服务请求排队算法①

Service-Request Queue Arithmetic

危 达　李建华　李 曼　叶 云　(中南大学 信息科学与工程学院 湖南 长沙 410075)

摘 要： 当有很多用户请求某一个特定服务提供者(SP)提供的服务，可能造成排队等候现象，已有技术普遍使用先来先服务(FCFS)的原则对服务请求(SQ)进行排队。提出了一种基于语义匹配的优先级排队算法，和一种基于语义匹配和先来先服务原则的综合排队算法，协调处理高优先级别 SQ 和先到 SQ。从数理上对服务请求综合排队算法进行了评估，证明算法只增加了常数时间消耗。描绘出了实现该算法的技术框架。

关键词： 服务提供者 服务请求 排队算法 先来先服务 语义匹配

当有很多用户请求某一个特定 SP 提供的服务，可能造成排队等候现象，已有技术普遍使用 FCFS 原则对 SQ 进行排队[1]。当某些后到的 SQ，要求在先来的 SQ 之前先得到处理，就需要一种基于优先级的排队算法来实现此要求。

1 引言

本文首先提出基于语义匹配的优先级排队算法，然后同时考虑语义匹配和 FCFS 原则，提出综合排队算法。综合排队算法致力于实现高优先级别 SQ 和先到 SQ 之间的协调，更合理的使用 SP 提供的服务，提高资源利用效率。

基于语义匹配的优先级排队算法，基本思想是通过 SQ 附加语义信息，SP 通过将这些语义信息与其本身所拥有的标签库中的信息匹配，匹配程度高的 SQ 将获得高的优先级[2]。

基于语义匹配和 FCFS 原则的综合排队算法，基本思想是在上一种算法的基础上，同时考虑 FCFS 的原则，得到一个优先值计算函数。每一个 SQ 如果不能即时被 SP 处理，通过此函数得到一个优先值，然后根据此优先值排队等候，实现合理有序使用 SP 提供的服务。

本文提出两种 SQ 排队算法，针对综合排队算法做出评估，论证该算法在时间消耗等方面，与原有的 FCFS 排队算法并没有很大的区别，却能够实现更合理的利用 SP 提供的服务，因此将会具有实际效益。本文最后论述了 SQ 综合排队算法的实现机制，描绘出了实现该算法的技术框架。

2 FCFS原则

FCFS 原则是指一队 SQ 在等候同一 SP 提供服务时，先来的 SQ 排在队列的的前面，而后来的 SQ 则排在队列的后面。

服务请求到达顺序位置值：假设 SQ 队列已有 p 个请求在等待，则这 p 个请求的到达顺序位置值按照先后顺序分别为：

$$1-\frac{0}{p},\ 1-\frac{1}{p},\ \cdots\cdots,\ 1-\frac{p-1}{p}$$

若第(p+1)个 SQ 到来，则此全部(p+1)个 SQ 到达顺序位置值分别为：

$$1-\frac{0}{p+1},\ 1-\frac{1}{p+1},\ \cdots\cdots,\ 1-\frac{p}{p+1}$$

其中，第(p+1)个 SQ 的到达顺序位置值为 $\frac{1}{p+1}$。

3 语义匹配

SQ 的语义描述信息与 SP 的标签库中的信息匹配，匹配程度高的 SQ 将获得高的优先级。实现语义匹配必须要求 SQ 消息附带语义信息，SP 有标签库，

① 收稿时间:2008-11-05

© 中国科学院软件研究所 http://www.c-s-a.org.cn

以下论述了这个两个对象的工作机制。

SQ 的语义描述信息：SQ 消息附加的语义信息，描述了 SQ 者请求该服务的要求、目的等信息。具体实现是改造 SQ 的 SOAP 消息头部分，使用这部分来传输语义信息，语义信息采用单个分词的形式，这样做的目的是简化本算法的复杂度，方便与 SP 标签库中的信息匹配。SOAP 消息头的典型应用是用来传送上下文的数据，它有两种方式，分别是显示和隐式消息头。在显式消息头中，消息头的所有信息会添加给服务的外部接口，作为附加的参数提供给客户端。隐式消息头的好处是，消息头信息并不是服务外部接口的一部分，因此不会影响服务的基本功能。本算法采用了 SOAP 消息的隐式消息头，在消息头放置分词形式的语义信息，在客户端编写程序读取这些语义信息，然后与 SP 的标签库中的信息匹配[3]。

