HPSearch for Managing Distributed Services

Authors: Harshawardhan Gadgil (hgadgil@cs.indiana.edu), Geoffrey Fox (gcf@indiana.edu), Shrideep Pallickara (spallick@indiana.edu)

KeyWords: Scripting, Management, Distributed Services

We introduce HPSearch [1, 2] as a scripting based management console to control the deployment of application components (brokers, in our case). The HPSearch script can also be used to deploy the dataflow applications that allow us to process data in a streaming fashion in a service-oriented architecture. We shift focus to allow another capability to the management console, related to the deploying and management of distributed service handlers.

Motivation:

SOAP (Simple Object Access Protocol) has become the de facto standard for communication between distributed services in the architecture provided by Web Services. The Web Service suite of specifications provides different aspects of Quality of Service (QoS) for communication between these Web Services. Thus for instance, the `WS-Security` specification provides *quality of protection* through message integrity, message confidentiality, and single message authentication while providing general purpose mechanism for associating security tokens with messages, `WS-Policy` defines syntax and policies of a Web Service, `WS-Addressing` provides transport neutral mechanisms to address resources (Web Services and messages) while `WS-Reliable Messaging` defines mechanisms for reliable transfer of messages between distributed applications in the presence of software component, system or network failures. This QoS is typically provided by adding information corresponding to the specification in the header of the SOAP message.

Currently, the Web Service community has been moving from the traditional RPC (Remote Procedure Call) based SOAP messaging to a more document oriented SOAP messaging. This implies a heavy request-response or one-way messaging between endpoints to meet QoS goals. For instance, while using `WS-Reliable Messaging`, the SOAP endpoints might need to exchange messages corresponding to acknowledgements or retransmissions of received messages.

When a SOAP message is received by an endpoint, it must have specific handlers (plugins) to process each header corresponding to each `Web Service` specification. SOAP processing toolkits such as Apache Axis [3] provide the means for defining handlers to process a SOAP message. Axis 1's design was oriented more towards RPC-style, synchronous, request-response interactions. Hence, it is difficult to address the arbitrary message exchange patterns between endpoints that is observed while satisfying different QoS constraints. Apache Axis 2 architecture [4] proposes to address this issue by re-designing and re-implementing the handler-chain architecture as a pipeline model.

We now introduce NaradaBrokering [5, 6, 7] which is a scalable, reliable publish-subscribe event delivery middleware. Recently NaradaBrokering has been augmented with a SOAP processing stack so that applications residing in different hosting environments (C++ based gSOAP, .NET based WSE, Perl-based SOAP::Lite) can interact with the substrate and also leverage all QoS provided to the NaradaBrokering clients. Ref. [8] describes this architecture in more detail.

We plan on developing handler applications as parts of service endpoints to provide an extensible architecture. A group of handlers would form a handler pipeline. Each handler would help provide different capabilities such as encryption, compression, logging, reliable delivery and security among others and would operate directly on the SOAP message.

Proposed Scripting Architecture:

We are investigating the use of HPSearch as a scripting interface to aid in the deployment and configuration of the handler pipelines. HPSearch is currently used to deploy brokers and manage links between brokers. HPSearch provides a custom Mozilla Rhino Javascript host-object (`NaradaBroker`) that helps create broker instances and manage links between brokers (Ref: `http://www.hpsearch.org/documents/notes/07-scripting/NaradaBroker.html`).

We discuss below, a possible scenario with respect to managing handler pipelines and configuring them. The architecture is presented in Figure 1. The figure shows a broker network and a blown up view of a single broker. The Broker exposes a `servlet` container. When a SOAP message arrives, the `servlet` determines the series of handlers the received message must pass through, depending on the roles specified in the header of the SOAP message.

As the number of brokers in a broker network grows, we face the problem of deployment and management of handlers at remote brokers. To address this issue, we propose to add custom host-object, `Handler,` to the suite of HPSearch objects. The `Handler` object would allow us to remotely manage the handler chain at a given broker. This involves, instantiating the handlers, querying the brokers for deployed handlers and re-configuring the handler chains at run-time.

