Towards Flexible Messaging for SOAP Based Services

Geoffrey Fox, Shrideep Pallickara
Community Grids Lab, Indiana University. USA.

Savas Parastatidis
School of Computing Science, University of Newcastle. UK.

http://www.naradabrokering.org
NaradaBrokering

- Virtualizes communication transport and endpoints
  - UDP, TCP, Parallel TCP, Multicast, SSL …..
- Based on a distributed network of cooperating broker nodes. (brokers support software overlay network)
- Plays the same role for Grid as MPI does in parallel processing
- Efficiently routes (content or endpoint-based) information from producers to consumers of content.
  - Subscriptions can be based on SQL, Regular expressions and XPath queries.
- JMS compliant and provides support for routing JXTA (P2P) interactions.
- Been deployed and tested in the context of multimedia conferencing and Grid applications.
- Introduces delays of order one to two milliseconds at each broker
Mean transit delay for raw message samples in NaradaBrokering: Different communication hops

Transit Delay (Milliseconds)

Message Payload Size (Bytes)

- hop-2
- hop-3
- hop-5
- hop-7

Pentium-3, 1GHz, 256 MB RAM, 100 Mbps LAN, JRE 1.3 Linux
Standard Deviation for message samples in NaradaBrokering
Different communication hops - Internal Machines
XGSP Web Service MCU Architecture

Use Multiple Media servers to scale to many codecs and many versions of audio/video mixing

- Session Server
  - XGSP-based Control

- Media Servers
  - Filters
  - Web Services
  - High Performance (RTP)
  - and XML/SOAP and ..

- NaradaBrokering
  - All Messaging
  - NB Scales as distributed

Gateways convert to uniform XGSP Messaging

- Admire
- SIP
- H323
- Access Grid
- Native XGSP

NaradaBrokering
Average Video Delays for one broker – divide by N for N load balanced brokers

Latency ms

Multiple sessions

One session

30 frames/sec

# Receivers
Reliable Delivery guarantees

- NB reliable delivery guarantees holds true under the following conditions
  - **Broker and Link Failures**: The entire broker network may fail. Guarantees are met even if just one broker recovers.
  - **Prolonged Endpoint disconnects**
  - **Stable Storage Failures**
    - Stores need to recover after failures.
  - **Unpredictable Links**
    - Events can be lost, duplicated or re-ordered
- **Supports various ordering/delivery guarantees.**
Transit delays/Standard deviations in a 3 broker network. NB-BestEffort(BE)(TCP) Vs NB-ReliableDelivery(RD)(UDP)

Mean delay (NBRD-UDP)
Mean delay (NBBE-TCP)
Std Dev (NBRD-UDP)
Std Dev (NBBE-TCP)
Support for large payloads

- NB incorporates services for **compressing** (and decompressing) large payloads.
- Also incorporates services for **fragmenting** large payloads into smaller fragments, and subsequently coalesce them into a large payload.
  - This capability along with NB’s reliable delivery capability is used to support fault-tolerant asynchronous GridFTP.
- Integrating NaradaBrokering with **BitTorrent**
  - **GridTorrent** is Web/Grid Service compliant
NB-enhanced GridFTP

Adds Reliability and Web Service Interfaces to GridFTP
Preserves parallel TCP performance and offers choice of transport and Firewall penetration
Forthcoming Features


- **Active replay** support: Pause and Replay of live streams.
  - Applying to real time annotation (**e-Sports**)

- **Replicated storage support** for fault tolerance and resiliency to storage failures.

- **Federation** of the rival **WS specifications** in the area of notification and reliable delivery.

- **Scripting Interface** to administer NaradaBrokering
  - Workflow runtime
Deployment of NaradaBrokering in Web and Grid Services

- Support the Service Internet
- Support the Service Context and Information Environment
- Deploy as Handlers/Filters in Service Infrastructure
- Deploy as Proxies in virtual Containers
- “Bind” SOAP to NaradaBrokering and get high performance or special QoS
  - A/V Streams
  - PDA Streams
  - HPC Streams

NaradaBrokering Supports Streams -- sets of messages – as basic concept
Layered Architecture for Web Services and Grids

- **Higher Level Services**
- **Service Context**
- **Service Internet**

**Base Hosting Environment**

**Protocol** HTTP FTP DNS ...

**Presentation** XDR ...

**Session** SSH ...

**Transport** TCP UDP ...

**Network** IP ...

**Data Link / Physical**

**Application Specific Grids**

- Generally Useful Services and Grids
- Workflow WSFL/BPEL

**Service Management** (“Context etc.”)

**Service Discovery (UDDI) / Information**

**Service Internet Transport → Protocol**

**Service Interfaces WSDL**
IOI and CIE in Service Infrastructure

- Identify two layers: IOI (Service Internet On the Bit Internet) and CIE (Service Context and Information Environment)

- IOI is most “straightforward” as it is providing reasonably well understood capabilities at a new “level” e.g.
  - WS-RM provides TCP like reliability at message level
  - WS-Transfer provides HTTP like capabilities
  - WS-Addressing replaces IP Header
  - WS-SecureConversation replaces SSL

- CIE includes inter-service “shared memory” used to manage and control context messages at “distributed operating system level (WS-Context a nice approach to this)
  - WS-Eventing, Notification can be thought of at this layer
Structure of SOAP

