

BPEL 组合服务流程动态性能的定量预测方法^①

姜 峰^{②*} 范玉顺^{*} 曾 森^{***}

(^{*}清华大学自动化系 北京 100084)

(^{**}中国农业银行总行信息技术管理部 北京 100005)

(^{***}桂林空军学院自动化系 桂林 541003)

摘 要 针对服务质量研究在流程模型相关性和流程性能动态性方面存在的不足,将吞吐量 and 响应时间引入到组合服务流程性能评估的流程级指标中,借助排队网络系统的相关结论,研究了组合服务流程动态性能的定量预测方法,给出了针对组合服务流程执行语言(BPEL)的动态性能临界值计算方法,并通过一个示例介绍了该方法在 BPEL 组合服务流程的动态性能定量预测和组合方案优化中的应用。

关键词 业务流程执行语言(BPEL), 组合服务, 动态性能, 性能预测, 排队网络

0 引 言

组合服务流程建模和性能优化是基于面向服务架构(service oriented architecture, SOA)构建应用流程的两个重要研究方向^[1]。业务流程执行语言(business process execution language, BPEL)^[2]可以编排任意复杂的业务流程,并支持流程的长时间运行和状态维持,因而被广为接受,成为事实上的服务流程组合规范之一。性能优化主要关注组合流程的非功能属性(称为服务质量(quality of service, QoS)),致力于寻找最佳的或最低能满足用户需求的组合方案,是近年来该领域的研究热点^[3]。按照性能优化的层次,可以将现有研究分为三类:针对服务个体选择的局部优化、针对组合方案的全局优化和针对组合服务流程运行时的动态优化。例如,文献[4]提出了一套服务质量指标,并通过归一化和加权计算给出了服务个体的性能排序方法;文献[5]针对服务时间、服务代价、可靠性、可维护性、满意度等 QoS 指标给出了组合服务流程的质量评估模型和计算方法, Mohabey^[6]、Claro^[7]、郭于明^[8]等人在此基础上分别采用整数规划、多目标规划、遗传算法等方法,研究了组合服务流程的路径选择和优化问题;文献[9]则重点讨论了组合服务流程执行过程中的动态调度问题。

这些研究覆盖了组合服务流程性能评估的大部

分领域,但存在如下两点不足:一是大多数质量评估模型没有与组合服务流程的建模规范例如 BPEL 规范结合起来,模型的实用性较低;二是组合服务流程的优化问题描述为设计时的组合模型静态优化和执行时的服务终端的动态调度两阶段问题,这种方式在动态性能优化时对系统的实时性要求很高,并且常常导致更复杂的动态性问题例如模型变更、实例迁移等问题的产生。针对上述不足,本文基于静态模型研究了 BPEL 组合服务流程的动态性能预测方法,即以系统吞吐量和响应时间两个指标为例,借助排队网络(queueing network, QN)^[10]系统的边界分析理论,建立基于 BPEL 组合服务流程静态模型的动态性能定量预测模型,从而提供了一种设计时针对动态性能的流程模型优化方法,并通过一个实例说明了该方法的具体应用。

1 BPEL 组合服务流程元模型

BPEL 是 IBM 和 Microsoft 共同推出的业务流程执行语言,其作用是将一组现有的服务组合起来,形成抽象的业务流程(描述流程参与方对外可见的消息交换)或可执行业务流程(描述流程参与方的全部实际行为)。其元模型如图 1 所示。

BPEL 组合业务流程的顶层描述元素是流程(Process),它由一系列活动(Action)组成。活动分为两类:简单活动(Simple Action)和复合活动(Compos-

^① 863 计划(2007AA04Z150)和国家自然科学基金(60674080, 60704027)资助项目。

^② 男,1981 年生,博士生;研究方向:面向服务的业务流程管理和 Web 服务资源管理等;联系人, E-mail: jiangfeng00@gmail.com (收稿日期:2009-02-16)

ite Action)。其中,与服务组合相关的活动是“调用(involve)”,它通过“操作(operation)”元素调用其它服务的接口,这些接口通常被定义在 Web 服务描述语

