

面向 Web 服务总线集成的分阶段 优先级事件驱动架构

吴众欣¹, 韩彪², 钱德沛^{1,2}, 王永剑²

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 北京航空航天大学计算机学院, 100191, 北京)

摘要: 针对 Web 服务集成过程中分阶段事件驱动架构 (SEDA) 仅考虑服务集成架构的资源消耗, 而对被集成的服务及由其构成的任务资源耗费考虑不足的问题, 提出了分阶段优先级事件驱动架构 (SPEDA). 选取评价指标, 通过熵权法对事件进行优先级评估, 而在阶段内对优先级高的事件给予优先处理; 使用非抢占式规则的排队系统构建集成任务资源消耗的数学模型, 通过合理安排事件的优先等级来降低集成任务资源消耗. 实验表明, SPEDA 采用 3 个优先级别, 其任务资源消耗比 SEDA 降低了 18%, 在考虑了优先级划分粒度的情况下, 5 个级别的集成任务资源消耗比 3 个级别的降低了 21%.

关键词: 服务集成; 资源消耗; 熵权法

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2009)12-0016-05

Staged Priority Event Driven Architecture Oriented to Service Integration Resource Consumption in Web Service Bus

WU Zhongxin¹, HAN Biao², QIAN Depei^{1,2}, WANG Yongjian²

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. School of Computer Science, Beihang University, Beijing 100191, China)

Abstract: The staged event driven architecture (SEDA) only considers the resource consumption of integration architecture, while the consumption of integrated services and integration tasks (which are composed of the integrated services) is not fully considered during the web service integration process. A staged priority event driven architecture (SPEDA) is proposed by estimating the priority of events with the entropy weight method under some special criteria. The high priority events are early handled in stages. The task consumption is modeled in a queuing system with non-preemptive priority rule, and can be decreased by suitably arranging priorities endowed with estimation methods. Experiments and a comparison with SEDA show that the SPEDA decreases 18% resource consumption of tasks with 3 priority classes. When the size of priority classes grain is considered, a comparison with 3 priority classes shows that the SPEDA with 5 priority classes decreases 21% resource consumption of tasks.

Keywords: service integration; resource consumption; entropy weight method

随着软件即服务、平台服务的发展, 服务组合中的服务集成已成为面向服务体系架构 (SOA) 应用关注的热点. 服务总线可以消除服务集成过程中的冗余重复开发, 提高原系统的重用度^[1]. 随着服务组

收稿日期: 2009-04-01. 作者简介: 吴众欣(1974—), 男, 博士生; 钱德沛(联系人), 男, 教授, 博士生导师. 基金项目: 国家高技术研究发展计划资助项目 (2006AA01A118, 2006AA01A124, 2009AA01Z144); 科技部国际合作重点项目 (2006DFA11080).

件接入数量的增加,服务总线上的工作负载也随之增长,这就需要用新的可伸缩框架来提高总线性能^[2].

Welsh 提出的分阶段事件驱动架构(Staged Event Driven Architecture, SEDA)为 Web 服务集成中的服务总线提供了可伸缩能力^[3],Web 服务总线多采用该架构来提高并发的处理能力.

服务集成系统分为两部分,即集成组件架构和被集成的服务,相应的系统资源消耗也需要从这两个部分来考虑. SEDA 架构只考虑了服务集成架构组件的资源消耗,而对被集成的服务,即各个服务构成的集成任务资源耗费考虑不足,它仍需关注以下问题:①阶段内处理的事件是否因占用相同的资源而不能形成并发;②事件传递的数据是控制数据还是应用数据;③根据阶段内事件处理的时间长短,改变事件处理顺序是否对任务资源消耗有影响.

本文的分阶段优先级事件驱动架构(Staged Priority Event Driven Architecture, SPEDA)考虑了集成服务的资源消耗,不仅具有松耦合的任务分阶段处理能力,还具有对集成服务资源消耗的控制能力,通过合理安排事件的优先等级,可以降低集成任务的资源消耗.