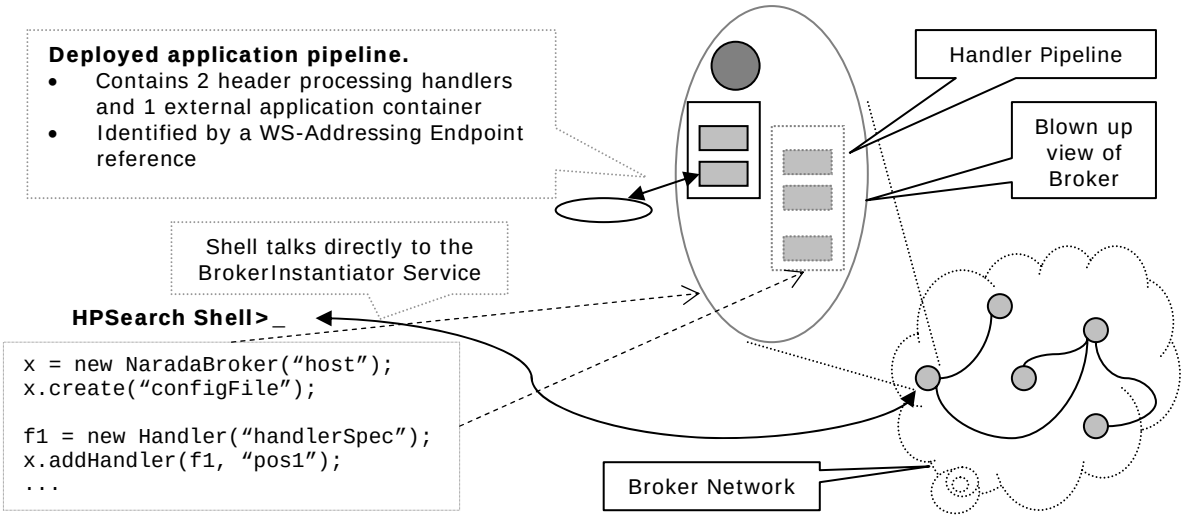

Figure 1: HPSearch for managing Distributed Service Handlers

Apache Axis defines handlers to facilitate custom processing of a SOAP request. We use a similar idea here by replacing Axis with NaradaBrokering as the queue manager for SOAP messages. Each service handler may be combined in a pipeline to expose a more complete message processing application.

Another advantage of using scripting would be to configure and deploy remote service applications. An application would consist of a series of handlers (such as logging, security among others) and an external service. We introduce a relay handler that acts as a bridge between the handler pipeline and the external service. After a message has been processed through the handler chain, it is forwarded to the external service. The reply is sent from the service to the relay handler, then through the handler pipeline and finally forwarded to the intended recipient.

References:

1. Harshawardhan Gadgil, Geoffrey Fox, Shrideep Pallickara, Marlon Pierce, Robert Granat, *"A Scripting based Architecture for Management of Streams and Services in Real-time Grid Applications"* (To appear) Proceedings of the IEEE/ACM Cluster Computing and Grid 2005 Conference (CCGrid 2005). Cardiff, UK.
2. HPSearch Project page: http://www.hpsearch.org
3. Apache Axis project: http://ws.apache.org/axis
4. Apache Axis 2 Project: http://www.apache.org/axis2
5. Geoffrey Fox, Shrideep Pallickara and Savas Parastatidis, *"Towards Flexible Messaging for SOAP Based Services"* Proceedings of the IEEE/ACM Supercomputing Conference 2004, Pittsburgh, PA
6. Shrideep Pallickara, Geoffrey Fox *"NaradaBrokering: A Distributed Middleware Framework and Architecture for enabling Durable Peer-To-Peer Grids"* Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware - 2003, Rio Janerio, Brazil June 2003
7. NaradaBrokering, Project Page: http://www.naradabrokering.org

8. Geoffrey Fox, Shrideep Pallickara, Marlon Pierce, Harshawardhan Gadgil, *"Building Messaging Substrates for Web and Grid Applications"* (To Appear) In special Issue on Scientific Applications of Grid Computing in Philosophical Transactions of the Royal Society, London, 2005