

SOAP

Desempenho

Latency performance of SOAP implementations

Artigo

Dan Davis, Manish Parashar, *Latency Performance of SOAP Implementations*, State University of New Jersey

IEEE Cluster computing and the grid 2002

Objectivo

- Avaliar o desempenho de várias implementações SOAP relativamente à latência na invocação de procedimentos.
- Análise comparativa ao JavaRMI e CORBA utilizando o Microsoft IIS e o Apache Tomcat.

Implementações SOAP

- Apache SOAP e Axis
- SOAP RMI
- SOAP::Lite module for Perl
- Microsoft SOAP Toolkit

Comparativamente a:

- JavaRMI
- CORBA

Bancada de Teste

- o LAN de 10 Mbps.
- o Cliente: PIII 550 MHz e 256 MB RAM.
- o Servidor: AMD Athlon 700 MHz e 160 MB de RAM.
- o Microsoft Windows 2000: professional e server.
- o Precisão: +- 0,05 ms.

Bancada de Teste

- o Para cada implementação foi construído um servidor acessível remotamente através das seguintes funções:
 - void doNothing()
 - void setSize(int size)
 - String getString()
 - int[] getInteger()

Bancada de Teste

- o Cada implementação foi testada de forma "black box".
- o Cada cliente efectua várias chamadas a uma das funções, registrando o tempo necessário.

Resultados *DoNothing()*

Sistema	Linguagem	Latência (m/h)	Latência (h/d)
JavaRMI	Java	1.2	0.8
CORBA	Java	1.5	1.3
Ms SOAP	VB	16.8	200.9
SoapRMI	Java	19.5	37.7
SOAP::lite	Perl	42.0	200.1
Apache SOAP	Java	23.4	205.9
Apache Axis	Java	15.6	13.1
TCP setup time	Java	1.9	1.2
HTTP p/ TOMCAT	Java	3.4	2.6
HTTP p/ IIS	-	1.1	1.5

Latência em ms

Resultados *getString()*

Sistema	Lat. Char[200]	Lat. Char[400]	Lat. Char[800]
JavaRMI	1.3	1.2	1.3
CORBA	1.5	1.5	1.6
Ms SOAP	16.6	23.7	34.6
SoapRMI	19.3	19.6	19.9
SOAP::lite	42.4	42.3	42.6
Apache SOAP	22.8	24.2	25.1
Apache Axis	17.1	19.0	19.6

Cliente e Servidor no mesmo host

Resultados *getString()*

Sistema	Lat. Char[200]	Lat. Char[400]	Lat. Char[800]
JavaRMI	1.2	1.6	2.3
CORBA	1.7	2.1	2.7
Ms SOAP	200.2	200.3	200.3
SoapRMI	44.1	41.4	41.6
SOAP::lite	200.2	200.3	200.3
Apache SOAP	109.9	213.8	237.4
Apache Axis	14.3	14.7	15.6

Cliente e Servidor em hosts diferentes

Resultados *getIntegers()*

Sistema	Lat. Char[200]	Lat. Char[400]	Lat. Char[800]
JavaRMI	1.5	1.6	1.8
CORBA	1.5	1.9	2.0
Ms SOAP	28.1	44.8	76.7
SoapRMI	22.2	24.4	31.5
SOAP::lite	229.6	493.6	1810.6
Apache SOAP	65.8	117.3	190.7
Apache Axis	73.2	150.2	229.4

Cliente e Servidor no mesmo host

Resultados *getIntegers()*

Sistema	Lat. Char[200]	Lat. Char[400]	Lat. Char[800]
JavaRMI	2.3	3.2	4.7
CORBA	2.7	4.0	5.5
Ms SOAP	33.3	55.8	104.1
SoapRMI	75.0	73.0	108.6
SOAP::lite	400.3	600.8	1120.1
Apache SOAP	310.3	380.5	561.7
Apache Axis	76.9	138.6	243.7

Cliente e Servidor em hosts diferentes

Conclusões

- Latência para as implementações SOAP é superior à do JavaRMI e do CORBA.
- Pior desempenho é da implementação utilizando PERL.
- A nível global a implementação com melhor desempenho é a Apache Axis.

Pontos de Melhoria

- Codificação binária do XML para tornar o SOAP mais eficiente.
- A nível do número de chamadas ao sistema efectuadas para enviar uma mensagem.
- XML parsing e codificação.
- HTTP chunking (para reduzir os atrasos no *buffering*).

Artigos Relacionados

- Gryazin, Seppala, *SOAP and CORBA productivity comparison for resource-limited mobile devices*.
- Kohlhoff, Steele, *Evaluating SOAP for high performance business applications: real-time trading systems*.
- Ng, Chen, Greenfield, *An Evaluation of Contemporary Commercial SOAP Implementations*.