

实时环境下一种混合广播调度策略

吴海 石磊 卢炎生

(华中科技大学计算机学院 武汉 430074)

摘要 在实时的非对称通讯环境下,自适应混合广播策略能够根据数据的请求模式以及时间限制等特征来动态分配周期广播和按需广播的带宽比例。将这一策略推广到基于事务的多数据项广播调度中,同时引入“分布式广播”思想,以更精细的周期广播粒度来解决过长的周期广播与事务及数据的实时要求之间的矛盾,并且动态分配时间槽。实验结果表明:改进的调度策略更适合于实时环境下的数据广播,具有更低的事务失败率以及更小的上行信道负荷。

关键词 混合广播策略,移动事务,分布式广播,事务失败率

Hybrid Broadcast Strategy on Real-time Environment

WU Hai SHI Lei LU Yan-sheng

(Computer College, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract In time-critical asymmetric communication environment, the adaptive hybrid broadcast—TC-AHB can dynamically assign the bandwidth ratio for periodic broadcast and on-demand broadcast according to data-requested mode and time limit. The paper extended the strategy to the broadcast based on transaction scheduling, adopted an idea called distributed broadcast to solve conflict between a long periodic broadcast interval and real-time requirement of data and transactions and assigned the time slot dynamically. The result of experimentation indicates that the strategies improved on adaptive hybrid broadcast mode are more suitable for data broadcast application in real-time environment and have lower transaction failure ratio and lower uplink load.

Keywords Hybrid broadcast strategy, Mobile transaction, Distributed broadcast, Transaction failure ratio

1 前言

在移动数据库系统中,为了支持大量用户并发访问服务器上的数据,有效地利用通信带宽,数据广播是一种常见的数据分发模式。由服务器将数据库中被大多数用户频繁访问的数据(即“热点数据”)组织起来向空中广播,移动用户从空中获取数据。常见的广播方式有静态广播、动态广播和混合广播三种。混合广播结合了静态广播和动态广播的优点,上行信道用于传输在广播程序中没有被满足的用户数据访问请求;下行信道带宽分为静态广播和动态广播两部分,其中静态广播周期地广播“热点数据”,动态广播根据用户需求动态地广播数据。

实时移动环境中移动用户对数据库数据的一次访问称为移动事务。移动事务对数据的访问概率、访问兴趣即数据的请求模式会随着时间不断变化,并且对数据具有实时性要求,即数据和事务往往具有时间限制,超过请求的时限,事务将会夭折,数据将会贬值,有时甚至会带来灾难性的后果;服务器不仅要动态地选择数据广播,还要合理地调度广播数据,根据系统负荷自适应地分配下、上行信道的带宽,使得移动实时事务失败率最低、用户平均访问时间最短。

2 混合广播调度策略

F. C. Jesus 提出一个自适应混合数据广播策略^[1],分别应用 PINOPT 算法和 EDF 算法广播数据。W. C. Lee 和 Q. L. Hu 等提出根据系统工作量选择合适的信道使用方式^[2],提出一个动态信道分配算法,以获得系统最优的信道分配方案和最短的用户访问时间。

文献[3]提出一种自适应混合广播调度策略 TC-AHB,该策略将混合广播策略拓展到移动实时数据库领域,能够根据数据的请求模式及数据的截止期等信息动态调整带宽分配,并且该策略使用采样技术来获取“热点数据”的请求模式。

2.1 TC-AHB 调度策略

TC-AHB 调度策略主要分成周期广播数据调度、按需广播数据调度和数据请求模式的采样 3 个部分。

2.1.1 周期广播的数据调度

周期广播调度使用待调度数据集和本次调度数据集两个全局存储结构。首先根据周期广播带宽、数据的请求次数、数据所需带宽等参数从待调度数据集中选取数据到本次调度数据集,然后根据数据已经占有的时间槽数目和加权截止期等参数来设置数据的具体广播时间。

到稿日期:2008-06-03 本文受国家部委重点科技攻关项目(项目编号:513150402)资助。

吴海(1972—),男,博士生,讲师,研究方向为移动环境下的数据库技术,E-mail:hust_wuhai@163.com;石磊 硕士研究生,研究方向为移动数据库;卢炎生(1949—),男,博士生导师,研究方向为移动数据库、软件测试、构件技术。

