

F9

November 19, 2004 11:15 AM

How GM TESTS WEB SERVICES

Frank Cohen
PushToTest

Frank Cohen

Frank Cohen is the "go to" guy when enterprises need to test and solve problems in complex interoperating information systems, especially Service Architectures and Web Services. Frank is CEO of PushToTest, a test automation solutions business and author of several books on testing information systems. For the past 20 years he lead some of the software industry's most successful products, including Norton Utilities for the Macintosh, Stacker, and SoftWindows. He began by writing operating systems for microcomputers, helping establish video games as an industry, helping establish the Norton Utilities franchise, leading Apple's efforts into middleware and Internet technologies, and most recently serving as principal architect for the Sun Community Server, Inclusion.net (OTC: IINC), and TuneUp.com. He serves as an active member and past board member of the Software Developers Forum, the leading computer software industry association in the Silicon Valley of California.

PushToTest

Title: How GM Tests Web Services

Presenter: Frank Cohen, Founder of PushToTest

Date: November 19, 2004

Abstract: General Motors is on the road to a Service Oriented Architecture (SOA) as its computing standard. To proceed they need to understand the scalability and throughput limits of the message-centric Web services approach that is the essence of SOA. General Motors chose SOA to build its next generation information systems, using Web services and ebXML technology. Service consumers and service providers will exchange Unified Business Language (UBL) Business Object Documents (BODs) to do things like get a purchase order and acknowledge an order. Frank Cohen presents the test methodology and findings from the GM Web Service Performance Benchmark project for its SOA applications and highlights a huge, impending industry-wide problem with Web services. Take home a test methodology to check SOA and Web Services for scalability in your environment.



How GM Tests Web Services

November 19, 2004

Frank Cohen

STARWest

fcohen@pushtotest.com

(408) 374-7426

Agenda

- SOAP RPC-Encoded does not scale
- SOA = BODs, Web Services, ebXML
- GM needs & the PushToTest solution
- Building the WSPB
- Findings
- Resources to watch

The “go-to” company

- Solve scalability problems
- Open-source TestMaker tool
- TestNetwork commercial tool
- Deliver custom test solutions
- Run scalability studies
- Train team: Dev, QA, IT

JAVATM TESTING and DESIGN

FROM UNIT TESTING TO AUTOMATED WEB TESTS

FRANK COHEN

The new book

Now Available

Prentice Hall

Chapters available
for free download.
thebook.pushtotest.com

Questions for you

- How many in your team/company?
- 1, <50, 50-200, 200-5000, 5000+
- Role of Development, QA, IT?
- Deliver function, optimize performance, QOS
- What kinds of systems?
- Web, Client/Server, Database, Portal, Desktop Application, Web-enabled Application, Integration
- Java, J2EE, Windows, .NET

SOAP RPC does not
scale.

Frank Cohen, 2002
IBM developerWorks

Find The Encoding

The screenshot displays the BEA WebLogic Workshop interface for configuring a web service named `responder_rpc.jws`. The **Project Tree** on the left shows the project structure, including `index.html`, `Lingo.java`, `responder_doc.jws`, and `responder_rpc.jws`. The **Design View** in the center shows a client calling the `respond` operation, with member variables `Lingo myLingo` listed below. The **Properties - responder_rpc** panel on the right shows the following settings:

- Name:** responder_rpc
- target-namespace:** namespace
- conversation-lifetime:** max-age 1 day, max-idle-time 0
- protocol:** form-post true, form-get false
- wsdl:** document
- schema:** file

A dropdown menu is open for the **soap-style** property, showing the following options:

- document
- rpc** (highlighted)
- default

The **Structure Pane** at the bottom left shows the `responder_rpc` service with the `respond(int wordcount, int delay)` operation and its context and myLingo variables.

At the bottom of the interface, the status bar indicates "Service Running", "Server Running", "Debugging", "Ln 1", "Col 1", and "INS".