SP 的标签库：标签库的信息由 SP 提供，描述了该服务的要求、适用范围和其他一些特征，符合这些特征的 SQ 应该被优先处理，而不会因为其到来得比较晚而处于长久的等待中。标签库的信息也是采用单个分词的形式进行描述。

4　基于语义匹配的优先级排队算法

基于语义匹配的优先级排队算法的过程是：

1) SQ 到达总线或者 SP。

2) 获取 SQ 消息头部分的 n 个分词的语义信息。(n 个语义描述分词至少多于分词最少要求数目 n_0 个，n_0 的大小由 SP 决定，合适的 n_0 直接决定匹配的效果)

3) 将此 n 个分词的语义信息与 SP 的标签库中的 N 个分词进行匹配，若其中有 n_1 个分词出现在标签库中，$(n-n_1)$个分词没有出现在标签库中，则该条 SQ 的语义匹配度为。因为 $0 \leqslant n_1 \leqslant n$，因此语义匹配度值是一个 0 到 1 之间的实数。

4) 根据语义匹配度高低将该条 SQ 插入到已有的 SQ 等候队列的相应位置。

5) 等候下一条语义请求的到来。

图 1 是该算法的过程示意图。

5　基于语义匹配和FCFS原则的综合排队算法

基于语义匹配和 FCFS 原则的综合排队算法，是在同时考虑 SQ 的语义信息和请求到来先后顺序的基础上提出来的。综合考虑两种因素，生成 SQ 的优先值。优先值大的 SQ 将被放置在等候 SP 的队列前端。SQ 优先值是一个数值在 0 到 1 之间的数字。

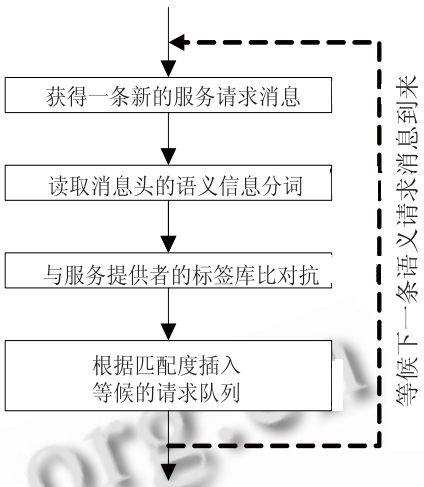

图 1　基于语义匹配的 SQ 排队算法过程示意图

SQ 的优先值为 0 时，即此 SQ 将不被 SP 服务，这种结果出现的情况是，SQ 所附加的语义信息在 SP 的标签库中找不到任何匹配的分词，且 SQ 到来的顺序为无穷个 SQ 的下一个，即 SQ 除非没有到达 SP。因此可以得出结论，一个到达了 SP 的 SQ，其所具有的优先值一定会大于一个无穷小的正数，不会等于 0。

如果一个 SQ 来自一个特殊的 SQ 者，要求其一旦到达 SP，这个请求就必须得到满足，即所具有的优先值应该等于 1。然而，这种情况是不会出现在处于服务协作双方的环境下，因为 SQ 者和 SP 是在全局可见环境下对等消息交换和相互作用的规则起作用的体系架构中工作，SP 不会因为 SQ 者的无条件抢占要求而改变自己的工作安排。SQ 者和 SP 是一种互相协商合作的对等关系。因此一个 SQ 优先值的最大值是一个无穷接近于 1 并且小于 1 的正数。

