RELATIONSHIP BETWEEN WEB SERVICE AND GRID SERVICE

The last decade has noted a substantial change in the ways global industries, businesses, and home users apply computing devices, including a wide variety of ubiquitous computing resources and advanced Web services. Initially, the focus was on localized computing resources and respective services; however, the capabilities have changed over time and we are now in an environment consisting of sophisticated, virtualized, and widely distributed Business On Demand utility services.

Major global organizations, solutions providers, service providers, and technology innovators, whom we have already discussed in the earlier chapters (and even those we have not yet discussed), have absolutely contributed to this technology evolution. In the previous chapter, we explored several grid architecture points, and the architectural relationships between grid and other distributed computing technologies. As we can see from this discussion and the following illustration, the evolution of Grid Computing is progressing at a very rapid rate. This computing evolution is tightly aligned with the incredible and very rapid evolution of the Internet and other open standards architectures.

PREVIEW

4.1Relationship between Web Service and Grid Service

Throughout this chapter, we have been discussing Web services, and the respective technical underpinnings of Web services. The basic premise of this architecture is the creation of interoperable services and their respective applications.

Readers are by now aware that Grid Computing is the process of resource sharing among a collection of participants : This involves interoperable access to sharable resources. The architectural evolution of Grid Computing selects Web services as the technology for defining these interoperable resources. The main criteria on this selection are the open protocol base and the interoperable messaging solutions, as proposed by the Web services architecture. We have already discussed the evolution of grid services and its adaptability to the emerging technologies. In the next section, we will explore the details on how grid services are defined around Web services architectures, and the standardization process of grid services.

Given the previous discussion, we can now spend some time in exploring the relation between Web services and grid services, how we can differentiate each of them, and in what situations they share similarities.

An application or service may have the ability to maintain a state, and that state may be pertaining to the user of that application. For example, a purchase order systemkeeps a user's order information between the interaction, the user , and the system (i.e., verify order, change order, update order, etc.), until the order is submitted for delivery. This state information (i.e., purchase order) may be local to the applicationor the service, or it may be stored in an external state machine(s) such as databases, other resources, local session state, and the resource/service object.

It is noteworthy to understand how the above state is managed in Web service scenarios. Generally speaking, we can classify service state management into two forms. These two forms are as follows:

1. Interaction aware state. Normally, in the world of the Web and Web services, a client may interact with a service for a long period of time, as we have discussed in the above purchase order case. These interactions are correlated using some information passed from the client to the service, along with the message. This can be a simple cookie, a session ID, or complex correlation information. The advantage of this architecture design point is that the server side is not managing any specific client state information, or creating a specific instance for a client. The server-side implementation is very scalable and stateless in nature. It is not preventing the real state being persisted in a state machine external to the application; instead, this is correlated to a specific client's state using the session ID. This is, typically, how normal Web and Web services are defined.
2. Application aware state. In these situations, services are aware of its client and create a specific instance of the service for the specific client, and pass that instance information (e.g., primary key) back to the client for interaction. At this stage, the client is holding a reference to the specific instance of the service/application, and hence, can interact with the service instance without passing any correlation information. These services are typically referred to as stateful services, because the state information is held in the

service itself and not passed back to the client. One important item to notice about this state management, similar to the above case, is that the service need not sustain the state; rather, the service may delegate the state to other state machines. The only requirement is that the client owns a reference to the service instance.

Source : http://elearningatria.files.wordpress.com/2013/10/ise-viii-grid-computing-06is845-notes.pdf