Document-literal SOAP

- Visual Studio .NET generates SOAP proxy to expose methods as services

```
namespace myService
{
    public class myService: System.Web.Services.WebService
    {
        public HelloService()
        {}

        [WebMethod]
        public string greetingService
        (String strMyName)
        {
            return "Hello, " + name + ", \
and welcome to my service.";
        }
    }
}
```

Why not document-lit?

- RPC is easier for the developer

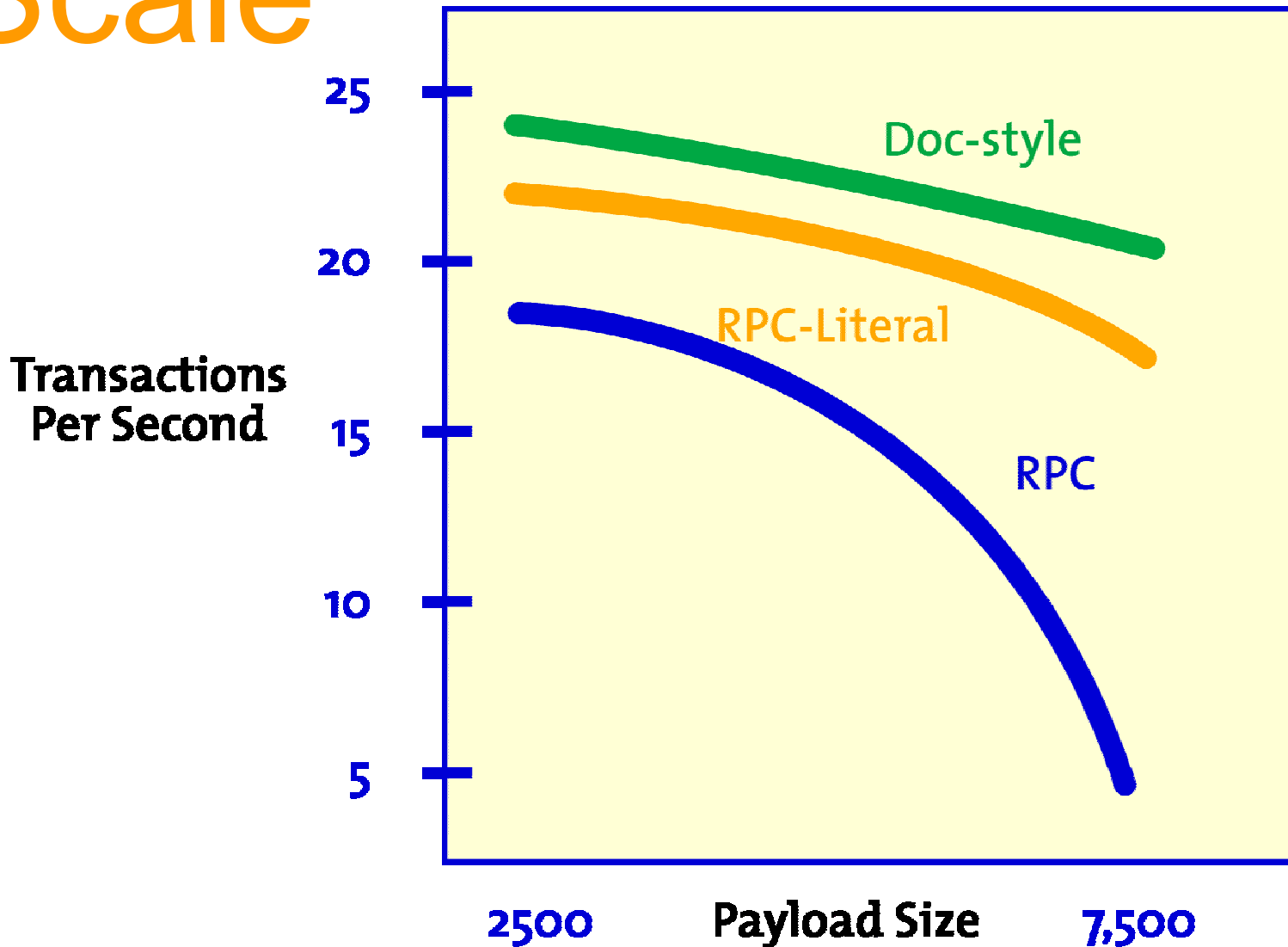
```
public class myComplexService{  
    public HashMap greetingService( firstname name ) {  
        HashMap ht = new HashMap();  
        ht.put( "Event", name );  
        ht.put( "Keycode", keycode );  
        ht.put( "PcAssemblyNumber", pcassembly );  
        ht.put( "SoftwareDottedString", dottedsw );  
        ht.put( "CurrentTime", new Date() );  
        ht.put( "CommandKey", "7" );  
        return ht;  
    }  
}
```