显而易见，一个 SQ 迫切要求得到服务，它的语义匹配程度，与其到达 SP 并准备进入等候队列的时间先后顺序是没有任何关联的。SQ 者发出一个针对特定 SP 的服务，不是根据环境状况进行服务选择，因此本文不关心是否还有其他 SP 的服务能更快的满足其要求。SQ 语义匹配度和 SQ 先后顺序之间没有联系，为了维持优先级值的平衡，它们产生影响的系数之和应该维持在一个稳定的范围，设其系数之和应该为 1。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

于是，本文以下提出了在众多 SQ 者对特定 SP 请求服务时，基于语义匹配和 FCFS 原则的综合排队算法的优先值数学模型。

基于语义匹配和 FCFS 原则的综合排队算法的优先值函数：

$$F(X,Y) = AX + BY \quad (1)$$

F(X,Y)：SQ 优先值函数

X：语义匹配度，$0 \leqslant X \leqslant 1$

Y：请求到达顺序位置，$0 < y \leqslant 1$

A：语义匹配度权值，$0 < A < 1$

B：请求到达顺序位置权值，$0 < B < 1$

$A + B = 1$

经数理推导，(1)式可以变化为

$$F(X,Y) = AX + (1-A)Y \quad (2)$$

即：

$$F(X,Y) = Y + (X-Y)A \quad (3)$$

通过(3)式代入语义匹配度的值 X'和请求先后顺序的值 Y'，优先值函数 F(X,Y)的求值变为对

$$F(X',Y')=E(A)=Y'+ (X'-Y')A \quad (4)$$

式求 E(A)的均值。一般认为 A 的分布是接近于 $\mu = \frac{1}{2}$、$\sigma^2=0$ 的一维正态分布 $N(\mu, \sigma^2)$。

因此，基于语义匹配和 FCFS 的综合排队算法，对于一个特例求值的过程为：

假设：

$A = \frac{1}{2}$(取 A+B=1 的中间值)

$X' = \frac{5}{10}$(SQ 语义信息共有 10 个分词，其中有 5 个匹配)

$Y' = 1 - \frac{7}{7+1}$ (在新的 SQ 到来前，请求队列已有 7 个等候请求)

则：

$$F(X',Y') = Y'+(X'-Y')A = \frac{1}{2}$$

于是，特例值所代表的 SQ 以优先值为 $\frac{1}{2}$，按从高到低的顺序插入到已有的等候队列中。

6　综合排队算法的评估

基于语义匹配和 FCFS 原则的综合排队算法，首先要构造 SP 的标签库，获得 SQ 的语义信息，然后对两者进行匹配，获得优先级，同时考虑 SQ 到达 SP 的先后顺序，使用优先值函数获得此 SQ 的综合优先值，本算法能够更合理有序的帮助 SP 提供服务。以下通过数理推导论述了本算法比原有的 FCFS 算法在时间消耗上仅增加了常数单位的时间。

请求队列最开始没有 SQ 在等待，当第一个需要等待的 SQ 到达，需要花费 t_1 的时间读取消息头的语义信息，需要花费 t_2 的时间与标签库分词进行匹配，需要 t_3 的时间计算优先值，然后进入等待状态。再来需要等待的服务，完成以上操作的时间加上排队等候的平均时间，与刚进入等待队列就进行等候的 SQ 是一致的，这个平均等待时间用 t'表示。若等候队列中保持有 n 个 SQ，则使用综合排队算法的所有请求总时间消耗为$(t_1+t_2+t_3+nt')$，使用 FCFS 的总时间消耗为 nt'，如果 n 足够大，则算法多消耗的时间只是一个常数级的时间$(t_1+t_2+t_3)$，却很好的提高了合理有序使用 SP 提供的服务。以下是总时间消耗的对比示意图：

