

J2EE 平台典型分布对象技术的性能

李凯里

(烟台大学计算机学院, 烟台 264005)

摘要: 为多种分布对象技术选择最佳设计方案时, 须对其进行性能测试与分析评估。该文通过圆周率的高精度计算, 即利用 Bailey-Borwein-Plouffe 算法对 J2EE 中几种常见的分布对象技术实施运行测试, 测试数据反映了分布对象技术的计算性能和启动特性。讨论测试结果对设计 J2EE 分布式计算系统的指导意义。

关键词: 计算性能; 分布对象; Web 服务

Capability of Typical Distributed Objects Techniques for J2EE Platform

LI Kai-li

(School of Computer, Yantai University, Yantai 264005)

【Abstract】 It is necessary to test and analyze the capability of various distributed objects techniques. To solve the problem to choose preferred plan from a number of distributed objects techniques, some familiar J2EE distributed objects techniques are tested by the high-precision arithmetic of circumference ratio(Bailey-Borwein-Plouffe arithmetic). A lot of data obtained by these tests reflect the capability of computing and starting on these techniques. The significance of these data for application is discussed to design distributed computing system on J2EE.

【Key words】 computing capability; distributed objects; Web services

分布对象技术种类繁多, 仅 Java 体系中就有 Jini(RMI), Servlet, CORBA, SOAP 等技术的 Java 实现。技术的多样性为不同用途提供了多种选择, 体现了设计的灵活性。但因为需要针对不同应用场合选择最佳方案, 所以必须对各项技术进行性能分析和评估。本文以 J2EE 系统为测试平台, 通过一个实际网络计算任务, 对上述几种典型分布对象技术进行性能测试, 对结果进行分析评估, 为系统设计提供了依据。

1 性能测试方案的设计

性能测试主要包括计算性能、内存消耗、启动时间、可伸缩性等。虽然分布对象技术工作于特殊分布环境, 但仍须考虑计算性能, 因此, 需要选择具有一定复杂度、结果为预知的计算任务作为其评判标准。本文按常用方法, 即以圆周率的高精度计算任务为例, 以对象的平均调用时间作为计算性能的评判依据^[1]。目前有很多计算高精度圆周率的方法, 但只有少数几种符合分布测试需要。本文采用最适用于分布式计算的 Bailey-Borwein-Plouffe 算法, 即

$$\pi = \sum_{n=0}^{\infty} [4/(8n+1) - 2/(8n+4) - 1/(8n+5) - 1/(8n+6)]/16^n \quad (1)$$

取小粒度对象以充分体现分布式计算系统的特点(如依赖网络的远程调用等), 即将式(1)中括号内的求和部分作为分布计算对象。通过频繁调用该计算对象的方法, 可以很好地模拟分布式计算环境, 使测试结果尽可能逼近实际情形。

由于进程加载、对象远程调用和分布数据处理等因素的影响, 因此会出现测试数据抖动现象。为消除数据抖动对性能评估的不良影响, 需要对各调用时间样本取均值, 并以趋于稳定的时间均值作为其测评依据, 本文引入如下迭代算法:

$$\bar{t}_n = (\bar{t}_{n-1}(n-1) + t_n)/n \quad (2)$$

其中, n 为调用次数; t_n 为第 n 次调用的耗时。

测试数据的耗时抖动现象如图 1 所示。

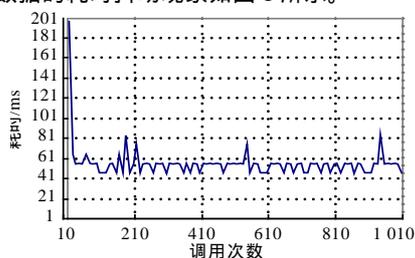


图 1 测试数据的耗时抖动现象

2 方案实施及测试结果

分布对象技术通常采用 C/S 模式, 但实施方式存在差异, 表现为工作机理及代码结构的不同。本文主要讨论前者, 并定义平均计算时间为计算性能; 平均计算时间与启动时最大平均耗时之比为启动特性。