Document-literal

- Document-literal starts with a populated XML tree

```
public class myComplexService{  
    public void callGreetingService( firstname name ) {  
        body = new SOAPBody();  
        body.addElement ( "Event", name );  
        body.addElement ( "Keycode", keycode );  
        body.addElement ( "PcAssemblyNumber", pcassembly );  
        body.addElement ( "SoftwareDottedString",dottedsw);  
        body.addElement ( "CurrentTime", new Date() );  
        body.addElement ( "CommandKey", "" );  
        . . .  
        SOAPRequest.send( body );  
    }  
}
```

SOAP RPC Does Not Scale



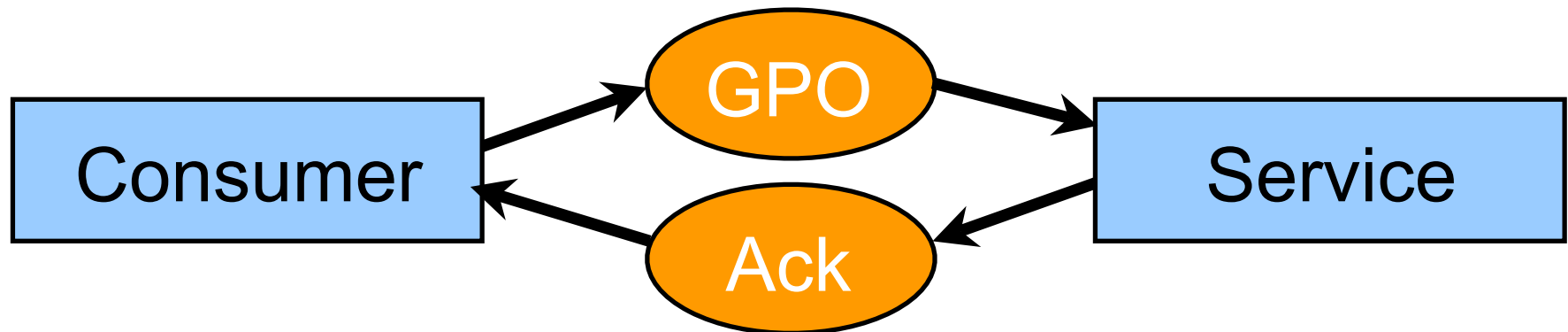
Service Oriented Architecture

General Motors

- GM migrating to common SOA
- Web Services and ebXML
- Needs to understand performance and bandwidth limits prior to adoption
- Built Proof-of-Concept in 2003
- BOD-based dealer parts ordering system
- Turned to PushToTest for independent performance test solution

UBL and BODs

- Universal Business Language
- 1.0 Spec released 4/2004
- <http://docs.oasis-open.org/ubl/cd-UBL-1.0>
- UBL in Automotive Industry
- Standards for Technology In Automotive Retailing
- <http://www.starstandard.org/>
- Business Object Document (BOD)
- <http://www.openapplications.org/>



