

基于 Hermes 系统的分布式 JAVA 消息服务系统的研究与设计

Research and Design of Distributed JMS based on Hermes

沈正伟 杨东勇 (浙江工业大学软件学院 浙江杭州 310023)
施冬材 (浙江大学计算机科学与技术学院 浙江杭州 310027)

摘要:Java 消息服务(JMS)适合于 Internet 环境下松耦合的分布式计算。现有的集中式的 JMS 实现容易出现消息代理的单点失败和过载。分布式的发布/订阅系统提供的接口与 JMS 的接口不兼容,导致企业应用难以从集中式服务向分布式服务迁移。本文提出基于 Hermes 的分布式 JMS 系统模型,在 Hermes 上创建分布式的消息代理服务,并保留了 JMS 接口。

关键词:消息中间件 JMS 发布/订阅 Hermes

1 引言

Internet 环境下的分布式计算具有大规模和高度动态性的特点,它要求各参与者之间的信息交互应当具有较高的松散耦合。传统面向对象中间件技术的实现基于系统实体间请求/应答模式,客户端和服务端之间的耦合程度高,无法满足 Internet 计算的需求。基于发布/订阅技术的消息中间件(发布/订阅系统)则可以很好地迎合这些需求。发布/订阅系统通常由发布者、订阅者和事件代理组成。订阅者向系统发出订阅,事件代理在收到发布者发布的事件之后,把同订阅匹配的事件发送到相应的订阅者。但分布式的发布/订阅系统提供的接口跟 JMS 的接口并不兼容,导致企业应用难以从集中式向分布式服务迁移。

本文提出了一个基于 Hermes 的分布式 JAVA 消息服务的架构模型,主要描述了模型中的组件构成和拓扑结构,并重点论述了模型的实现策略。

2 Hermes 和 JMS 介绍

发布/订阅系统正逐渐成为国内外研究的热点,比较具有代表性的原型系统有 Gryphon, SIENA, JEDI, Bayuex, Rebeca, 和 Hermes^[1]等。

Hermes 是剑桥大学 Peter 提出的一个发布/订阅中间件系统,它结合了发布/订阅、P2P、数据库、中间件和网络等多个领域的概念,实现了基于事件类型和内容的混合的事件模型,基于 RBAC 的安全控制、基于阻塞控制的路由,并且支持复合事件处理,具有表达能力强、路由效率高、安全可靠等特点。

JMS^[2]是 SUN 公司于 1998 年提出一套旨在统一各种接口的规范,提供访问企业消息系统的标准 API,它便于消息系统中的 Java 应用程序进行消息交换,并且通过提供标准的产生、发送、接收消息的接口简化企业应用的开发。加入目前实现的各种消息中间名字,并指出应用现状。

JMS 是基于主题和内容的混合式的发布/订阅系统,是工业级别的发布/订阅系统实现。Hermes 是基于 Scribe^[3]的分布式的发布/订阅系统实现。Hermes 系统较好地满足了现有 JMS 系统向分布式服务扩展时需要融合的特点,因此我们在 Hermes 系统的基础上实现分布式 JAVA 消息服务系统。

3 JMS - Hermes 系统模型

JMS - Hermes 系统把 JMS 架构加载到 Hermes 系

统中, JMS 服务器通过一定方式作为分布式服务节点加入 Hermes, 实现 JMS 在服务上的分布式特点。对 JMS 客户端而言, 只需要通过原有 JMS 调用接口和服务端进行交互, 保证服务的标准性。模型中还包含了服务定位器的概念。

图 1 描述了 JMS - Hermes 系统模型的拓扑结构。

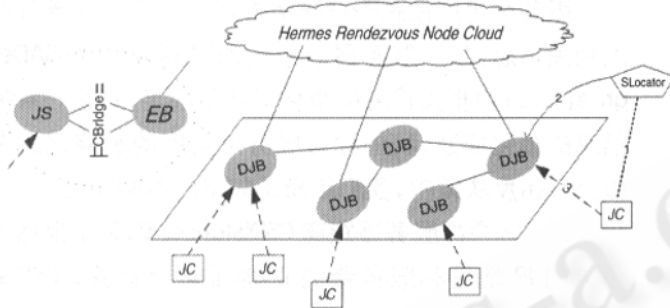


图 1 JMS - Hermes 系统模型拓扑结构图

模型保留了 Hermes 系统和 JMS 架构中的一些组件。如下:

(1) 事件代理 (Event Broker)。Hermes 系统保留组件, 用于 P2P 网络上事件的路由, 需要与 JMS 服务节点进行间接信息交换, 记为 EB。

(2) 集合点 (Rendezvous Node)。Hermes 系统保留组件, 用于描述分布式系统中的主题类型, 记为 RN。

(3) JMS 服务节点 (JMS Server)。JMS 架构保留组件, 负责与 JMS 客户端的通讯, 记为 JS。JMS 服务节点需要与事件代理进行间接信息交换以实行分布式服务。