2.1.2 按需广播的数据调度

对于待调度数据集中的剩余数据, TC-AHB 策略采用 EDF-Batch 策略进行调度。由于采用周期广播带宽比例上限阈值, 部分请求的实时性需求在此按需广播周期中得到满足。

2.1.3 数据请求模式的采样技术

TC-AHB 调度策略的采样思想为: 在周期广播的末端, 阻塞那些有可能被剔除出本次调度数据集中的“热点数据”的调度。部分用户由于无法在这一时段获取这些“热点数据”, 从而通过上行信道提出请求, 调度模块通过这些请求以及采样时间等参数来估算数据项的请求模式。在每一广播周期, 本次调度数据集中数据项的请求频率都将有一衰减, 衰减因子一般为 0.7~0.9。从而保证某些“热点数据”存在被剔除出调度数据集的可能性, 又保证了这些请求模式的光滑延续性。

2.2 基于事务的调度

基于事务的调度指以事务存取的数据项集合作为完整的广播调度单位, 只有当事务存取的多个数据项全部调度成功时, 事务才调度成功。

K. Lam 提出 EQSD 和 RP 的算法^[4], 明确考虑了事务包含的数据项数目和定时限制, 使得请求失败率最低和平均响应时间最短。J. L. Huang 通过分析多信道广播环境中多数数据项广播的特性, 得出平均访问时间的理论特征, 采用遗传算法来产生数据广播程序^[5]。Y. D. Chung, Y. I. Chang 和 G. L. Lee 等提出 QEM 以及相关改进算法^[6,7], 将一个客户申请的所有数据项尽可能一起调度来减少平均访问时间。刘云生等提出 EDF-T 等调度算法^[8], 考虑了事务包含的数据项数目和时间限制对广播调度的影响, 有效地保证了事务调度的成功率, 适用于实时系统的数据广播。但是以上算法都没有考虑到数据访问模式的变化。

3 Improved TC-AHB 调度策略

TC-AHB 策略相对于固定周期广播的带宽的混合调度, 请求成功率有了提高。但在实际应用中, 该策略仍然可以做 3 点改进:

(1) 支持基于事务的调度

TC-AHB 策略设计基于单数据项, 而在实际应用中, 数据访问的原子工作粒度为事务, 且事务往往是多数数据项的。本文提出一种改进的 TC-AHB 策略(Improved TC-AHB), 改进策略将 TC-AHB 推广到基于事务的广播调度中。

(2) 细化广播粒度, 解决数据实时需求

周期广播只调度“热点数据”, 而在基于事务的调度中, 事务请求的数据项并不完全都是“热点的”, 一些事务的请求不能够在周期广播中得到满足。在周期广播带宽比例较大的情况下, 某些事务的截止期可能已经被超出, 细化广播粒度则可以解决过长的周期广播与数据实时要求之间的矛盾。

(3) 动态分配时间槽

借鉴文献[2]的思想, 动态分配数据广播中的时间槽。

3.1 基于事务的调度

令事务 T_i 请求的数据集为 DS_i , 其中数据项均为等长数据。定义该长度为单位长度 1, 数据项 k 具有截止期 D_k , 事务 T_i 具有截止期 TD_i , 令数据项 k 的有效截止期 $EffectiveD_k = \min\{TD_i, D_k\}$ 。

在周期广播调度中, 计算数据项 k 的带宽需求 $BW_{Requiredk}$ 采用 $1/EffectiveD_k$; 在按需广播调度中, 本文采用基于事务的 EDF-T 调度策略^[8], 来调度请求队列中事务的数据请求。

3.2 分布式周期广播

过长的周期广播可能会导致请求队列中某些即将到期事务的失败。改进的广播策略采用一种分布周期广播的思想以解决这一矛盾, 即将截止期将至的数据插入到周期广播中。BW 为周期广播占整个广播的百分比; 服务器每广播 APL^[3] (Atom Periodic Length) 时间长度的周期广播数据之后, 再广播 ADL^[3] (Atom On Demand Length) 时间长度的按需广播数据。如此循环反复, 直到 BW 比例带宽被周期广播用完。剩余带宽广播按需广播数据。定义周期广播原子粒度 APL 和按需广播原子粒度 ADL 如下:

If $BW > 1 - BW$

ADL = 1; APL = $BW / (1 - BW)$

Else

APL = 1; ADL = $(1 - BW) / BW$

最终整个广播周期分配如图 1 所示。系统采用 $(1, m)$ 分布式索引来指示被调度数据在周期广播中的分布, 同时降低移动用户的调谐时间。

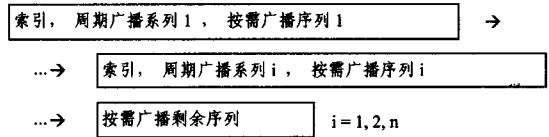


图 1 Improved TC-AHB 调度策略的广播周期分配图

分布式策略将周期广播数据分散至整个广播周期中, 中间嵌套按需广播数据。这一策略不仅提高了即将错过截止期的事务被及时调度的几率, 又由于“热点数据”的分散广播, 从而能够满足更多随机到达事务的数据请求。“分布周期广播”有效降低事务的失败率, 提高信道的利用效率。

3.3 Improved TC-AHB 策略的实现

3.3.1 参数说明

(1) BW: 周期广播占整个广播周期的比例。

(2) BW_Threshold: 周期广播带宽占整个广播周期的比例上限。

(3) BW_Required $_i$: 在数据项 i 被调度的前提下, 广播数据项 i 所用带宽的下限。 $BW_{Requiredi} = (S_i + 1) / (D_i + 1)$, D_i 为数据项 i 的相对截止期, S_i 为数据项 i 的长度。

(4) Requests_Received $_i$: 在上一广播周期中对数据项 i 的请求总次数。

(5) Priority $_i$: 数据项 i 在周期广播队列中的优先级, $Priority_i = Requests_Received_i / BW_{Requiredi}$ 。

(6) Allocated(i, t): 在本次调度中, 截至相对时刻 t 数据项 i 已占有的时间槽。为保证有限的周期广播带宽不被浪费, 需满足下述约束:

$$Allocated(i, t) / BW_{Requiredi} \leq t, t = 0, 1, \dots, T$$

当不满足该式时, 时间槽 t 将不会分配给数据项 i , 其中 $T = T_{total} \times BW$, T_{total} 为广播周期的长度。

(7) Pseudodeadline(i, t): 数据项 i 在时刻 t 的加权截止期, 它的值为 $(Allocated(i, t) + 1) / BW_{Requiredi}$ 的上取整,

时间槽 t 分配给在时刻 t 处具有最小加权截止期的数据项。加权截止期考虑了防止带宽浪费和数据的紧迫性。

(8) CutoffThreshold $_i$: CutoffThreshold $_i$ 为数据项 i 的相对截止期 D_i 与上次周期广播的长度的比值。该值说明: 只有数据项 i 的请求数超过数据项 i 在上一个广播周期中的最小调度次数时, 该数据项 i 才有被再次调度的意义。

(9) 待调度数据集 PendingScheduleDataSet: 用来存储上一广播周期的调度数据集以及当前请求队列中的数据项。

(10) 调度数据集 ScheduledDataSet: 它用来存储在本次周期广播中即将被调度的数据项。

3.3.2 算法思想

采用有效截止期 EffectiveDk 替代数据项的截止期 D , 将事务的实时限制纳入调度策略考虑范畴; Improved TC-AHB 策略时间槽分配算法统一分配周期广播和按需广播的时间槽。在周期广播中先调度截止期最近且占用广播时间最短的数据, 然后插入按需广播的紧急数据。如此循环往复, 直到周期广播数据播完, 按需广播的数据调度采用截止期最短优先的算法分配时间槽。

3.3.3 分布式周期广播时间槽分配算法

```

t=1; BWLen=Ttotal×BW;
While (t <= Ttotal)
  For i = 1 to APL && BWLen > 0 && t <= Ttotal
    计算 ScheduledDataSet 中所有数据项的加权截止期
    Pseudodeadline;
    选取 Pseudodeadline 最小且满足
    (Allocated(i)/BW_Requiredi) <= t 的数据项 i;
    将 k 放入时间槽 t 中, t++; BWLen--;
  End For
  For i = 1 to ADL && t <= Ttotal
    使用 EDF-T 策略, 选取最早截止期事务 T;
    选取 T 中最早截止期数据项 k, 标记 k 已被调度;
    将 k 放入时间槽 t 中, t++;
  End For
End While