1 SPEDA 架构

SPEDA 架构如图 1 所示,其将集成任务分解为阶段网络,再通过事件通信将各个阶段连接在一起,而用户只需关注阶段内的处理逻辑和阶段间的连接逻辑. 每个阶段是一个自治体,它拥有自己的事件队列、处理器(由用户提供的阶段内处理逻辑执行者)和带有反馈功能的线程池. 如果在一个阶段完成工作,将会触发一个事件,该事件会传到下一个阶段的事件队列中,而准入控制器决定事件是否入队. SPEDA 架构采用熵权法,在一定的指标集合作用下对事件进行优先级评估,优先级高的事件在阶段中能得优先处理. 队列对到来的事件进行优先级排序,按照事件的优先级评估值安排优先处理等级.

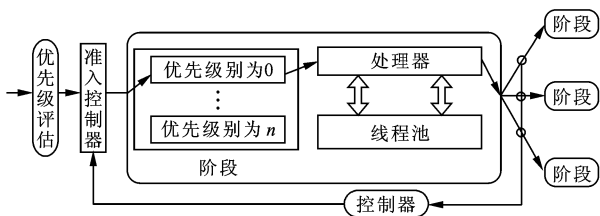


图 1 SPEDA 架构

线程池的大小根据优先级事件队列的长短进行动态调整. 基于反馈机制,通过对事件准入控制器的控制,进行输入事件的流量调节,使得 SPEDA 针对负载具有自动适应能力和负载均衡控制能力.

SPEDA 的集成任务资源消耗的数学模型,可以用来分析降低任务资源消耗的因素.

2 SPEDA 集成任务资源消耗模型

将 SPEDA 架构中的每个阶段抽象为排队论中具有非抢占规则的独立系统. 依照 Little 定理, $N = \lambda W$, 其中 λ 是一个独立阶段的事件到达率在时间段 $[0, t]$ 上的平均值, W 是事件在阶段中的平均耗费时间, N 是阶段中在时刻 t 持有的事件个数. 相关参数定义如下.

W_i : 第 i 个事件在阶段中的等待时间.

R_i : 对于第 i 个事件的残留处理时间. 当第 i 个事件到达时,事件 j 还在处理中,则 R_i 就是事件 j 达到完成状态所剩余的处理时间.

X_i : 第 i 个事件的处理时间.

N_i : 第 i 个事件到达时阶段中事件的个数.

$\bar{X}$: 平均处理时间, $\bar{X} = E\{X\} = \frac{1}{\mu}$, 其中 X 是

处理时间集合.

μ : 单位时间内处理事件的个数.

$\bar{X}^2$: 处理时间的第二矩, $\bar{X}^2 = E\{X^2\}$.

从上面定义可知

$$W_i = R_i + \sum_{j=i-N_i}^{i-1} X_j \quad (1)$$

对式(1)左右两端取期望值,可以得到

$$E\{W_i\} = E\{R_i\} + E\left\{\sum_{j=i-N_i}^{i-1} E\{X_j \mid N_i\}\right\} = E\{R_i\} + \bar{X}E\{N_i\}$$

当 $i \rightarrow \infty$, 可得到

$$W = R + \frac{1}{\mu}N \quad (2)$$

式中: R 是平均残留时间.

依照 Little 定理, 将 $N = \lambda W$ 带入式(2), 可以得到 $W = R + \frac{\lambda}{\mu}W$, $\rho = \frac{\lambda}{\mu}$, ρ 称为利用因子. 于是可得到

$$W = \frac{R}{1-\rho} \quad (3)$$

如果采用积分法, 并认为时间平均是其整体平均, 则有

$$R = \frac{1}{2} \lambda \bar{X}^2 \quad (4)$$

假设阶段内处理事件的时间服从 General 分布,到达该阶段的事件被划分为 n 个不同的优先级,则级别 1 表示最高优先级. 级别 k 的事件到达率为 λ_k ,处理时间的第一矩为 $\bar{X}_k = 1/\mu_k$,第二矩为 $\bar{X}_k^2$. 又设所有级别的事件到达过程是与处理时间无关的独立泊松过程.