(4) JMS 客户端 (JMS Client)。JMS 架构保留组件, 完成与 JMS 服务节点连接, 为 JMS 应用提供标准功能接口, 记为 JC。

同时在模型中加入了新的组件, 对新组件的描述如下:

(1) 融合桥 (Merging Bridge)。连接 JMS 服务节点和事件服务节点, 为 JMS 服务与事件代理间的信息交换提供直观工具, 称为融合桥, 记为 MBridge。通过使用融合桥, 可以保证 JMS 服务器实现, 事件代理实现方面作很小的改动即可完成分布式 JMS 的实现。

(2) 服务定位器 (Server Locator)。客户端请求连接时, 通过一定的策略为客户端选择最适合的服务器节点, 与其进行连接。另外, 当与客户端连接的服务器

节点意外出错导致连接失败时, 提供相应的机制重定向连接, 把客户端连向其他服务器, 重定向连接的操作对客户端是完全透明的。提供这种服务器选择策略和连接重定向的节点对象称为服务定位器, 记为 SLocator。

(3) 分布 JMS 代理 (Distributed JMS Broker)。JMS 架构和 Hermes 系统融合后, 既能为 JC 提供 JMS 服务, 又能实现分布式事件路由功能的节点, 称为分布 JMS 代理节点, 记为 DJB。分布 JMS 代理是事件代理、JMS 服务节点和融合桥的组合。事件代理把需要传递给 JMS 服务节点的信息交给融合桥, 融合桥得到具体信息后再传递给相应 JMS 服务节点; 反之亦然。可描述为: $DJB = EB \text{ M} \text{ Bridge} \text{ JS}$, 其中表示信息互换。

4 分布式 JMS 实现策略

在 JMS 产品向分布式过渡时, 对客户端而言, 最重要的因素就是分布式服务网络的透明性, 即: 在分布式环境下, 客户端所能发起的操作接口保持一致。这就要求在客户端初始化连接服务时, 指定单个主机名称和端口号的模式仍需被保留。

4.1 客户端自主选择服务器的缺点

客户端通过指定服务器主机名称和端口号对服务器节点进行选择, 以建立与服务器之间的连接, 这种连接建立方式存在以下缺点:

(1) 客户端需要确保其所连向的服务器节点处于活动和运行的状态, 违背了分布式服务的透明性原则。

(2) 客户端不断地从其已知服务器节点获取连接服务, 导致这些服务器节点的连接负荷过重, 而新加入的服务节点无法被有效利用。

采用服务定位器解决上述问题。服务定位器的主要功能是为客户端找到其可连向的较优服务器节点。

4.2 服务定位器模型

服务定位器中存在一个用于监听客户端连接请求的监听对象, 记为 ConnListener。服务定位器中还需要保持一个服务节点信息列表, 记为 ServerInfoList。列表中的每一项均记录一服务器节点的关联信息 (ServerInfo)。在连接请求到达时, 根据这些相关信息, 服务定位器对 ServerInfoList 中的服务节点进行选择, 得到可用服务器节点。

(1) 定义 1。服务可用度 (Service Availability De-

gree) 是服务器节点信息中用来表征服务器在可用性方面的一个量化特征参数, 记为 SA_{Degree} 。 SA_{Degree} 值表现为 float 型, 值越大, 表示服务器节点的可用性越高。

如果整个分布式服务网络中仅存在一个或者几个服务定位器, 随着连接请求的增多, 这些服务定位器将成为系统的性能瓶颈; 同时, 当所存在服务定位器失败时, 客户端连接请求失败。为此, 我们提出域的概念。

(2) 定义 2。域 (Domain) 是对分布式服务拓扑结构中所有服务节点进行物理分划而得到的概念。把物理空间相近的某组服务节点组合在一起, 就构成了一个域。

我们为 SLocator 设定所属域 (SDomain) 属性, SLocator 仅对域内服务器节点信息进行记录, 为客户端选择域内服务器节点。同一管理域内允许存在多个 SLocator 节点。

综上所述, 得到对服务定位器的形式化描述:

SLocator :: = { SLocatorId, ConnListener, SDomain, ServerInfoList = $\sum_{i=0}^m$ ServerInfo }, m 为 SDomain 内服务器节点数量, 其中

SLocatorId: 标识 SLocator 的字段。

ServerInfo: 服务器节点记录在 SLocator 上的相关信息, 如图 4-1 所示。其中 ServerId (服务器 Id) 标识服务器节点, HostName (主机名) 记录服务器节点的主机名/IP 地址, ListeningPort (监听端口) 表示服务器节点用于监听客户端连接的端口号, ConcurrentConns (现有连接数) 为服务器节点上所存在的客户端连接总数, AvailableConns (可用连接数) 为服务器节点的总连接容量减去现有连接数而得的值, ProcessAbility (处理能力) 是对服务器性能评价的一个指数, 一般由管理员根据具体情况配置而定。

