

JMS PTP 体系结构的 MDB 组件的设计和通信性能测试^{*}

冼广铭 王知衍 黄 鲲

(华南理工大学计算机科学与工程学院 广州 510640)

摘 要 JMS 定义了 Java 访问消息中间件标准的和统一的接口,克服了 RPC 中间件技术的同步通信、紧密耦合等的局限性。消息驱动 Bean(MDB)组件作为一种完善的 JMS 客户机,很好地利用了已有的 EJB 和 J2EE 技术。本文使用 JMS PTP 扇出体系结构,以消息驱动 Bean 组件作为消息异步消费者,并且以消息大小、应答模式、线程数、JVM 选项等作为影响吞吐量的 4 个因素,对其通信性能进行了测试。采用方差分析、最小正规二次无偏估计等数理统计方法分析实验数据,获得了各个因素及其各个水平对吞吐量边际均值的具体影响程度,为通信性能的增益提供了有效的参考价值。

关键词 消息中间件, JMS PTP 体系结构, 消息驱动 Bean, 方差分析, 最小正规二次无偏估计, 通信性能测试

Design and Communication Performance Test of Message Driven Bean Component Based on JMS PTP Model

XIAN Guang-Ming WANG Zhi-Yan HUANG Kun

(School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640)

Abstract JMS (Java Message Service) defined the standard and uniform interface of message oriented middleware (MOM) for Java and has overcome the drawbacks of synchronous communication and compact coupling of RPC (remote procedure call) middleware. As a kind of JMS client, message driven bean (MDB) utilized the perfect technology of EJB and J2EE. In this paper, by using MDB as an asynchronous message consumer and JMS PTP fan-out architecture, communication performance test has been performed according to relationship between the four factors of message size, acknowledge mold, threads number, JVM option and the dependent variable (throughput). Experimental data were analyzed by the statistic methods of ANOVA (analysis of variance) and MINQUE (Minimum Norm Quadratic Unbiased Estimation). Simulation results indicated that communication performance could be improved according to estimated marginal means of throughput affected by each factor and its levels.

Keywords Message oriented middleware, JMS PTP architecture, Message driven bean, Analysis of variance, Minimum norm quadratic unbiased estimation, Communication performance test

JMS(Java Message Service)是由 SUN 公司提出的一个关于消息传递和接收的标准^[1]。消息中间件系统(MOM)有 BEA 的 Message、Progress 的 SonicMq、Microsoft 的 MSMQ 等。为了实现 MOM(Message Oriented Middleware)系统的无缝连接和相互操作, JMS 定义了 Java 访问消息中间件标准的和统一的接口,它是一种松散的企业级应用集成方法。

JMS 提供者者和 JMS 客户端通过 JMS 应用程序接口来进行交互(如图 1 所示)。JMS 消息提供者保持了 JMS 消息队列和主题。消息发送者和消息提供者可以从 JNDI 名字中查找到消息队列和主题。

由 JMS 提供的 API 可以支持分布对象计算、远程过程调用、消息传递等的分布模型^[2]。

JMS 客户端和 JMS 提供者通过消息进行通信, JMS 消息由消息头(Header)、属性(Properties)、消息体(Body)3 部分组成。

1 JMS 和 MDB 组件的设计^[3,4]

1.1 RPC 中间件技术及其局部性和 JMS 的优点

如今分布式系统的复杂程度越来越高,因此由于

DCOM、RMI 与 CORBA 等 RPC 中间件技术逐渐显示出同步通信和紧密耦合的局限性,容易造成程序阻塞、客户接受异常等情况。

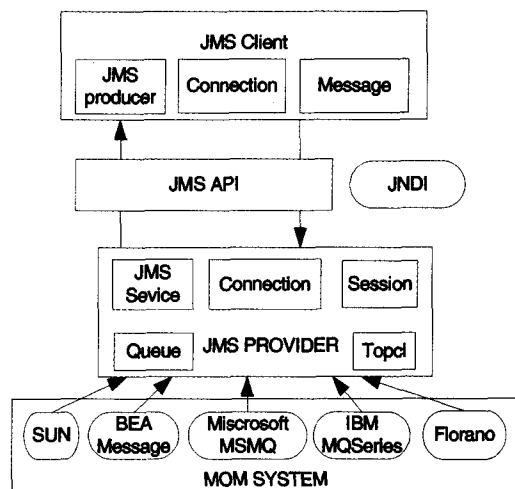


图 1 JMS 体系结构空间

^{*} 科技部科技型中小企业技术创新基金无偿资助项目(立项代码: O2C26214400224)、广东省科技计划项目资助(项目编号: 2002A1020104)。
冼广铭 博士生,研究方向为软件工程和计算机通信信息处理等;王知衍 博士生导师,IEEE 会员,主要研究领域为图形图像处理;黄 鲲 博士研究生,研究方向为计算机辅助工程设计。