```

4 仿真实验与性能评价

改进策略的算法空间复杂度为 $O(n)$, 时间复杂度最坏情况为 $O(n^2)$ 。由于影响系统的因素较多, 我们通过仿真实验来评价新的混合广播策略。

4.1 仿真系统设计

在排队模型中, 客户的到达时间一般满足指数分布规律。本文认为移动数据库系统即为一个排队服务系统, 采用指数分布来描述事务的到达时间间隔。Client 使用的事务集合由单独进程生成, 事务的截止期在给定范围内均匀分布。Zipf 分布能够较好地描述非均匀的偏斜访问模型, 如互联网中网站的被点击模式、数据库系统中数据的被访问模式等等, 所以本文选用 Zipf 分布描述数据项的请求模式。

仿真系统采用 C/S/M 结构, 客户进程 Client 在指数分布时间间隔发生器的触发下定时发送事务请求, 以模拟多移动客户的实时要求; 服务器进程 Server 接收事务的数据请求, 调度及广播数据; 监视器进程 Monitor 同时接收来自 Client 的数据请求和来自 Server 的广播数据, 进行实时对比, 动态计算事务失败率、上行信道负荷等性能指标。

进程间通信的数据单位为定长页面(Page), 广播序列即为页面的集合。页面头有一全局时钟信息, 通过页面的广播, 系统的所有进程在此时钟下同步。

4.2 实验参数

在表 1 给出的系统参数中, 数据项的偏斜因子 θ 、事务到达时间间隔的平均值 X 以及事务的截止期范围 (MinD, MaxD) 为可调参数, 本文正是基于这些参数来对各调度策略的性能进行实验评价。

表 1 仿真系统参数表

参数名称	分布	参数值	可调参数
广播带宽	—	4 page/ms	N
广播周期	—	50 ms	N
每页面的索引数	—	20	N
每页面数据项数	—	2	N
每页面的事务数	—	2	N
每事务请求的数据项数	均匀分布	1~10	N
总事务数	—	5000	N
总数据项数	—	1000	N
数据项请求模式分布	Zipf 分布	$\theta: 0.95$	Y
事务到达时间间隔	指数分布	X ms	Y
事务截止期	均匀分布	(MinD, MaxD)ms	Y

表 2 列出 Improved TC-AHB 策略所使用的算法参数。这些参数应用于请求频率的采样以及周期广播带宽的自适应调节等模块中。

表 2 Improved TC-AHB 的算法参数

参数名称	符号	参数值
周期广播带宽比例上限	BW_Threshold	0.7
被采样项的期望请求数	Expected_Sample_Size	5
“热点数据”请求频率的衰减因子	Cooling_Factor	0.9

4.3 性能评价指标

本文采用如下两个指标作为调度策略性能的评价指标。

- 事务失败率 TFR(Transaction Failure Ratio)

如果事务所请求的数据集在其截止期内均得到满足, 则事务成功。在实时数据广播中, 相对于访问时间和调谐时间, 降低事务的失败率更为重要。

- 上行信道负荷 ULL(Up Link Load)

由于周期广播, 用户对热点数据的请求可能在请求发出之前已经得到满足, 从而上行信道的通信负荷得到降低, 系统可以将更多的带宽应用于下行信道。所以上行信道的负荷是一重要的系统性能参数。本文采用 EDF-T 调度策略的上行信道负荷为 100% 参考标准。

4.4 实验结果及分析

实验选取 Zipf 分布的偏斜因子为 0.95, 此时数据偏斜约为 80~20, 较为接近现实应用中的数据请求模式。系统在完成 Improved TC-AHB 策略的同时, 实现了 EDF-T 策略以及 TC-AHB 策略, 以供性能比较。

4.4.1 平均到达时间间隔对事务失败率 TFR 的影响

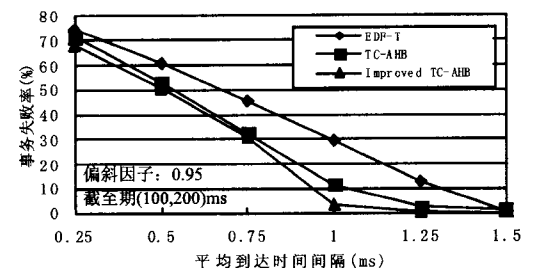


图 2 平均到达时间间隔对事务失败率的影响

实验结果图2说明,就整体趋势而言,Improved TC-AHB比EDF-T和TC-AHB有约20%,3%的性能优势,原因在于由“分布周期广播”的实现。在1ms处,前两者的性能差异增大到10%,Improved TC-AHB策略最早到达最优。