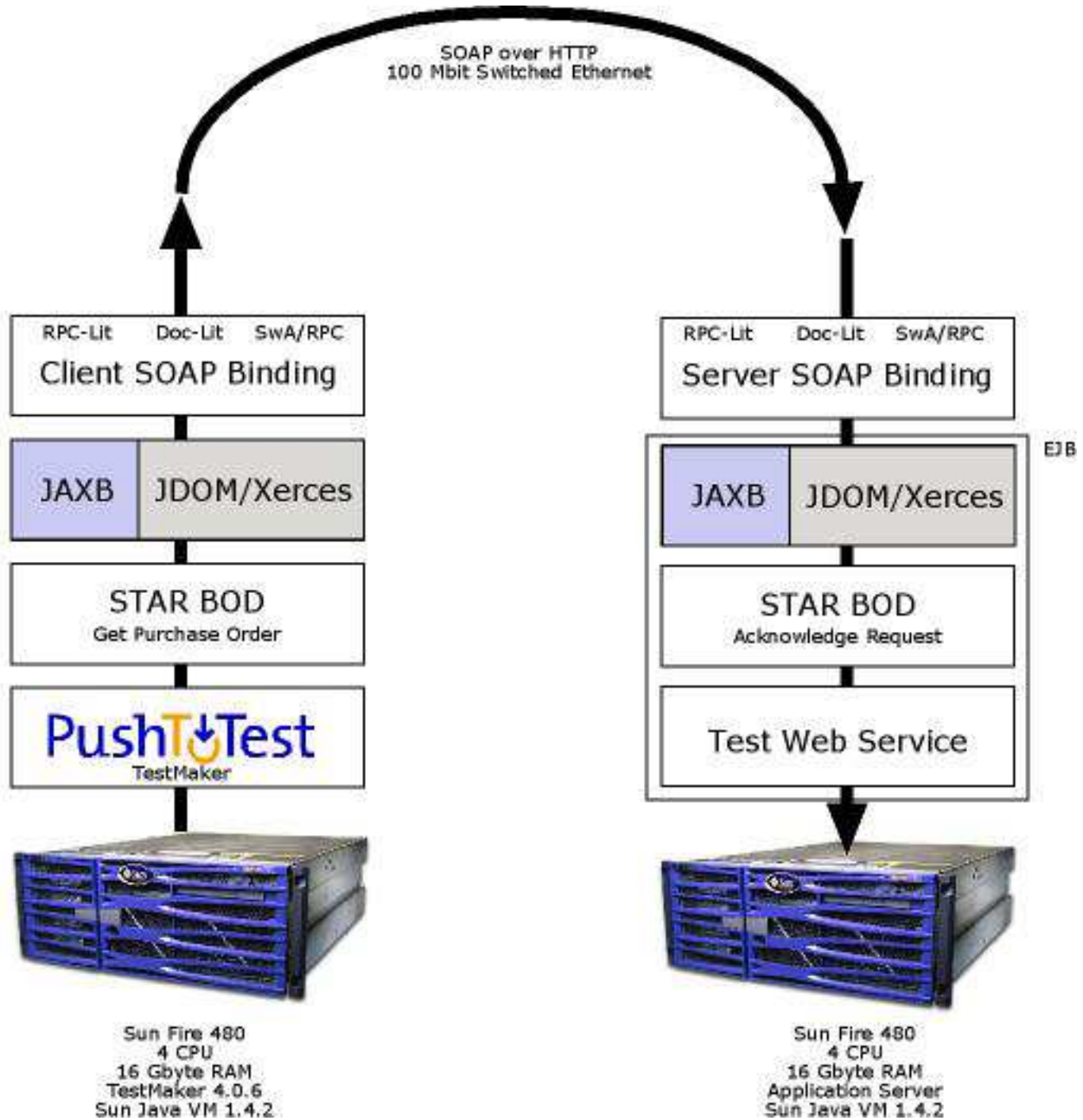
What They Are Asking

- What types of applications best fit SOA?
- What applications should avoid use of Web Services?
- What can be adjusted to improve performance?
- How do we predict capacity for WS?
- What are the best practices, architecture, design practices at WS app layer?