非抢占优先级的规则定义为:正在接受处理的事件允许在没有任何干扰的情况下完成处理过程,即便此时有较高级别的事件进入处理阶段. 在队列中,最高优先级的事件被优先处理. 综上,定义参数如下.

N_Q^k : 队列中级别为 k 的事件平均数目.

W_k : 级别为 k 的事件平均排队时间.

$\rho_k = \lambda_k / \mu_k$: 级别为 k 的事件对系统的利用率.

R : 平均残留处理时间.

假设系统的总利用率小于 1,即 $\rho_1 + \rho_2 + \dots + \rho_n < 1$,对于最高优先级(级别为 1)的事件,排队延迟时间为

$$W_1 = R + \frac{1}{\mu_1} N_Q^1$$

利用 Little 定理 $N_Q^1 = \lambda_1 \bar{W}_1$,可导出

$$W_1 = \frac{R}{1 - \rho_1} \quad (5)$$

对于次高优先级(级别为 2)的事件,排队延迟时间与最高优先级的情况一般是相似的,但当包含次高优先级的事件到达队列时,已有一个事件正在处理(非抢占),则有

$$W_1 = R + \frac{1}{\mu_1} N_Q^1 + \frac{1}{\mu_2} N_Q^2 + \frac{1}{\mu_1} \lambda_1 W_2 \quad (6)$$

依据 Little 定理,可以得到

$$W_2 = \frac{R}{(1 - \rho_1)(1 - \rho_1 - \rho_2)} \quad (7)$$

依次类推,在 $k > 1$ 都适用的队列中,等级为 k 的事件等待时间为

$$W_k = \frac{R}{(1 - \rho_1 - \dots - \rho_{k-1})(1 - \rho_1 - \dots - \rho_k)} \quad (8)$$

级别为 k 的事件平均延迟时间为

$$T_k = \frac{1}{\mu_k} + W_k \quad (9)$$

则任务资源消耗可定义为

$$C_{\text{task}} = \sum_{i=1}^k C(T_k) \quad (10)$$

式中: $C(T_k)$ 是阶段资源消耗函数,随时间延迟增加

而增大. 在阶段中,执行子任务和保持子任务状态耗费的阶段资源随延迟时间的增长而增大,所以任务资源的消耗,是系统中各个阶段的资源消耗的总和.

如果对处理时间较短的事件赋予较高的优先级,阶段中的事件平均延迟就比较小,任务资源消耗也就会降低. 按照优先级来分级粒度,分级越细,处理时间短的事件就越容易被优先执行. 当处理事件很多时,粒度细的优先级分级带来的事件平均延迟比较小.

3 事件优先级评估方法

熵权法是在给定评价对象集后,分析各指标竞争的相对激烈程度^[4]. 事件的优先级评估可以分为 3 大类:①冲突资源,即各个阶段接收到事件后就要执行自身阶段所要处理的应用逻辑,此时需要考虑该事件和其他同时执行体的事件所占用的资源是否会发生冲突(占用资源相同,不会并发执行,关于应用占用资源的判定过程,请参考文献[5]);②阶段内事件处理时间,即事件在阶段内平均的延迟时间;③数据类型,即对于不同的数据类型,应区别对待,并给予不同的重视程度,到达队列的数据类型可分为控制类、应用类和混合类 3 种.

熵权法的优先级评估过程如下.

(1)评价指标有 m 个,需测定 n 次,指标值构成判断矩阵 $A = (r_{ij})$, $i = 1, 2, 3, \dots, m$,某一类别下的指标为 $V_k = (r_{kj})$, $i \leq k \leq m$.

(2)对判断矩阵进行标准化. 标准化类型分为效益型和成本型. 对于效益型,有

$$R_i^* = \max\{r_{ij}\} \quad r'_{ij} = \frac{r_{ij}}{R_i^*}$$

由此可将 A 标准化为 B , $B = (r'_{ij})$, $i = 1, 2, 3, \dots, m$. 对于成本型,有

$$R_i^* = \min\{r_{ij}\} \quad r'_{ij} = \frac{R_i^*}{r_{ij}}$$

由此可将 A 标准化为 B , $B = (r'_{ij})$, $i = 1, 2, 3, \dots, m$.