4.4.2 截止期对事务失败率 TFR 的影响

由实验结果图3所知,随着截止期增大,各算法性能差别越来越明显。EDF-T策略没有利用数据的请求模式,随着截止期的变化,性能并没有显著提升,在截止期(120ms,220ms)之后,Improved TC-AHB策略比TC-AHB策略有8%左右的优势。原因在于“分布周期广播”提高了即将错过截止期的实时事务的响应概率。

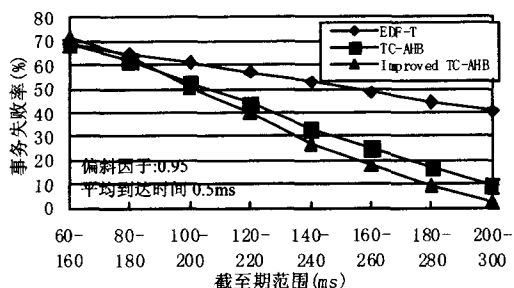


图3 截止期范围对事务失败率的影响

4.4.3 上行信道负荷比较

图4显示,在 θ 从0变化到0.6时,Improved TC-AHB的上行信道负荷ULL比TC-AHB逐渐有8%的降低。而 θ 处于0.6到1.4之间,这一差距拉大到10%以上。主要原因是数据的偏斜程度随 θ 的增大而呈非线性增长,从而使周期广播的调度数据集中的数据项逐渐增多;调度数据集的大小直接决定了周期广播的长度,周期广播越长,未发出的数据请求得到满足的几率越大;同时,数据集中数据请求频率的大小又直接决定了多少用户的数据请求能够同时得到满足。

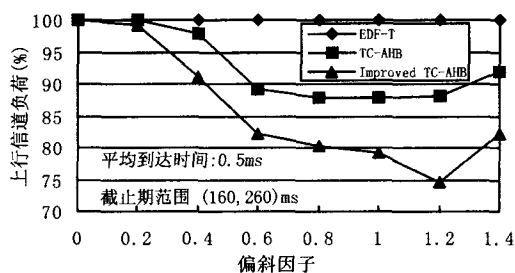


图4 上行信道负荷比较

数据的偏斜约为80~20,improved TC-AHB策略的ULL理论最优值应比按需调度低20%。实验结果与理论最优值仍然10%的差距,这是Improved TC-AHB策略可以进一步改进的重要提示。

4.4.4 偏斜因子对事务失败率 TFR 的影响

在实验中可同时得到Zipf偏斜因子对事务失败率影响的结果,如图5所示。

三种调度策略的性能均随着数据的偏斜因子增大而提高。在 $\theta=1.4$ 时,三者几乎达到了最优性能。从整体上看,Improved TC-AHB比TC-AHB性能约有8%的提升,比EDF-T提升性能约20%以上。Improved TC-AHB策略性能的增强一方面来自于TC-AHB策略的基础,另一方面由于周期广播数据的分散,使得更多未提交的数据请求有机会在客

户端即得到满足。

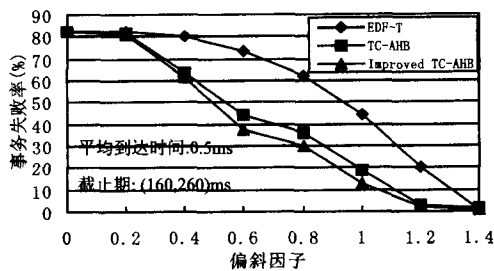


图5 偏斜因子对事务失败率的影响

4.4.5 实验结果分析

仿真系统的实验结果表明:

(1)Improved TC-AHB策略适用于实时环境下的数据广播,它既有效利用高频数据的“热点”特征,又充分考虑了事务及数据的实时限制,在按需广播中及时调度紧急请求,即较低的事务失败率。

(2)Improved TC-AHB策略扩大周期广播的覆盖长度,使得更多的“热点数据”请求得到满足,同时通过将细小按需广播片断穿插于周期广播之内,使得更多即将到期的数据请求及时得到调度,即上行信道负荷更小。