Use Cases

- Concurrent virtual users
- Payload size
- XML manipulation API
- SOAP Encoding style
- Coupling technique
- Network connection

The Web Service Performance Benchmark



Test Web Service

- SOAP-based Web Service
- RPC-literal, Document-literal, and SOAP with Attachments over RPC-encoded
- JAXB, JDOM/Xerces, DOM
- Bindings built by app server tools
- Stuff elements

18,000 Test Cases

Concurrent virtual requests (5)

Consumer XML handling library (2)

Request payload size (5)

Request SOAP encoding style (3)

Service XML handling library (request) (2)

Service XML handling library (response) (2)

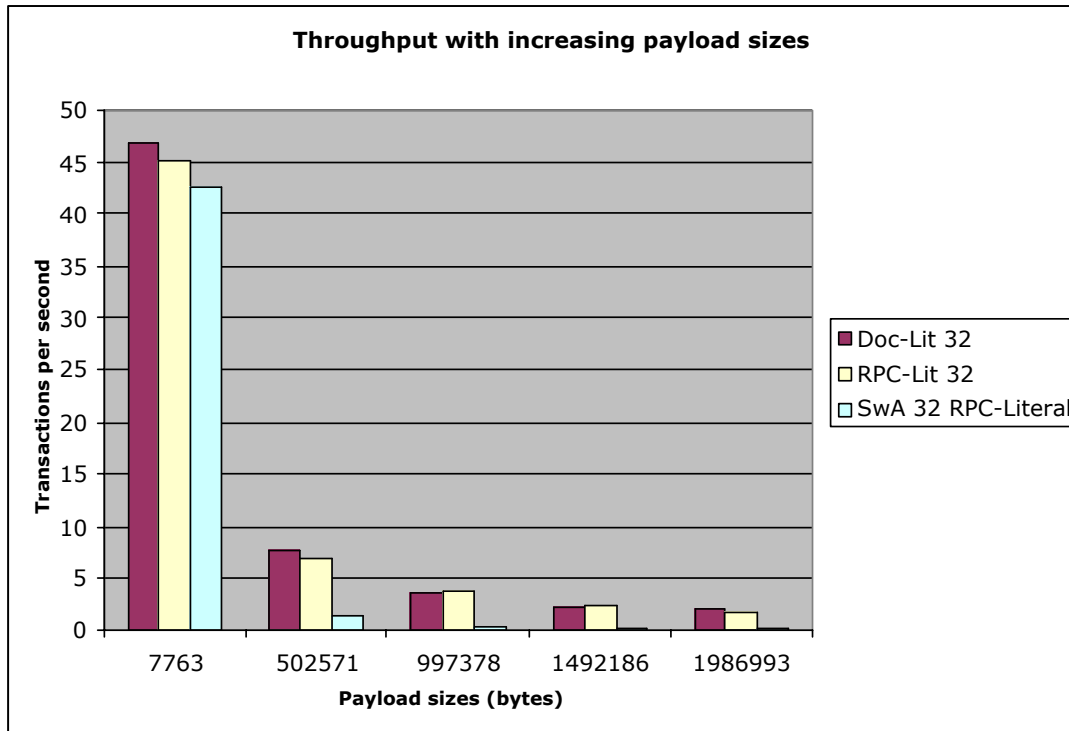
Response SOAP encoding style (3)

Response payload size (5)

Coupling technique (2)