与 RPC 中间件技术相比,面向消息的中间件可以进行松散的异步通信,使上述 RPC 中间件技术问题得到很好的解决。

采用异步方式传递消息,发送和接受是异步的。消息一旦发送到若干队列中,无需等待处理过程都完成,客户机就可以继续处理程序。

JMS 对一个消息可以发送给一个或多个接受者,支持一对一和一对多的异步或者是同步通信。而在实际开发过程中,同步通信比较少使用。

JMS 消息发送可以分为两种类型:点对点(PTP)通信模型和发布/订阅(Pub/Sub)通信模型。

1.2 消息驱动 Bean 通信原理

消息驱动 Bean 可以通过一个 MOM 服务器发送和接收消息,它使用了重用消息传递组件。

客户端并非直接调用消息驱动 Bean,因为 MDB 没有本地和远程接口。当 JMS 队列或是 JMS 主题接收到消息后,EJB 容器将调用消息驱动 Bean。

客户端只能通过消息驱动 Bean 监听的队列或主题与 MDB 进行通信。MDB 是一个实现了两种特殊接口的类:一种是消息传递接口(java.jms.MessageListener),另一种是与 EJB 容器之间的接口(javax.ejb.MessageDrivenBean)。

2 基于 JMS PTP 体系结构的消息驱动 Bean 的通信性能测试配置

2.1 JMS PTP 扇出体系结构^[5]

如图 2 所示,实验使用点对点通信,消息生产者发送消息给一个特定的队列,消息消费者(MDB)从同一队列中接收消息。每个消息生产者对应唯一的消息消费者。

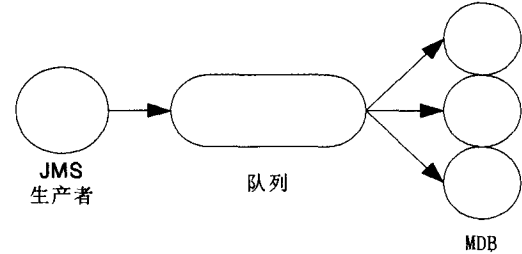


图 2 JMS PTP 扇出体系结构

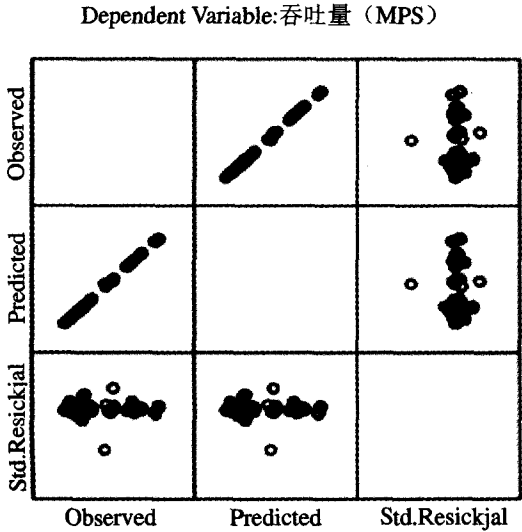


图 3 残差图

2.2 测试条件和目标

如表 1 所示,实验采用了 3 种消息大小 MesSize(1kB, 10kB,20kB),4 种应答模式 AckMold(AUTO_ACKNOWLEDGE, DUPS_OK_ACKNOWLEDGE, CLIENT_ACKNOWLEDGE(80),NONE),4 种线程数量 Threads(10,20,30,40)和 3 种 JVM 选项 JVMOpt(HotSpot Server, HotSpot Client, Classic),共 144 种组合情况,分别研究了这 4 种因素对吞吐量(Throughput)的影响。吞吐量的单位为 MPS(每秒处理的消息数)。

表 1 因素变量表
Between-Subjects Factors

		Value Label	N
消息大小	0	1kB	48
	1	10kB	48
	2	20kB	48
应答模式	0	AUTO_	36
		ACKNOWLEDGE	
	1	DUPS_OK_	36
		ACKNOWLEDGE	
	2	CLIENT_	36
		ACKNOWLEDGE(80)	
	3	NONE	36
线程数	0	10	36
	1	20	36
	2	30	36
	3	40	36
JVM 选项	0	Hot Spot Server	48
	1	Hot Spor Client	48
	2	Classic	48