2.1 SOAP 的 Java 实现技术(Axis)

Axis 是 SOAP 的 Java 实现技术, 它基于 HTTP 协议, 运行时需要一个 Web 应用服务器作为支撑。本文选用 Tomcat4.1, 如图 2 所示。



图 2 SOAP 的 Java 实现

作者简介: 李凯里(1958 -), 男, 副教授、硕士, 主研方向: 计算机网络应用, 分布对象技术

收稿日期: 2007-10-25 **E-mail:** klwplx@163.com

根据测试方案设计服务对象 `cmp_term_sum.jws`，并将其部署到 Web 服务器上。每次调用该方法就完成式(1)中多项式的计算。

```
public class cmp_term_sum{
    public BigDecimal term_sum(int i, int scale){...} //i 为调用次数，
    //scale 为计算精度 }

```

客户端需要运行 Axis 工具，根据 WSDL 在本地生成如下 4 个 Java 类：

(1)`Cmp_term_sum.java`。定义 Web 服务接口，在本文中只有一个 `term_sum` 方法。

(2)`Cmp_term_sumService.java`。定义用于获取 Web 服务接口的方法。

(3)`Cmp_term_sumServiceLocator.java`。接口 `Cmp_term_sumService` 的本地实现。

(4)`Cmp_term_sumSoapBindingStub.java`。Web 服务客户端桩通过该类与服务器交互。

完成上述工作后，须设计客户应用程序(完成远程对象调用和数据处理)，并连同上述 4 个类一起编译、运行。计算精度统一取小数 1 000 位(获得的计算数据见第 3 节)。

SOAP 的运行结果如图 3 所示，其平均计算时间为 27.23 ms，启动特性为 0.54。

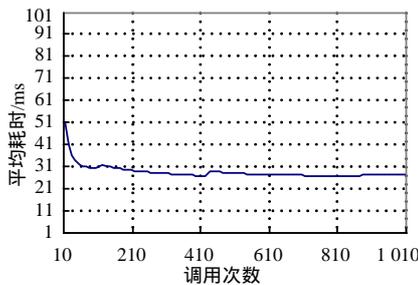


图 3 SOAP 的运行结果

2.2 Jini 的 Java 实现技术

如图 4 所示，Jini 技术按注册服务、查找服务和调用服务的周期执行^[2]。

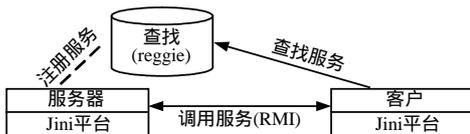


图 4 Jini 技术原理

服务器与客户端都运行于 Jini 平台并遵照共同的接口标准，本文定义了如下远程接口 `cmpi`：

```
public interface cmpi extends Remote{
    public BigDecimal term_sum(int i, int j) throws Remote
    Exception;} //方法 term_sum()用于计算式(1)的求和项

```

上述接口由服务器应用程序实现(采用了服务代理对象)，先运行该服务器应用程序，然后将代理对象注册到查找服务器 `reggie`(由 Jini 平台提供)。

远程客户需要运行客户应用程序，先通过查找服务，找到与标准接口相匹配的服务代理对象的信息，然后据此调用该服务代理对象(相当于 RMI 存根对象)，完成远程服务调用并得到计算结果。RMI 即远程方法调用，是提供对象之间相互通信的 Java 技术。

Jini 的运行结果如图 5 所示，平均计算时间 7.39 ms，启动特性为 0.52。

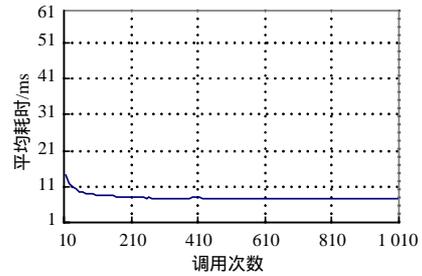


图 5 Jini 的运行结果

2.3 CORBA 的 Java 实现技术

与 Jini(RMI)相比，CORBA 支持更广泛的分布对象(如 C/C++、Ada 等)^[2]。该技术要求定义一个中性语言接口，本文定义了计算 PI 的接口如下：

```
module PIApp{interface PI{string term_sum(in long i, in long
scale);}; };