结束语 Improved TC-AHB策略虽然能够有效地适应实时数据广播应用,但依然存在不足,可以做改进。

(1)采样技术的进一步完善:确定“热点数据”的采样时间较为机械,数据的采样时间直接影响数据的采样精度以及采样的系统消耗。如何动态地调整采样时间,是今后一个重要研究任务。

(2)按需广播部分的调度策略的选用:Improved TC-AHB策略在按需广播中使用的是EDF-T调度,该调度适用于Improved TC-AHB是否最优,是一个值得探讨的问题。

(3)周期广播与数据实时性需求之间矛盾的化解:本文引入“分布周期广播”解决周期广播越长可能有越多的实时数据请求超出截止期的矛盾。周期广播的粒度确定是否应该引入其它参数,是今后的一个重要研究工作。

参考文献

- [1] Jesus F C, Ramamritham K. Adaptive dissemination of data in time-critical asymmetric communication environments [C] // Proc. of the 11th Euromicro Conf. 1999;195-203
- [2] Lee W C, Hu Q L, Lee D L. A study of channel allocation methods for data dissemination in mobile computing environments [J]. ACM/Baltzer Journal of Mobile Networks and Applications (MONET), 1999, 4(2): 117-129
- [3] Jesus F C, Ramamritham K. Adaptive dissemination of data in time-critical asymmetric communication environments [J]. Mobile Networks and Applications, 2004, 9(5): 491-505
- [4] Lam K, Edward C, Yuen C H. Data broadcast for time - constrained read-only transactions in mobile computing systems [C] // Proc. of Advance Issues of E-Commerce and Web-Based Information Systems. 1999;11-19
- [5] Huang J L, Chen M S, Peng W C. Broadcasting dependent data for ordered queries without replication in a multi-channel mobile environment [C] // Proc. of the 19th Int. Conf. on Data Enginee-

- [6] Lee G L, Lo S C. Broadcast data allocation for efficient access of multiple data items in mobile environments [J]. *Mobile Networks and Applications*, 2003, 8(4): 365-375
- [7] Chang Y I, Hsieh W H. An efficient scheduling method for que-

ry-set-based broadcasting in mobile environments[C]//Proc. of the 24th Int. Conf. on Distributed Computing Systems Workshops, 2004;478-489

- [8] 刘云生, 杨进才, 廖国琼. 移动环境中实时事务数据的广播调度 [J]. *小型微型计算机系统*, 2004, 25(4): 531-534

(上接第 153 页)

(iv)若存在 $g_1:t_1', \neg g_1 \wedge g_2:t_1'' = head(t)$, 则必然可以运用定义 6 的规则(5), 提出循环活动, 然后转到第(i)步; 否则转到第(v)步。

(v)此时必然满足定义 6 的规则(1), 即可完全消去并行运算符号。

由此可以看出, 定义 7 中所有角色的全局执行是在有限步内得出结果的。

4.2 一致性问题

一个按照各自的执行过程进行 Web 服务组合, 然后再将各参与方并行执行, 最终效果应该符合编排中所描述的那样, 局部模型的并行组合之后的语义应该等价于全局模型的语义, 即 $T'_c = T_c$, 这就是一致性问题。

一致性表现在以下两个方面: (i)各个参与方并行执行之后, 活动的执行顺序应该符合编排里所描述的全局过程; (ii)在某些特定的时刻, 对于编排里所规定的某些信息(变量), 各个参与方必须达成一致的共有的认识(common view), 当参与方独立执行时, 这个共有的认识也不能被破坏。由于各个参与方是相互独立而且异步地执行的, 显然, 不一致的情况会经常发生, 例如: (i)有一个全局的编排活动 $A = a_1; a_2$, 映射到 R_1 和 R_2 上的局部执行活动分别为 $M_1(A) = a_1$ 和 $M_2(A) = a_2$, 显然两个局部活动的并行结果 $a_1 || a_2$ 与全局的编排 $a_1; a_2$ 不一致; (ii)编排中规定了两个角色在交互中必须对某个变量达成一致, 但是当独立执行时候, 其中一个角色改变了该变量的值, 而另一个角色还仍然保留着原来的值。一致性问题的处理非常重要, 是因为 WS-CDL 规范仅仅用声明性的方法要求以交互信息校准(Interactive Information Alignment)的方式或编排协调(Coordination)的方式保证同步, 但是并没有明确定义这些方式的执行机制。文献[8]中通过为全局模型和局部模型建立轨迹语义进一步探讨了一致性问题, 提出了一些解决不一致问题的方案, 如提出“支配选择”的概念, 设定某个角色为“支配角色”, 由它来支配其余角色应该进入哪个选择分支。文献[9]也建立了模型, 通过校准机制着重讨论了一致性问题。