- SOAP defines a very obvious message structure with a **header** and a **body**

- The **header** contains information used by the “Internet operating system”
  - Destination, Source, Routing, Context, Sequence Number …

- The **message body** is partly further information used by the operating system and partly information for application when it is not looked at by “operating system” except to encrypt, compress it etc.
  - Note WS-Security supports separate encryption for different parts of a document

- Much discussion in field revolves around what is referenced in header
Deployment Issues for “System Services”

- “System Services” (handlers/filters) are ones that act before the real application logic of a service.
- They gobble up part of the SOAP header identified by the namespace they care about and possibly part or all of the SOAP body.
  - e.g. the XML elements in header from the WS-RM namespace.
- They return a modified SOAP header and body to next handler in chain.

```
Header
  ────────────
  |            |
  | WS-RM      |
  | Handler    |
  |            |
  |            |
  |            |
  | WS-.........|
  | Handler    |
  |            |
  |            |
  |            |
  |            |
  |            |
  |            |
  Body
```

E.g. ……. Could be WS-Eventing WS-Transfer …. 
NaradaBrokering as a Handler

- Refactor original broker into handlers which can talk either to brokers or other handlers
  - Similar to P2P deployment of NB

![Diagram showing the refactoring of NaradaBrokering into handlers.]

- Publisher
- Broker
- Subscriber

- Original Service Appl. Logic
- NB as Handler
- NB as Service Handler
- Service Appl. Logic
Handlers/Filters in-memory Processing

Built-in Handlers (replaced if use NB Substrate)
Handler(Filter) approach: Advantages

- **Entails no changes to the service endpoints:** this facilitates incremental addition of capabilities transparently to “application” (real service)
- Filters can be developed and tested independent of the service endpoints thus providing greater robustness.
- Promotes code reuse since different filters (part of container not service) corresponding to security, compressions, logging or timestamps etc. can be utilized by multiple services.
Problems with JAX-RPC Handlers

- Java Web Service Endpoint standard
- Handlers are statically pre-configured; it is not possible to dynamically configure the handler-chain.
  - This implies that for every message the order in which the handlers process them is fixed. Furthermore, every message traverses every handler.
- It is not clear if these handlers can in turn connect to other services.
  - If this is allowed, it is not clear if messages (requests, responses and faults) issued by such a handler needs to traverse the handler-chain associated with the original service.
- Difference between handler and service not clear
  - Both are (SOAP) message based functionalities
Proxy Distributed Processing

• A handler is like an in memory “service” so one can build handlers that can alternatively be deployed “outside” application service and look like a service.
  – Natural for some cases like Reliable Messaging but always possible as SOAP Intermediary

• Support and architecture of handlers/services that can be inside or outside containers is not clear?
  – Build handlers to work in conventional or virtual (distributed) containers and use workflow to link

---

WS-RM Service

WS-RM enabled SOAP  \rightarrow  Legacy Service

WS-RM removed from SOAP Header
Other NaradaBrokering-based Handlers

• We are building handlers WS-RM, WS-Reliability, WS-Eventing, WS-Notification and their federation for UK e-Science program.

• Additional NB filters/handlers can incorporate functionality for heart-beat, network proximity, performance monitoring.

• NaradaBrokering will support replicated subscribers
  – NB can utilize the usage, performance, liveness and proximity metrics to locate nearest-least-utilized service (or resource) instance.

• Similar ideas can work with .NET(WSE Filters), SOAP::Lite and gSOAP (plugins)
SOAP capabilities for NaradaBrokering

• NaradaBrokering can supply handlers that operate on traditional SOAP over HTTP Messages to give reliability, notification etc.

• NB can function as a SOAP intermediary. Based on actor (SOAP1.1) or role (SOAP1.2) attributes.

• One can bind SOAP to NaradaBrokering which can exploit different transports and different (non XML) representations
  – NB becomes a transport handler (the last handler) that includes support for the SOAP processing model as can’t use Axis default
  – Enables Web/Grid Services to interact directly with the NB substrate
Permeating service endpoints

Messages traverse handler chain prior to delivery to substrate and after receipt from substrate.
Transport of SOAP messages

Transports supported: TCP, UDP, SSL, HTTP, HTTPS, Multicast, Parallel TCP streams.

Depending on payload/application type, transports deployed could vary from SSL, HTTP, TCP or Parallel TCP streams.

Transport Interfaces
(Communication through proxies, NATs & firewall boundaries)

Data accumulated by Monitoring Service

Specific to a transport
High Performance Transport in NaradaBrokering

• Can set up a High Performance Conversation where traditional SOAP over HTTP used to negotiate stream which is then transported by SOAP NB binding
  – Note SOAP header of stream delivered in initial SOAP negotiation
  – Later messages in stream just transport changes in SOAP header (message number etc.)

• Initial SOAP header and negotiation stored in CIE (WS-Context) so can always reconstruct full conventional SOAP message if needed

• Used in GlobalMMCS Audio/Video conferencing with special RTP NB transport
Supporting Service Discovery in CIE

- Services advertise themselves in WSDL
- Substrate support for SQL, XPath and Regular expression based queries can be leveraged to discover services.
- The advertise/discover operations can span different subsections of the distributed substrate.
  - The advertise/discover operations can be restricted to specific clusters.