```

通过 `idlj` 工具可以派生出所有必需的服务框架、客户存根、接口、实现、辅助类等，具体如下：

(1)`PIImplBase.java`。该抽象类是一个服务器框架 (Skeleton)，它可为服务器提供基本的 CORBA 功能，实现了 `PI.java` 接口。服务器应用程序扩展该类。

(2)`PIStub.java`。该类是客户机桩(存根)，可为客户机提供 CORBA 功能。

(3)`PI.java`。该接口含有 IDL 接口的 Java 版本。扩展了 `org.omg.CORBA.Object`。并提供标准的 CORBA 对象功能。

(4)`PIHelper.java`。该类提供辅助功能，包括将 CORBA 对象引用转换为适当类型所需的 `narrow()` 方法。

(5)`PIHolder.java`。是一个终态类，其中含有 `PI` 类型的公有实例成员。

(6)`PIOperations.java`。是一个接口类，其中包含方法 `term_sum(int i, int scale)`。

这些接口类带有协议性质，在客户机和服务器端，需要分别设计应用程序 `PIClient.java` 和 `PIServer.java`，以实现上述各接口类的具体计算功能，如图 6 所示。

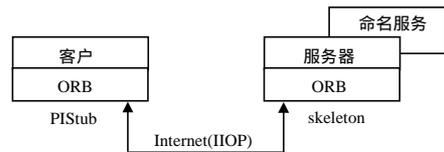


图 6 CORBA 的 Java 实现

测试运行时，服务端应用程序要将服务对象注册到一个命名服务器上，供客户机查找调用。本文启动了 JDK 自带的 `tnameserv` 命名服务器，并运行客户应用程序完成调用。

CORBA 的测试结果见图 7，平均计算时间为 11.64 ms，启动特性为 0.92。

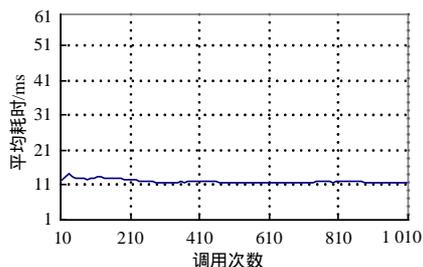


图 7 CORBA 的测试结果

2.4 Servlet 对象技术

Servlet是应用于Java服务器端的对象技术，它通过Web服务器接收来自客户的请求，执行计算任务并返回结果^[3]，是基于HTTP协议的分布对象技术。客户机通常经由浏览器与Servlet通信，但为了与其他技术相一致，本文通过“URL url = new URL(“http://hostname:port/servlet/PIServlet”);”语句发出请求，逐次返回求和数据并代入式(1)进行计算。在服务器端，服务对象类PIServlet通过service()方法响应请求，直接实现一个计算式(1)求和项的Servlet对象实例。该服务类(PIServlet.class)事先已部署在Web服务器上，供客户机调用。

上述Servlet不支持对象的注册和查找服务，不属于完备的分布对象技术，即Servlet技术不适用于需要大量调用不同远程对象方法的场合。对其进行测试是因为需要对其的运行环境，即Web服务器(HTTP协议)进行评估。

Servlet的运行结果显示，其平均计算时间11.08 ms，启动特性0.64。

3 方案综合评估

3.1 计算精度

圆周率标准值如下：

3.141 592 653 589 793 238 462 643 383 279 502 884 197 169 399 375 105 820 974 944 592 307 816 406 286 208 998 628 034 825 342 117 067 9...598 253 490 428 755 468 731 159 562 863 882 353 787 593 751 957 781 857 780 532 171 226 806 613 001 927 876 611 195 909 216 420 198 9

所有方案的计算数据取小数点后1 000位，与标准值的核对结果表明，计算无误。

3.2 分析评估

由表1可知，在计算性能方面，Jini技术最好，SOAP最差。因为Jini以RMI机制为对象间的通信基础，是较底层的技术，所以执行效率较高。SOAP是Web services的基础技术之一，是基于XML的在分布式环境中实现信息交换的简单协议^[4]，虽然其功能强大(如支持对象的发布、查找、绑定操作)，但存在开销较大的问题，影响了执行效率^[4]。