(3)第 i 个指标的熵定义为

$$H_i = -k \sum_{j=1}^n f_{ij} \ln f_{ij} \quad i = 1, 2, 3, \dots, m \quad (11)$$

式中: $f_{ij} = \frac{r'_{ij}}{\sum_{j=1}^n r'_{ij}}$; $k = \frac{1}{\ln n}$.

假定 $f_{ij} = 0$ 时, $f_{ij} \ln f_{ij} = 0$.

(4)用 H_i 来确定评价指标 i 的评价权值 ω_i ,则有

$$\omega_i = \frac{1 - H_i}{m - \sum_{i=1}^m H_i} \tag{12}$$

式中: $0 < \omega_i < 1, \sum_{i=1}^m \omega_i = 1$.

(5)熵权 ω_i 规格化属性矩阵为

$$A = \begin{bmatrix} \omega_1 r_{11} & \cdots & \omega_1 r_{1n} \\ \vdots & \ddots & \vdots \\ \omega_m r_{m1} & \cdots & \omega_m r_{mn} \end{bmatrix}$$

(6)事件的优先级为

$$P_{V_k} = \sum r_{ij} \omega_i$$

根据熵权法可以比较出各类事件的优先级.

4 实 验

在扩展 ServiceMix 的 Web 服务总线平台上搭建的实验环境:1 个 Web 服务总线服务器节点,若干个服务组件节点. 硬件配置:4 个 2.4 GHz 的 CPU,2 GB 内存,80 GB SCSI 硬盘,节点间以 1 GB 带宽的以太网互联. 软件环境:Linux 操作系统, JDK1.5.

评价指标为 3 组:资源组、执行时间组、数据类型组. 资源组反映各个事件运行时占用的资源情况,如果不是密集型程序(CPU, memory, I/O, network),该指标值为 4;如果特征因子属于一种密集型程序因子,指标值减 1^[5]. 事件执行时间为起始到结束的执行时间. 在数据类型中,控制命令赋值为 4,混合数据赋值为 3,应用数据赋值为 2. 评价指标如表 1 所示.

表 1 评价指标表

资源赋值				数据类型赋值			执行 时间
CPU	内存	I/O	网络	混合 数据	控制 数据	应用 数据	
1,0	1,0	1,0	1,0	4	2	3	T

3 类事件得到的指标数值可构成判断矩阵,如表 2 所示.

表 2 指标值构成的判断矩阵

事件 序号	资源 赋值	执行时间/ms	数据类型赋值
1	2	2 400	2
2	1	1 600	3
3	1	960	4

事件占用的资源越少越好,执行时间越短越好,如果数据类型是控制命令,则各个指标的理想数值如表 3 所示.

表 3 控制命令下指标的理想数值

资源赋值	执行时间/ms	数据类型赋值
4	960	4

执行时间属于成本型矩阵,占用资源是效益型矩阵,则相应的标准化矩阵如表 4 所示. 通过标准化矩阵计算获得的熵值如表 5 所示.

表 4 标准矩阵

事件序号	资源赋值	执行时间/ms	数据类型赋值
1	0.50	0.4	0.50
2	0.25	0.6	0.75
3	0.25	1.0	1.00

表 5 熵值

熵函数	资源赋值	执行时间/ms	数据类型赋值
H_i	0.946 4	0.937 1	0.965 5

综上,可以得出熵权值,如表 6 所示. 由熵权值推导出的规格化属性矩阵见表 7. 由表 7 计算出的事件优先级如表 8 所示.