表 2 方差分析结果
Tests of Between-Subjects Effects
Dependent Variable:吞吐量(MPS)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	7963027.111 ^a	10	796302.711	1250.790	.000
Intercept	18599093.8	1	18599093.78	29214.463	.000
Mes Size	7111586.889	2	3555793.444	5585.250	.000
AckMold	834250.278	3	278083.426	436.799	.000
Threads	366.222	3	122.074	.192	.902
JVMOpt	16823.722	2	8411.861	13.213	.000
Error	84673.111	133	636.640		
Total	26646794.0	144			
Corrected Total	8047700.222	143			

a. RSquare=.080(Adjusted R Squared=.080)

表 3 方差成分分析
Variance Estimates

Component	Estimate
Var(AckMold)	7707.170
Var(MesSize)	74066.003
Var(JVMOpt)	162.220
Var(Error)	625.289

Dependent Variable: Throughput

Method: Minimum Norm Quadratic Unbiased Estimation

(Weight = 1 for Random Effects and Residual)

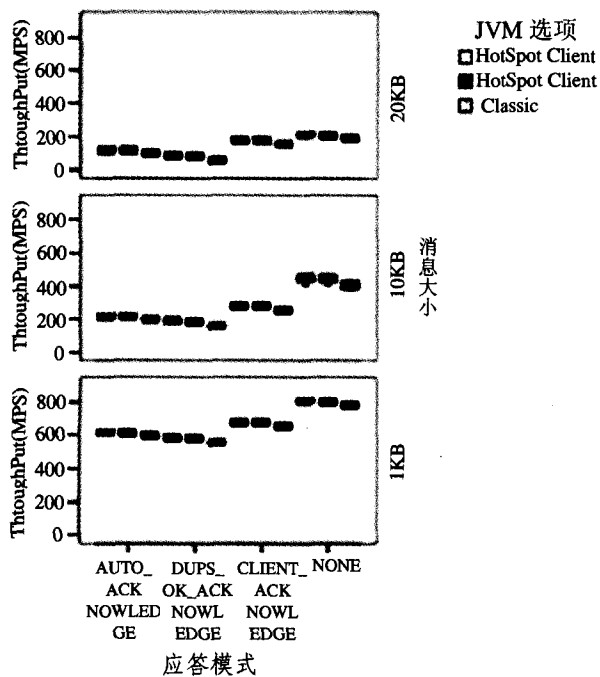


图4 吞吐量的总体分布

3 测试结果分析

3.1 模型拟合结果

从表2的F检验结果可以看出, MesSize(消息大小)、AckMold(应答模式)和JVMOpt(JVM选项)的显著性概率 p (Sig. = 0.000) 值均小于 0.01, 而 Threads(线程大小)显著性概率 p (Sig. = 0.902) 值大于 0.01。可见4个因素当中, 消息

大小、应答模式和 JVM 选项对吞吐量的影响是明显的, 而线程大小对吞吐量的影响可以忽略不计。

$R\text{-Squared} = SS(\text{Model}) / SS(\text{Total}) = 0.989$, 说明模型解析了总变异的 98.9%, 达到了良好的效果。

图3显示模型预测值与观察值之间的线性趋势非常明显, 而标准化残差的分布也合理, 说明模型拟合得极为理想。

3.2 各因素的方差成分分析结果

剔除线程数, 分别对消息大小、应答模式和 JVM 选项采用最小正规二次无偏估计(MINQUE)进行方差成分分析。

各随机效应对吞吐量变化的影响如表3所示。可见消息大小(MesSize)对吞吐量的影响是最大的, 应答模式(AckMold)次之, JVM 选项(JVMOpt)第三。

图4显示吞吐量受消息大小、应答模式和 JVM 选项等因素影响的总体分布, 可见消息大小对吞吐量分布的影响明显, 而吞吐量总体分布在不同的应答模式和 JVM 选项中也呈现出一定的规律。