言(web service description language, WSDL)^[11]所定义的服务描述文档中。

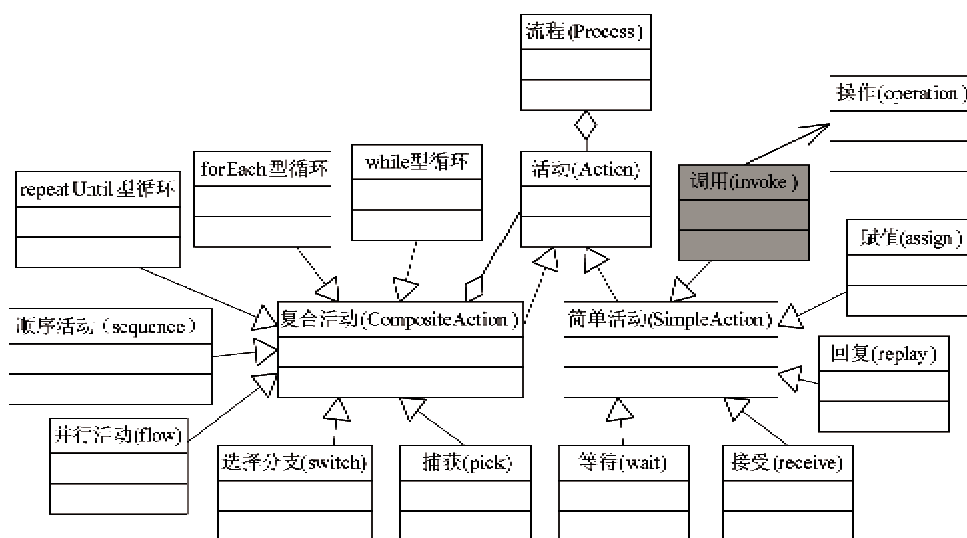


图1 BPEL 组合服务流程元模型

2 组合服务流程的性能动态性和性能指标

2.1 性能动态性问题描述及建模

在组合服务流程两阶段性能优化中,组合服务流程的性能通常基于单个服务的质量模型给出,可以用多元组 $(ST, SC, Re, \dots)$ 表示,其中 $ST = f(ST_1, ST_2, \dots)$, $SC = g(SC_1, SC_2, \dots)$, $Re = r(Re_1, Re_2, \dots)$ 等分别表示组合流程的各种性能指标,它们是成员服务对应指标的函数。在这种性能计算模型中,成员服务是没有状态的,组合流程模型和成员服务一旦确定,流程的性能就随之确定,而与流程实例与成员服务间的交互无关。

但实际上例如网格服务是有状态的,组合服务流程的性能不仅与模型有关,还与成员服务的状态有关。在图2所示的流程实例与成员服务的交互

模型中,流程模型的每个实例调用其成员服务都会形成对成员服务的一次占用,后续到来的实例只有在当前实例调用结束后才能被执行。因此,随着对成员服务接口调用的实例数的增加,实例与成员服务间会产生排队现象,导致同一组合流程模型的不同实例具有不同的性能。

若将所有组合服务流程的成员服务组成的集合视作一个服务资源系统,把组合流程的每个实例看作服务资源系统的一个客户,则所有流程实例和成员服务构成一个排队网络(QN)系统,实例的每个 invoke 操作的执行即相当于 QN 系统中一个客户的到达,而多个实例对同一成员服务的调用就会形成 QN 系统中的排队现象。据此,本文将借助于 QN 系统的有关分析方法研究这种由模型的多实例引起的性能变化问题。

由于面向服务架构是一种开放的架构,在组合服务流程网络中,同一个服务资源可以被不同的流程实例调用。流程实例可以来自同一个流程模型,例如 P_{1i} 和 P_{1k} ;也可以来自不同模型,例如 P_{mn} 和 P_{1kc} 。这种情况类似于 QN 系统中的单类型客户和多类型客户。为便于说明,本文只考虑前者。

2.2 动态性能指标选取

性能指标选取是性能预测和评估的首要工作。吞吐量和响应时间两个指标是 QN 系统的重要度量指标,它们反映了系统的执行能力。首先给出组合服务流程动态性能预测中吞吐量和响应时间的定

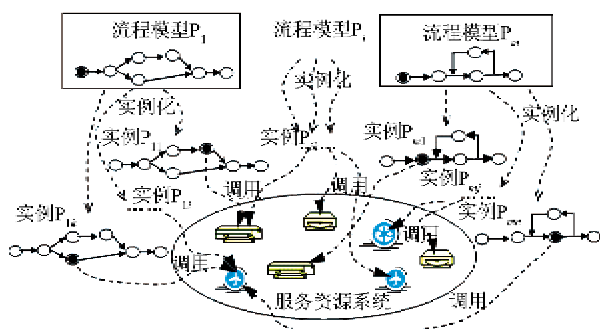


图2 组合服务流程的多个实例与成员服务交互的模型

义:

• 吞吐量 T : 组合服务流程在单位时间内可以被完成的实例数。

• 响应时间 R : 组合服务流程的每个 invoke 活动的响应时间。它包括调用成员服务的等待时间和成员服务的执行时间。后者由被调用成员服务执行的复杂程度决定,前者则依赖于网络中流程实例的并发状态。

按照上述定义,吞吐量和响应时间不仅反映了服务资源的利用率,其变化还会使同一流程模型的不同实例具有不同的性能,反映了一定时间内组合服务流程的可执行能力。因此这两个指标是服务的提供者 and 使用者都关注的指标。而由于现有组合流程性能评估的研究都基于服务无状态和服务能力无限假设,因此都没有考虑到组合服务流程实例与成员服务间交互引起的动态性能变化问题,吞吐量和响应时间也就没有被视作流程级的性能指标^[3-5]加以研究。如前所述,不仅网格服务是有状态的,由于成员服务的实现通常需要一定的资源做支撑,如占用 CPU、占用网络带宽和其它资源等,而资源能力是有限的,因此成员服务的能力也是有限的。据此,本文讨论组合服务流程的吞吐量和响应时间两个指标具有现实意义。

3 基于 QN 系统的动态性能定量预测

边界分析是 QN 系统的重要性能分析方法,该方法不仅可以定量描述系统的最优和最差性能的动态变化趋势,还能揭示系统的性能瓶颈,从而为系统优化提供指导。本文采取 QN 系统的边界分析技术,研究组合服务流程的吞吐量和响应时间的预测方法。

3.1 QN 系统的边界性能分析方法

首先直接引入 QN 系统的有关结论^[10]。对于一个 QN 系统,如下结论成立:

(1) 若 QN 系统是一个开放系统,则系统吞吐量和响应时间满足

$$T(\lambda) \leq \frac{1}{D_{\max}} \quad (1)$$

$$\frac{D}{1 - \lambda D} \leq R(\lambda) \leq \frac{D}{1 - \lambda D_{\max}} \quad (2)$$

(2) 若 QN 系统是一个封闭系统,则系统吞吐量和响应时间满足

$$\frac{K}{D + Z + \frac{(K-1)D_{\max}}{1 + Z/(KD)}} \leq T(K)$$

$$\leq \min \left(\frac{1}{D_{\max}}, \frac{K}{D + Z + \frac{(K-1)\bar{D}}{1 + Z/D}} \right) \quad (3)$$

$$\max \left(KD_{\max} - Z, D + \frac{(K-1)\bar{D}}{1 + Z/D} \right) \leq R(K)$$

$$\leq D + \frac{(K-1)D_{\max}}{1 + Z/(KD)} \quad (4)$$

式(1) - (4)分别描述了一个 QN 系统在开放和封闭两种情形下的吞吐量和响应时间的上界和下界。其中: $D, \bar{D}$ 和 $D_{\max}$ 分别表示系统的服务需求总量、每个资源的平均服务需求量和资源的最大服务需求量; λ 是开放 QN 系统的客户到达速率; K 和 Z 分别表示封闭 QN 系统的客户总数和客户到达的间隔时间。这里,所谓开放系统是指客户总数不确定,每个客户以 λ 速率进入 QN 系统;所谓封闭式系统是指客户总数为 K , 每两个客户到达的间隔时间为 Z 。