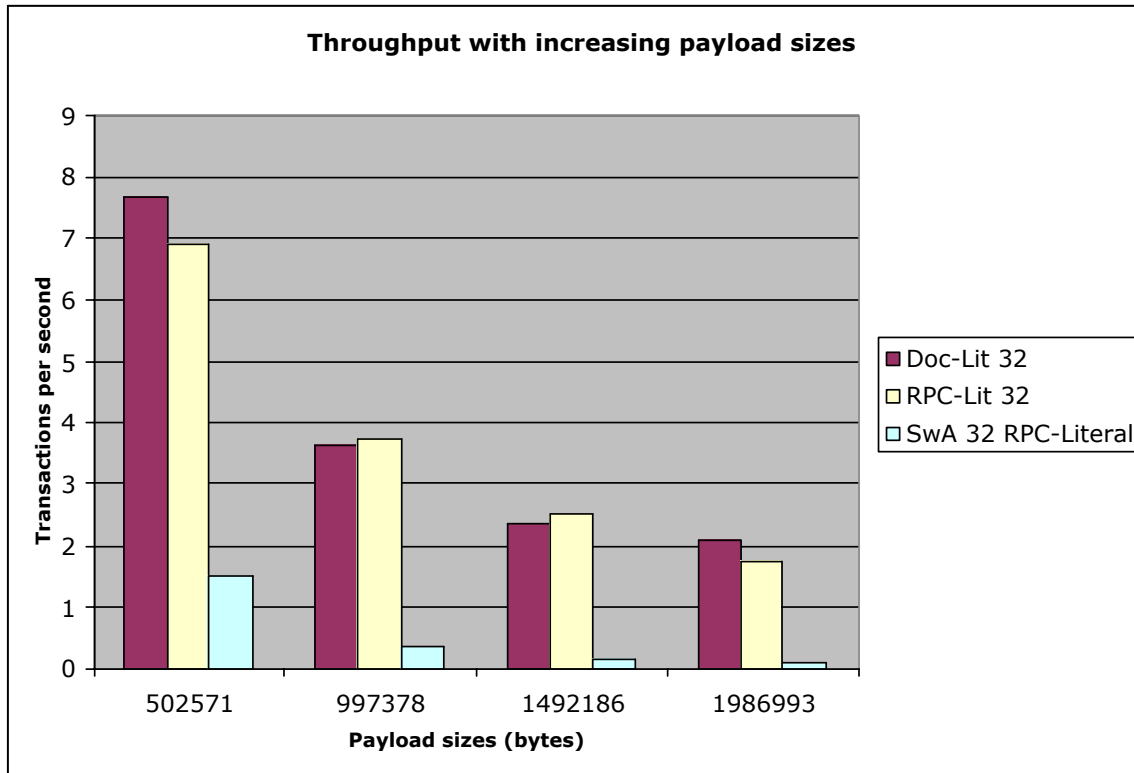
Findings

Throughput vs Payload



- Throughput declines with payload size
- Tool provides EJB-based bindings
- Minimum GPO BOD is 7768 bytes
- Network saturates