3.3 各因素的吞吐量边际均值图分析

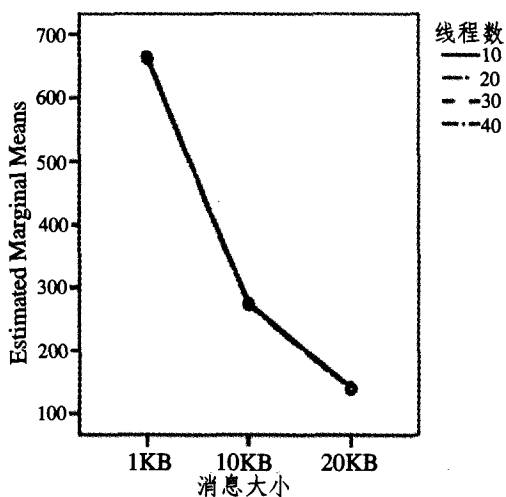
根据上述分析结果, 按照各因素对吞吐量影响从小到大的程度(线程数——JVM 选项——应答模式——消息大小)进行进一步的分析。

3.3.1 线程数对吞吐量的影响

图5(a)和(b)分别表示了不同消息大小、不同应答模式受线程数影响的吞吐量边际均值图。

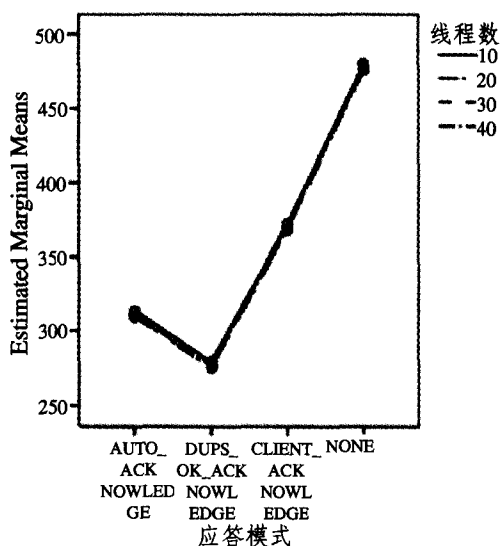
两图中, 4种线程数(10, 20, 30, 40)的边际均值曲线都是几乎重叠的。因此, 线程数对吞吐量的影响是极为微小的, 达到了可以忽略的地步, 这与方差分析的结果是一致的。

Estimated Marginal Means of 吞吐量 (MPS)



(a) 不同消息大小

Estimated Marginal Means of 吞吐量 (MPS)



(b) 不同应答模式

图5 受线程数影响的吞吐量边际均值图

3.3.2 JVM 选项对吞吐量的影响

从图6(a)和(b)中可以看出, 在发送消息大小或应答模式不同的情况下, 选用 JVM 的 HotSpot_Server 和 HotSpot_Client 对吞吐量边际均值的影响差别是不大的, 而选用 JVM 的 Classic 时吞吐量边际均值有所下降。

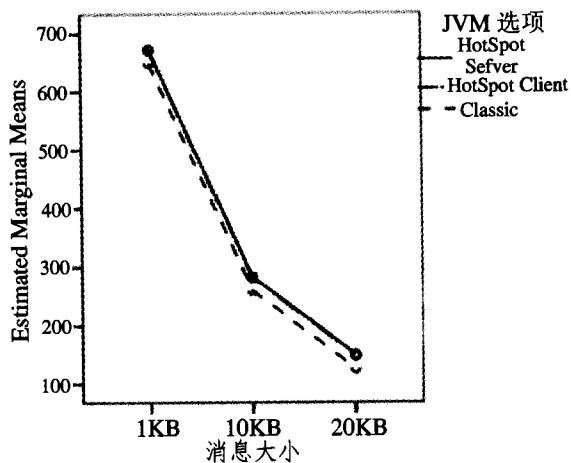
3.3.3 应答模式对吞吐量的影响

从图7(a)、(b)和(c)中可看出, 由于无需应答消息, 采用 None 应答模式的吞吐量边际均值是最高的。采用 Client_Acknowledge 比默认的模式 Auto_Acknowledge 获得了吞吐量上的性能增益。采用 Dups_OK_Acknowledge 时, 吞吐量较小。

3.3.4 消息大小对吞吐量的影响

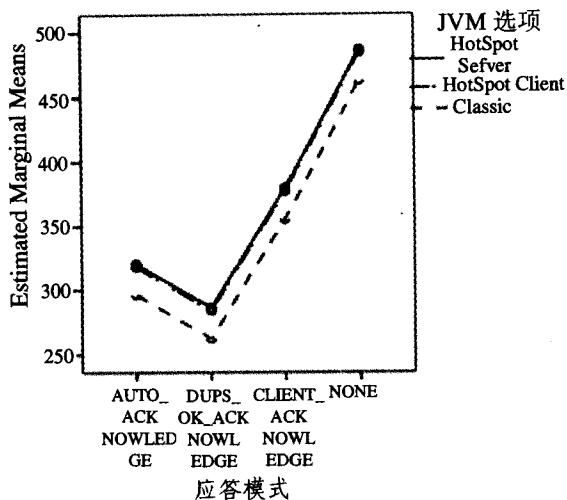
图8(a)和(b)均显示了在不同线程数、应答模式的情况

Estimated Marginal Means of 吞吐量 (MPS)