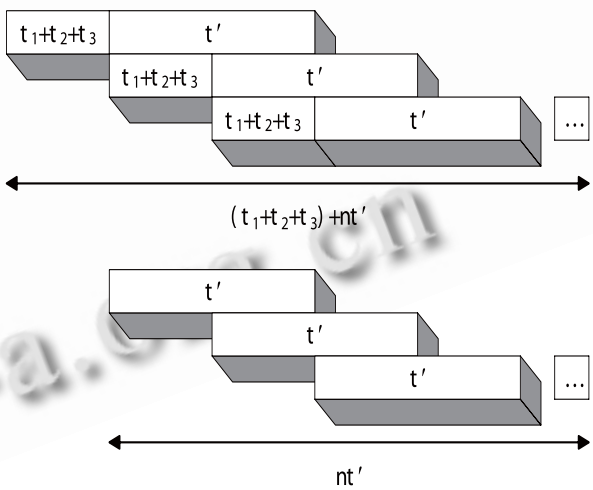

图 2　综合排队算法(上)和 FCFS 算法(下)的时间消耗对比示意图

7　综合排队算法的实现机制

本算法需要实现以下程序功能：

1) 实现读取 SQ 语义信息的功能；

2) 构造 SP 的服务信息标签库；

3) 实现语义信息分词和 SP 标签库分词匹配的功能；

4) 综合匹配度和到达顺序计算 SQ 优先值并重新组合 SQ 等候队列。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

因此至少三个组件是必须实现的，第一个是 SP 的服务信息标签库，第二个是读取 SQ 消息语义信息并和标签库作匹配的组件，第三个是按照基于语义匹配和FCFS原则的综合排队算法计算SQ的优先值并重新组合 SQ 等候队列的组件。

以上组件如果由 SP 实现，则增加了其负担，不符合面向服务架构的松散耦合特征。在基于总线的面向服务的架构中，总线具有消息路由、消息转换、协议中介和事件处理等功能，本算法功能可以通过在企业服务总线上做出扩展来实现。

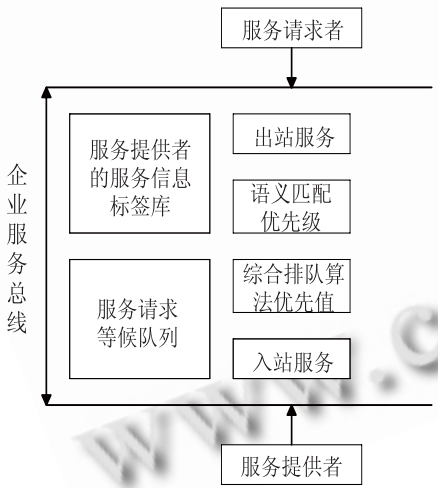

图 3 总线机制实现综合排队算法的技术框图

SQ 消息到达企业服务总线，经过出站服务和入站服务，到达 SP。进行总线扩展的办法是在总线已实现 FCFS 算法的基础上，在总线上部署 SP 的服务信息标签库和 SQ 等候队列组件的容器，在出站服务和入站服务之间增加语义匹配组件和计算优先值组件。

语义匹配优先级组件读取请求消息的语义信息分词，与总线上部署的 SP 的服务信息标签库匹配，然后再根据已有的 SQ 队列组件排队情况，由综合排队算法优先值组件计算 SQ 的优先值。根据优先值，将 SQ 插入到 SQ 等候队列组件的等候队列中。如图 3 所示。

参考文献

1 Papazoglou MP, Traverso P, Dustdar S, Leymann F. Service-Oriented Computing: State of the Art and Research Challenges. IEEE Computer Society, November 2007.

2 Martin D, Domingue J. Semantic Web Services. IEEE INTELLIGENT SYSTEMS, 2007,1541-1672/ 07:12－17.

3 实现隐式和显式 SOAP 消息头. http://www.ibm.com/developerworks/cn/webservices/ws-tip-headers.html, 2005.

© 中国科学院软件研究所 http://www.c-s-a.org.cn