表1 各Java实现技术的测试结果

技术方案	计算性能/ms	启动特性
SOAP	27.23	0.54
Jini	7.39	0.52
CORBA	11.64	0.92
Servlet	11.08	0.64

SOAP、CORBA和Servlet技术计算性能欠佳的另一个原

因可能是环境平台性能造成的。实际应用中SOAP与Servlet都依赖Web服务器，会受到服务响应时间的影响。CORBA则更多地归因于初始化ORB、创建服务对象、查找调用远程对象以及IIOP通信机制等诸多运行环节带来的时间开销^[5]。

CORBA技术的启动特性明显高于其他技术，由于它采用的命名服务tnameserv是Java自带的轻型服务器，因此执行速度较快。CORBA的实现环节较多，但每个模块不大，容易加载，便于迅速启动。

Servlet技术执行环节不多，且没有涉及对象注册与查找，但其计算速度没有得到显著提高。

目前流行的Web服务不适用于高速分布式计算的应用场合。各个技术的运行曲线平稳度存在差异，其中，SOAP曲线波动最大，表明其受分布环境各种因素的影响较大。

4 结束语

本文通过上述测试分析得到如下结论：(1)对单一语言(Java)的应用场合，须考虑计算性能，因此，Jini技术是最好的选择。(2)在多语言跨平台环境下，须考虑服务功能特性，应采用SOAP或CORBA技术，而CORBA的计算性能优于前者。(3)对需要频繁启动的应用，即需要大量调用不同远程对象方法的场合，CORBA技术是较好的选择。(4)只有必须采用Web服务进行互操作或远程调用时，才考虑SOAP技术。(5)采用Web服务器的场合都不能忽略其性能对计算效率的影响。

上述结论为J2EE平台分布对象技术实用系统的设计提供了参考价值。但本文某些结论不够确切，CORBA启动特性为何明显较高，SOAP技术的数据稳定度为何较差，Web服务效率不高的主要原因等问题有待深入研究。

参考文献

- [1] 王克宏, 王少锋, 姜河, 等. Java2 程序设计[M]. 北京: 清华大学出版社, 2002.
- [2] Boger M. Java 与分布式系统[M]. 曹学军, 译. 北京: 机械工业出版社, 2003.
- [3] Moss K. Java Servlet 开发人员指南[M]. 袁勤勇, 张玉魁, 译. 北京: 清华大学出版社, 2003.
- [4] 顾宁, 刘家茂, 柴晓路, 等. Web services 原理与开发实践[M]. 北京: 机械工业出版社, 2006.
- [5] Harkey R O D. Java 与 CORBA 客户/服务器编程[M]. 亢勇, 译. 北京: 电子工业出版社, 2004.

(上接第255页)

协同发展为主体业务发展目标的道路运输物流企业。在研究道路运输物流企业的各种日常经营管理及业务过程管理后，构建可支撑企业实际业务运作的“四领域”主体业务模型，并以此设计、开发和实施道路运输物流信息平台。本文研究成果已在国内某道路运输物流企业的信息化建设中得到应用，取得了较好的经济和社会效益。笔者将进一步改造、完善该平台，使其具有普遍推广的价值，在道路运输物流企业的日常运营和业务管理中发挥重要作用。

参考文献

- [1] Coyle J J. 运输管理[M]. 北京: 机械工业出版社, 2004.
- [2] 蔡淑琴. 物流信息系统[M]. 北京: 中国物资出版社, 2002.
- [3] 曾登高. NET 系统架构与开发[M]. 北京: 电子工业出版社, 2003.
- [4] Perry S C. C#和.NET 核心技术[M]. 北京: 机械工业出版社, 2006.
- [5] 微软公司. Microsoft .NET Server 企业解决方案[M]. 帅凌鹰, 黎晓冬, 译. 北京: 清华大学出版社, 2003.