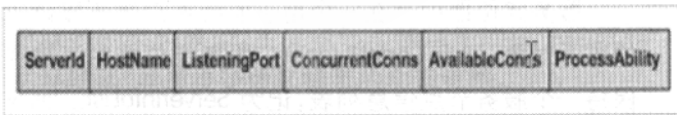


图 2 ServerInfo 表示图

4.3 基于 SLocator 的实现策略

服务定位器主要为客户端提供较优服务器节点的选择功能, 基于 SLocator 的实现策略不涉及对以下内

容的讨论: 域的具体划分和管理、域内多 SLocator 节点的维护、服务器节点加载/离开时 SLocator 的一致性维护策略。

4.3.1 服务可用度决策

服务可用度决策通过对连接请求客户端和服务节点的一系列相关信息实行量化描述, 计算客户端节点存在下的服务器节点服务可用度值。

服务可用度依赖于连接请求客户端和服务节点的相关信息, 某一服务器节点的服务可用度值 $SA_{Degree}(C, S)$ 由相关信息的量化描述所决定, 对这一系列信息的量化描述, 我们称为服务可用度的影响元。在服务可用度决策中, 我们定义了如下三类影响元:

(1) 客户端服务器距离 $CSDistance(C, S)$ 。根据客户端的 IP 地址和服务器的 IP 地址进行计算, $CSDistance(C, S)$ 值越大, 服务可用度越低。

(2) 服务器连接可用率 $CARS$ 。表示服务器连接饱和和程度的值, $CARS$ 值越大, 服务可用度越高。具体计算式为:

$$CAR_s = \frac{AvailableConns_s}{(ConcurrentConns_s + AvailableConns_s)} \circ$$

(3) 服务器处理能力 PAS 。表征服务器性能指标值, PAS 值越大, 服务可用度越高。

每一影响元对服务可用度的影响程度都不同, 我们称表示影响程度的元素为服务可用度的影响因子。跟影响元的类型相对应, 服务可用度决策中存在如下几个影响因子:

(1) $CSDistance$ 影响因子 τ 。表示 $CSDistance$ 对服务可用度的影响程度, 为负值。

(2) CAR 影响因子 ϵ 。表示 CAR 对服务可用度的影响程度, 为正值。

(3) PA 影响因子 λ 。表示 PA 对服务可用度的影响程度, 为正值。

τ, ϵ 和 λ 的具体值大小视系统的具体情况而定。

由上, 得到服务可用度决策表达式为:

$$SA_{Degree}_{(C,S)} = \tau \square CSDistance_{(C,S)} + \epsilon \square CAR_s + \lambda \square PA_s, \text{ 其中 } \square \text{ 表示相乘。}$$

4.3.2 服务选择策略

基于服务可用度决策, SLocator 为请求连接客户端选择可用服务器。在客户端的连接请求到达 SLocator 后, SLocator 作如下处理:

第一步,得到客户端的 IP 地址信息。

第二步,得到 ServerInfoList 列表,通过 SDE,为每一服务器节点计算其 $SADegree_{(c,s)}$ 值,选取 $SADegree_{(c,s)}$ 值最大的服务器节点。

第三步,对所选服务器节点执行 ping 操作,测试服务器节点的实际可用性。

第四步,如果 ping 测试成功,返回所选节点信息给客户端;如果失败,则从 ServerInfoList 列表中删除此节点信息,再返回第二步。

5 总结

本文提出的基于 Hermes 系统的分布式 JAVA 消息服务系统框架,保留了 JMS 接口,解决了分布式发布/订阅系统对 JMS 产品支持不足的问题。模型采用融合桥方式实现 JMS 产品与分布式发布/订阅系统间的集成。采用服务定位器实现客户端的接入,不仅避免了单一服务节点连接拥塞的问题,还为客户端提供服务可靠性的保证。

参考文献

- 1 Pietzuch PR. Hermes: A scalable event-based middleware [Ph. D. Thesis]. University of Cambridge, 2004.
- 2 Sun Microsystems Inc. Java Message Service Specification Version 1.1, April 12, 2002.
- 3 M. Castro, P. Druschel, A. - M. Kermarrec, and A. Rowstron, SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communication, 20(8), pages 100-110, Oct. 2002.
- 4 G Fox, S Pallickara. JMS Compliance in the Narada Event Brokering System. In Proceedings of the 2002 International Conference on Internet, 2002.
- 5 马建刚、黄涛、汪锦岭、徐罡、叶丹,面向大规模分布式计算发布订阅系统核心技术,软件学报,2006.
- 6 汪锦岭,面向 Internet 的发布/订阅系统的关键技术研究[博士学位论文],中国科学院研究生院,2005.