本文中局部模型的建立以及全局模型到局部模型的映射过程都忽略了一个基本的事实, 即卫士条件和循环条件中可能包含不同角色的变量。例如有一个编排 C 的活动 $A = g_1: P[]g_2: Q$, 其中 $g_1 = (v_1 < 8 \wedge v_2 < 7)$, $g_2 = (v_1 > 8 \wedge v_2 > 7)$ 。此时映射到局部模型之后, 对于 R_1 , 因为只能访问到其本地变量 v_1 而不能访问远程变量 v_2 , 所以必然不能采取正确的选择, 对于 R_2 也是如此, 这样不一致的情况就产生了。为解决这类问题, 需要引入同步机制(synchronization), 在选择之前通过加入一些额外的交互来达到同步。上述例子中对 A 加入同步交互之后变成 $A' = (c1[1, 2] || c2[2, 1]); g_1: P[]g_2: Q$, 其中 $c1[1, 2]$ 代表从 R_1 到 R_2 且通道名为 $c1$ 的交互, 将 R_1 的本地变量 v_1 的值传递到 R_2 中作为一个副本 \tilde{v}_1 , $c2[2, 1]$ 代

表从 R_2 到 R_1 且通道名为 $c2$ 的交互, 将 R_2 的本地变量 v_2 的值传递到 R_1 中作为一个副本 \tilde{v}_2 。这样在进行选择的时候, 就可以采取正确的方式达到一致性。

结束语 正如文献[1]中提到的那样, 在 Web 服务的发展过程中, WS-CDL 规范似乎出来得太早。WS-CDL 规范没有形式化基础, 规范的开发也没有基于任何一种可执行的程序语言。本文为 WS-CDL 建立了一个形式化模型, 扩展了轨迹语义, 加入条件, 更好地模拟了编排的执行, 通过对带条件的轨迹语义定义全局执行来讨论全局与局部的一致性问题。进一步的工作包括建立 WS-CDL 规范到全局模型的自动转换过程以及通过校准和同步来完善一致性问题的讨论。

参考文献

- [1] Barros A, Dumas M, Oaks P. A Critical Overview of the Web Services Choreography Description Language [OL]. <http://www.bptrends.com>, 2005
- [2] Decker G, Zaha J M, Dumas M. Execution Semantics for Service Choreographies. Preprint # 4329, Faculty of IT, Queensland University of Technology, May 2006
- [3] Foster H, Uchitel S, Magee J, et al. Model-based analysis of obligations in web service choreography[C]//Proceedings of International Conference on Internet and Web Applications and Services, 2006. 2006, 149:19-25
- [4] Busi N, Gorrieri R, Guidi C, et al. Towards a formal framework for Choreography[C]// Proceedings of 3rd International Workshop on Distributed and Mobile Collaboration, 2005. DMC'05
- [5] Yang H L, Zhao X P, Qiu Z Y. A Formal Model for Web Service Choreography Description Language (WS-CDL) [C]// Proceedings of IEEE International Conference on Web Services (ICWS'06). September 2006; 893-894
- [6] Zhao X P, Yang H L, Qiu Z Y. Towards the Formal Model and Verification of Web Service Choreography Description Language [C]//Proceedings of Web Services and Formal Methods (WS-FM'06). September 2006; 273-287
- [7] Qiu Z Y, Cai C, Zhao X P, et al. Exploring into the Essence of Choreography. Preprint 2006-63 of Institute of Mathematics, Peking University [OL]. <http://www.math.pku.edu.cn:8000/en/preindex.php>
- [8] Qiu Z Y, Zhao X P, Cai C, et al. Towards the theoretical foundation of choreography[C]//Proceedings of the 16th international conference on World Wide Web (WWW'07). May 2007; 973-982
- [9] Kazhamiak R, Pistore M. Choreography Conformance Analysis: Asynchronous Communications and Information Alignment [C]//Proceedings of Web Services and Formal Methods (WS-FM'06). September 2006; 227-241