(a)不同消息大小

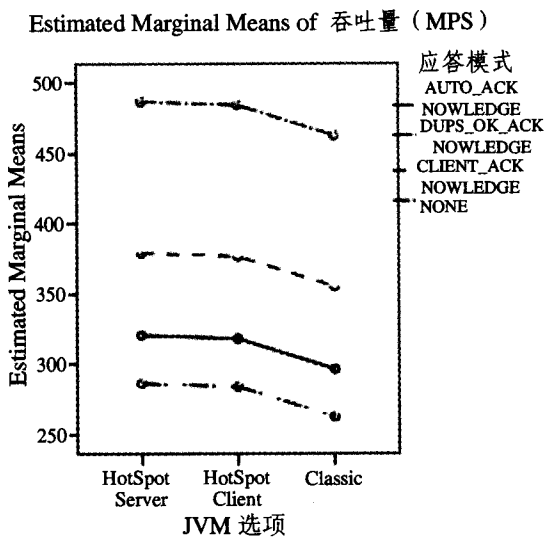
Estimated Marginal Means of 吞吐量 (MPS)



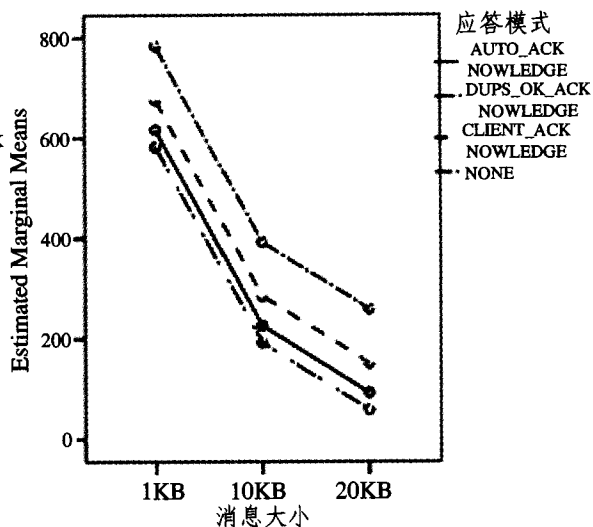
(b)不同应答模式

图 6 受 JVM 选项影响的吞吐量边际均值图

Estimated Marginal Means of 吞吐量 (MPS)

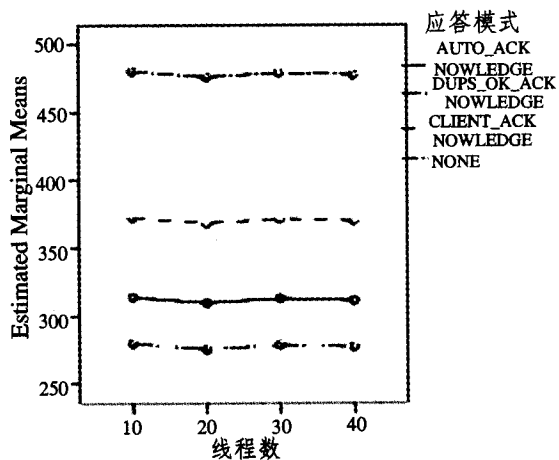


(a)不同JVM选项



(b) 不同消息大小

Estimated Marginal Means of 吞吐量 (MPS)