Encoding Styles

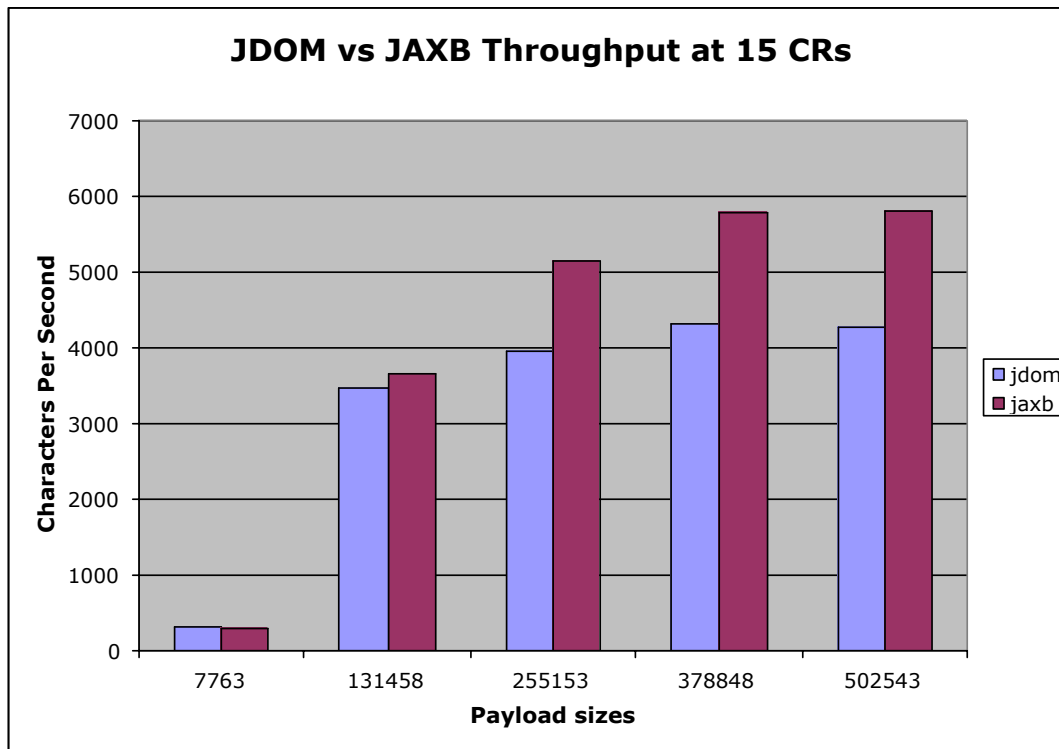


- Document-literal encoding did best

- RPC-literal encoding 10% less

- SOAP with Attachments delivered poor throughput

JAXB and BODs

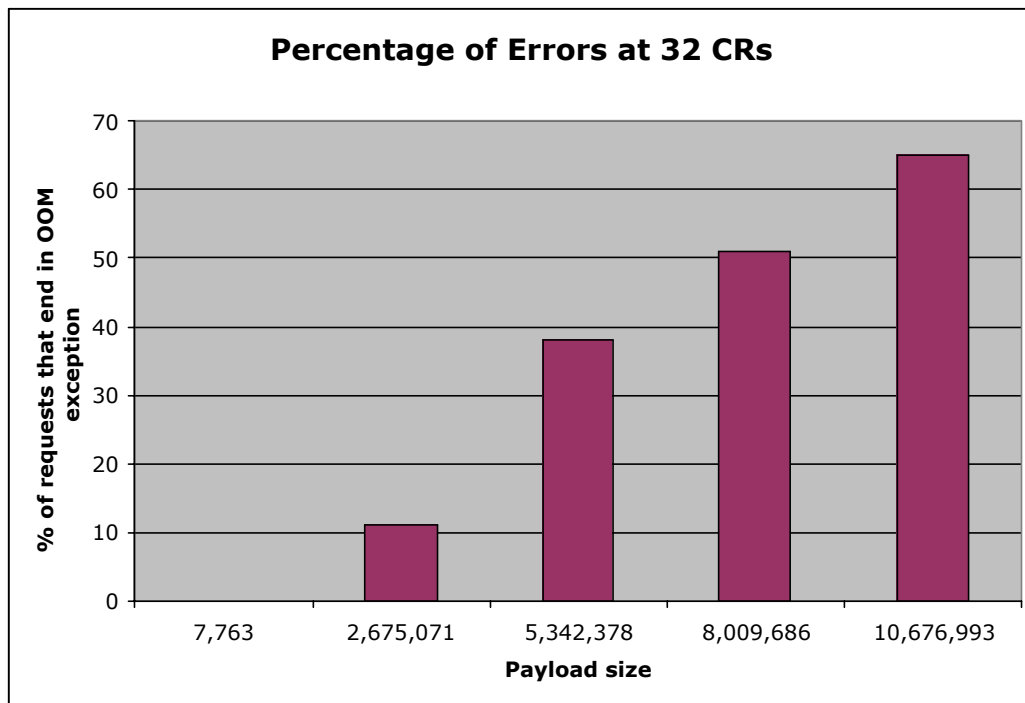


- As payload sizes grew, JAXB outperformed JDOM/Xerces
- Sun includes BODs for release tests
- JAXB team focus on performance

CPU and Network

- CPU usage reaches saturation quickly
- Network shows signs of being second bottleneck
- Software shows signs of scalability problems

Resource Exceptions



- As payload size and concurrent requests increase, JVM throws OOM exceptions
- Waits for garbage collector
- No management spec: WSDL entity stating maximum payload size

Today's SOA tools
do not scale for UBL.

