the following:

Select * from TBL_USERS where id=’{’ or ‘1’=’1}’

⇔ select * from TBL_USERS where id=’’ or ‘1’=’1’. The condition after ‘or’ will be always true so this injection will select all the things from TBL_USERS.

Prevention:

1)Parameterized SQL statements: i.e., PreparedStatements Parameter values are quoted. This preserves intent of the query.

2)Use Stored Procedures: (but implement this carefully).

3)Escape user-input: All user-supplied input should be escaped (e.g. using double quotes).

4)Whitelisting (Positive filtering): Specify a set of characters that are allowed and everything else is rejected.

5)Privilege Settings: Give least privilege to your application (only DB reads, writes only where required, use non-admin accounts).

3.Consider the code fragment below for a form to handle posting of comments to a forum:

comments.jsp:

```
<form action=ControlServlet method="post">
<textarea name="comments" cols="40" rows="5">
</textarea><br>
<input type="submit" value="Post" />
ControlServlet.java

protected void doPost(HttpServletRequest request, HttpServletResponse
response){
try{
    String commentString = request.getParameter("comments");
   DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");
   stmnt = connection.createStatement();
   String sqlStr =
   "INSERT INTO TBL_COMMENTS (COMMENT_DATE, COMMENT) VALUES ("
      +"'"+fmt.format(new Date())+"',"
      +"'"+commentString+"')";
   int result = stmnt.executeUpdate(sqlStr);
   stmnt.close();
}
```

What are the security problems exposed by this code ? How can this code be modified to avoid these problems ?

Answer:

1) SQL Injection
2) XSS

Modify:

```
String commentString = request.getParameter("comments");
// code for removing the <,>,&… characters in commentString
String sqlStr = "INSERT ……";
stmnt = connection.prepareStatement(sqlStr);
……
```

Sample Questions: Performance & Metrics

Medium Length Answer Questions

1.Describe what Performance is, and explain why it is important?

Answer:

Performance is the time taken to complete a unit operation on a Web application.

2.Describe what Scalability is, and explain why it is important?

Answer:

Scalability is to keep performance constant even with increase in the number of users and requests.

Why:

1)Wide Audience - Millions of users, users are not only humans but also other web applications which regularly poll your application for data.

2)Interactive - User queries drive the application logic on the server. Users also usbmit information via parameters that need to be processed

3)Dynamic - Data generated is dependent on user input/session generated. This means that pages cannot be generated in advance. Result of interactivity but not the same.

4)Always On - Applications are always available on the Internet - there's no shutdown or Ctrl-Al-Del. If a popular site goes down, it becomes news and impacts the perception of reliability.

5)Integrated - Applications depend on other applications. If one goes down, a chain of applications may fail (e.g. credit-card processing applications)

Long Answer Questions

1.Describe the steps involved in performance engineering in Web Applications?

Answer:

1)Defining performance goals and target system

2)Selecting appropriate metrics and factors affecting performance

3)Creating a model for the workload and the outcomes

4)Designing tests to evaluate the actual performance

5)Analysing the results of the tests.

6)Repeating the above steps.

2.Identify and explain 5 types of performance metrics?

Answer:

1) Response time - the time it takes a system to react to a human request. Also called Round-trip Time (RTT)

2) Throughput - The rate at which requests are completed from a computer system is called throughput

3) Availability - fraction of time the system is up and available to its customers - - also called its uptime.

4) Reliability - is the probability that a system is going to function properly and continously in a fixed period of time

5) Resource Utilization - The amount of resources consumed by an application, expressed as a percentage of resources available

3.Explain how Little's Law is useful for measure performance of Web

Applications?

Answer:

Little's law can be used not only to examine any "black box" and holds for subsystems as well, but to derive other operational laws for performance analysis (e.g. General Response Time Analysis, Bottleneck Analysis, etc.)

4.State Interactive Response Time Law. Explain how it can be used for measuring the performance of a web application ?

Answer:

An interactive system where N users generate requests that are serviced by one resource and the responses come back to the users. After a think time Z, the users submit the next request.

Therefore, total cycle time for requests from one user is R + Z .

In time period τ , a user generates about τ /(R + Z) requests. Therefore, system throughput

X = N(τ/(R+Z))/τ = N/(R+Z)

R = (N/X)−Z.

We can get R/Z from this law.

1)A ridiculously large R may suggest a performance problem.

2)R>Z indicates the response time cannot fulfill the users' request and may cause a conjunction at server side.

3)Large Z indicates that the application has either a lot of continuous tasks which need users to wait till they are completed, or a high cognitive load which makes users confused and not know what to do next.

(Please see: http://blog.sina.com.cn/s/blog_6716d7bf0100wv4o.html)

5.State the General Response Time Law. How do we use this law to perform bottleneck analysis ?

Answer:

For a balanced system (arrival rate equals departure rate), the total number of requests in the system

Q0 = Q1 + Q2 + . . . + QM

where M is the total number of resources in the system. From Little's Law, it follows that

X0.R0 =X1.R1 +X2.R2 +...+XM.RM

Dividing both sides of the equation by X0 and using the forced flow law, we get

R0 =R1.V1 +R2.V2 +...+RM.VM

If the general response time of accessing a resource is large, it may indicate that accessing this resource is a bottleneck for the whole application.