表 6 熵权值

熵权函数	资源赋值	执行时间/ms	数据类型赋值
ω_i	0.009 6	0.011 3	0.006 2

表 7 规格化属性矩阵

事件序号	资源赋值	执行时间/ms	数据类型赋值
1	0.004 8	0.004 5	0.003 1
2	0.002 4	0.006 8	0.004 7
3	0.002 4	0.011 3	0.006 2

表 8 事件优先级数值

事件序号	1	2	3
优先级数值	0.012 4	0.013 9	0.019 9

优先级数值高的事件优先执行. SPEDA 架构下集成任务的事件响应时间变化如图 2 所示,可以看出,整体集成任务的完成时间没有改变,即集成组件架构的资源消耗是不变的. SPEDA 的事件平均延迟如图 3 所示,可见延迟有所减小. 根据式(10),如果任务资源消耗与时间呈现线性关系,SPEDA 集成任务资源耗费会降低 18%. 注意:本文关心的

是两个对象,一个是集成组件架构,另一个是集成的服务,也就是集成任务的执行体.所以,SPEDA 架构考虑了更多的资源消耗对象,说明其考虑问题更加全面.

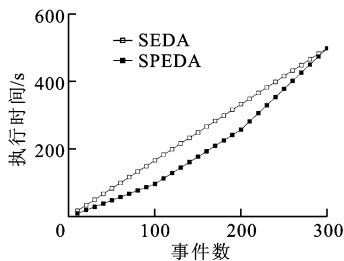


图2 整体任务执行时间

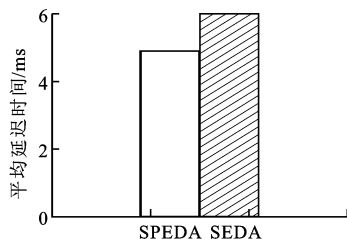


图3 平均延迟时间

经过熵权评估后,将优先级数值为 0.011 9、0.017 9、0.027 8、0.041 7、0.065 5 的 5 个事件分成 3 个优先级(根据不同的优先级粒度,将优先级分为 3 组),各个分组之间的优先级关系为

$$(0.041\ 7 \cup 0.065\ 5) > (0.017\ 9 \cup 0.027\ 8) > (0.011\ 9)$$

不同优先级划分粒度的平均延迟时间如图 4 所示,可以看出,5 个优先级比 3 个优先级的平均

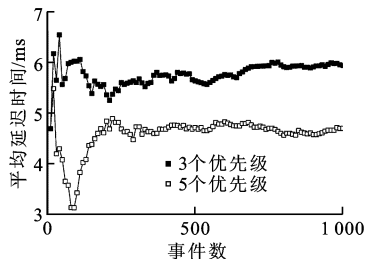


图4 不同优先级划分粒度的平均延迟时间

延迟时间要小,也就是利用较细的优先级粒度,可以降低集成任务资源的耗费.根据式(10)知,如果任务资源消耗与时间呈线性关系,任务资源消耗可以降低 21%.在总任务产生的事件数比较多的情况下,应该选择细粒度的优先级划分方法.

5 结束语

SPEDA 不仅考虑了集成架构组件的资源消耗过程,同时也考虑了集成服务的系统消耗,它不仅具有松耦合的任务分阶段处理能力,也具有对集成服务更强的资源控制能力,因此能降低集成任务资源的耗费.

参考文献:

- [1] CHAPPELL D. Enterprise service bus[M]. Sebastopol, USA: O' Reilly Media, 2004.
- [2] UENO K, TATSUBORI M. Early capacity testing of an enterprise service bus [C]//Proceedings of International Conference on Web Service. Los Alamitos, CA, USA: IEEE Computer Society, 2006: 709-716.
- [3] WELSH M D, CULLER D. Adaptive overload control for busy internet servers [C]//Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems. Berkeley, USA: USENIX Association, 2003:4.
- [4] 胡周君, 胡志刚, 李林. 一种基于性能评估的元任务调度算法 [J]. 西安交通大学学报, 2008, 42(8): 972-976.
- [5] HU Zhoujun, HU Zhigang, LI Lin. Performance evaluation based meta task scheduling algorithm [J]. Journal of Xi'an Jiaotong University, 2008, 42(8): 972-976.
- [5] WANG Ke, WU Zhongxin, LUAN Zhongzhi, et al. Reducing the cluster monitoring workload by identifying application characteristics [C]//7th International Conference on Grid and Cooperative Computing. Los Alamitos, CA, USA: IEEE Computer Society, 2008: 525-531.

(编辑 苗凌)