Frank Cohen, 2004

Scalable SOA/UBL

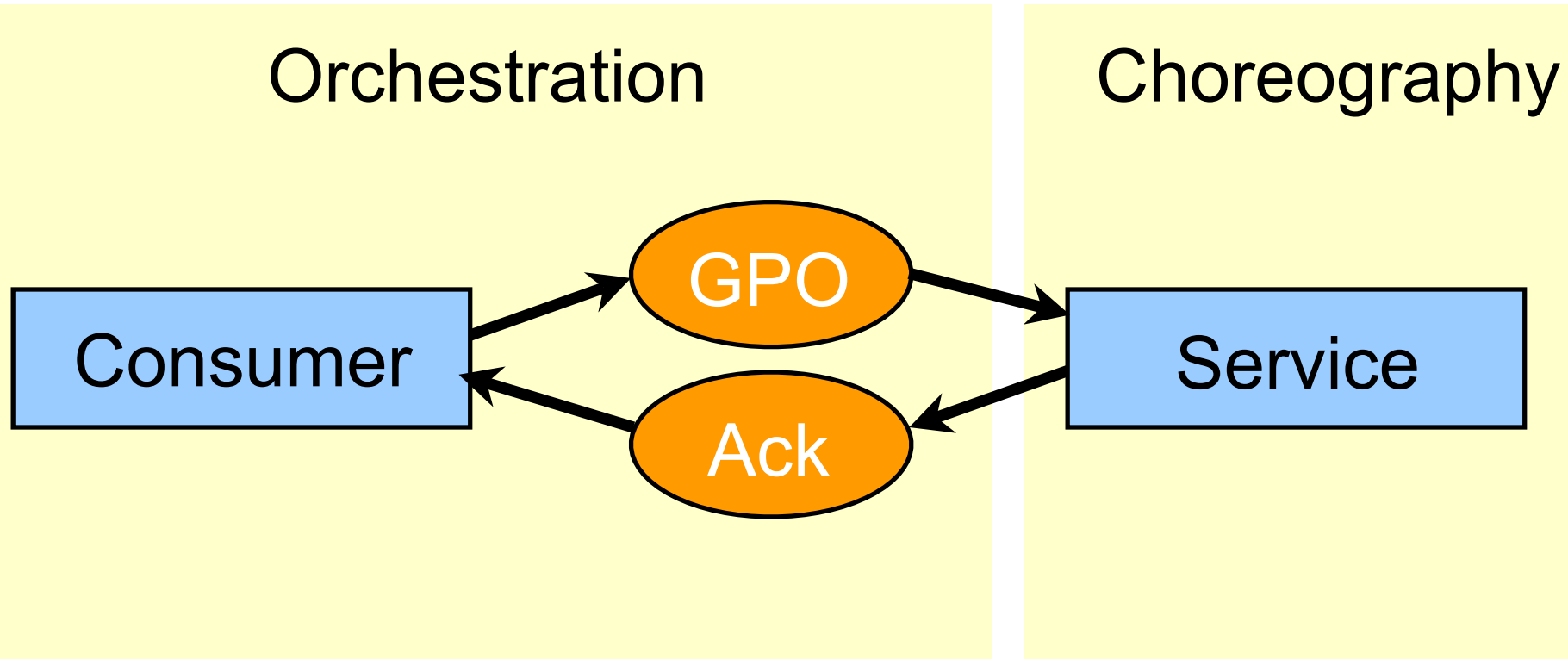
- Streams-based APIs
- JSR 173, Streaming API for XML
- <http://www.jcp.org/en/jsr/detail?id=173>
- Look for tools in 2005
- Gigabit Networks
- XML Compression

What does it take?

- Building and maintaining well performing SOA WS systems requires:
- A good understand of system architecture, code modules in SOAP stacks, and debugging skills
- Patches
- Practical software coding techniques from thought leaders

SOA Next Steps

SOA Next Steps



- Enterprises asking questions
- What is the best practice to build?
- How do you measure performance?

Resources for you

Resources

- www.pushtotest.com/ptt/kits/index.html
- www.pushtotest.com/Docs/scribblings/beapetstore
- www.pushtotest.com
- fcohen@pushtotest.com
- phone: 408 374 7426