(c)不同线程数

图 7 受应答模式影响的吞吐量边际均值图

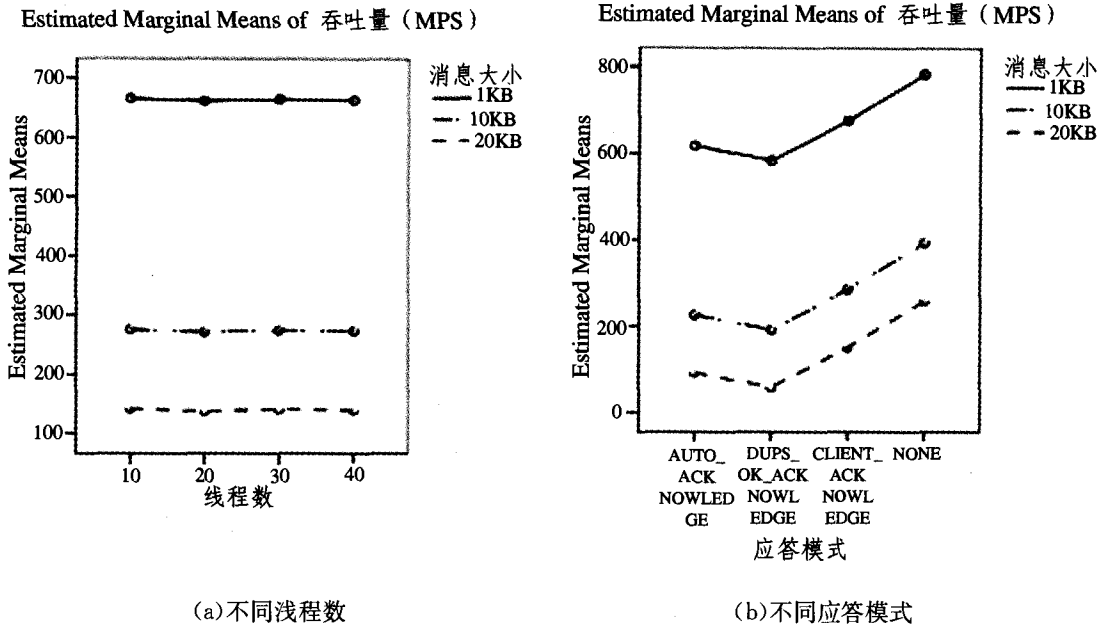


图 8 受消息大小影响的吞吐量边际均值图

表 4 吞吐量最大和最小边际均值比较

吞吐量边际 均值最大	消息 大小	应答模式	JVM 选项	Mean	Std. Error	95% Confidence Interval	
						Lower bound	Upper bound
吞吐量边际 均值最大	1kB	None	HotSpot Server	808.750	3.753	801.308	816.192
吞吐量边际 均值最小	20kB	DUPS_OK_ ACKNOWLEDGE	Classic	60.000	3.753	52.558	67.442

下,当发送的消息较小时(1kB),吞吐量的边际均值较大;当发送的消息增大时,吞吐量的边际均值随之减少。当发送消息大小从1kB增加到10kB时,吞吐量的边际均值降幅较大;当发送消息大小从10kB增加到20kB时,吞吐量的边际均值降幅减缓。

3.4 吞吐量的总体分布以及最大和最小边际均值比较

表4给出了在不考虑线程数的影响时,吞吐量最大和最小边际均值的比较。

当消息大小为1kB,应答模式为None,JVM选项为HotSpot_Server时,吞吐量最大边际值为808.750MPS,95%的置信区间为(801.308,816.192)。当消息大小为20kB,应答模式为DUPS_OK_ACKNOWLEDGE,JVM选项为Classic时,吞吐量最小边际值为52.558MPS,95%的置信区间为(52.558,67.442)。

结论 本实验基于JMS PTP模型,采用MDB组件作为消息的异步消费者。以消息大小、应答模式、线程数量和JVM选项作为因素和吞吐量作为因变量,对通信性能进行了

有效的测试。通信性能测试中消息大小具有3种水平,应答模式具有4种水平,线程数量具有4种水平和3种JVM选项具有3种水平,共144种组合情况。通过采用方差分析、最小正规二次无偏估计等数理统计方法对实验数据进行分析,获得了令人满意的测试结果。所得结果对JMS PTP体系结构的MDB组件的设计和开发具有良好的参考价值。

参考文献

- 1 Lee Hsiu-Hui, Tseng Chun-Hsiung. A Software Framework for Java Message Service Based Internet Messaging System [A]. In: Parallel and Distributed Computing, Applications and Technologies, (PDCAT)2003 Proceedings [C]. 2003, 161~165
- 2 Fabrice K, Laurent P. Distributed wisdom: Toward next-generation middleware? IEEE Distributed Systems Online [J]. 2005, 6 (3):5
- 3 刘小华,陈亚强. J2EE应用开发详解. 北京:电子工业出版社[M],2004
- 4 赵强,乔新亮. J2EE应用开发(WebLogic+JBuilder). 北京:电子工业出版社,2005
- 5 扎德罗兹尼,等. J2EE性能测试. 北京:电子工业出版社,2003