由(1) - (4)式可知,确定系统的吞吐量和响应时间等边界性能,可以归结为确定系统的服务需求量 $D, \bar{D}$ 和 $D_{\max}$ 。

资源 R_i 的服务需求量 D_i 由式

$$D_i = V_i S_i \quad (5)$$

给出,其中, V_i 和 S_i 分别表示单个客户对资源 R_i 的请求服务次数和每次服务时间。因此有

$$D = \sum_i^M D_i, \bar{D} = D/M, D_{\max} = \max(D_i) \quad (6)$$

此外,根据 QN 系统的利用率定律^[10],资源 R_i 的利用率可表示为

$$U_i = T_i S_i = T D_i \quad (7)$$

其中, T_i, S_i, D_i 和 T 分别表示资源 R_i 的吞吐量、有效服务时间、服务需求量和整个资源系统的吞吐量。

式(7)表明,系统的吞吐量与资源的服务需求量成反比。因此服务需求量最大的资源即是系统的性能瓶颈。

3.2 BPEL 组合服务流程动态性能的边界计算

对于 BPEL 组合服务流程及其成员服务资源构成的系统,不妨记 BPEL 组合服务流程为 $P = \{A_1, A_2, \dots, A_N\}$, 其中 $A_i, i = 1, 2, \dots, N$ 表示组合流程 P 的活动节点;记 $R = \{R_1, R_2, \dots, R_M\}$ 为流程 P 的所有成员服务所部署的物理资源集。则 BPEL 组合服务流程的每个实例活动 A_i 对资源 R_j 的调用形成排队系统的一次客户到达和服务过程,因此整个系统构成一个排队网络 QN 系统。计算系统的边界吞吐量和响应时间,就是要确定系统的类型和成员服务资源的服务需求量。

系统的类型可以由标记的 BPEL 组合服务流程描述文件直接确定。系统的类型一旦确定,相应的 λ 或 K, Z 参数即可随之确定。

由于系统的吞吐量 T 被定义为组合服务流程在单位时间内可以被完成的实例数,因此成员服务资源的服务需求量可以表示为

$$D_i = V_i S_i = \sum_{A_j \in A} V_j S_{ji} \quad (8)$$

其中: D_i 表示资源 R_i 的服务需求量; V_j 表示活动 A_j 的平均执行次数,它等于单位时间内活动 A_j 被执行的次数与吞吐量的比; S_{ji} 表示资源 R_i 对活动 A_j 的有效服务时间。

下面给出 BPEL 组合服务流程中 S_{ji} 和 V_j 的计算方法。

3.2.1 资源服务时间

BPEL 组合服务流程调用其成员服务,成员服务的执行通常需要一定的资源做支撑,如占用一定的 CPU 或网络带宽等。资源服务时间 S_{ji} 就是对这些资源开销的度量。

资源服务时间的取值来源于两个途径:一是服务提供者度量,并在服务的 WSDL 文件中进行明确的标示;二是依据历史数据进行统计估算,例如求平均值。

在计算服务需求量时, S_{ji} 的取值规则是:

(1) 若活动 A_j 不是一个 invoke 活动,则 $S_{ji} = 0$;

(2) 若活动 A_j 是一个 invoke 活动,则 S_{ji} 等于上述两种途径确定的值。若资源 R_i 的开销未被提供或无法估算,则取 $S_{ji} = 0$ 。

3.2.2 平均执行次数

平均执行次数 V_j 等于单位时间内活动 A_j 被执行的次数与系统吞吐量的比。它的取值可以通过组合服务流程 BPEL 文档的结构化分析获得。

对于 BPEL 中的简单活动,平均执行次数可以按照定义直接求解;组合活动的求解规则如下:

(1) 顺序活动(Sequence)。Sequence 活动 A_j 被定义为一组子活动 A_{jk} 的顺序执行,表示为 $A_j = (A_{j1} > \dots > A_{jk})$ 。活动的平均执行次数的计算规则为 $V_{jk} = V_j$ 。

(2) 并行活动(Flow)。Flow 活动 A_j 被定义为一组子活动 A_{jk} 的并发执行,表示为 $A_j = (A_{j1} \mid \dots \mid A_{jk})$ 。活动的平均执行次数的计算规则同 Sequence 活动。

(3) 选择分支(Switch)。Switch 活动 A_j 被定义为一组子活动 A_{jk} 在满足不同条件下的选择执行。

假设各子活动 A_{jk} 的执行条件的满足概率为 p_{jk} , 则 Switch 活动可表示为 $A_j = (A_{j1} \mid p_{j1}, \dots, A_{jk} \mid p_{jk})$, 且 $\sum_k p_{jk} = 1$ 。活动的平均执行次数的计算规则为 $V_{jk} = p_{jk} V_j$ 。

(4) 捕获活动(Pick)。Pick 活动 A_j 的定义及平均执行次数的计算方法与 Switch 活动相似。所不同的是 Pick 活动的判断语法为 onMessage 或者 onAlarm, 而 Switch 活动的判断语法为 case 和 otherwise。

(5) RepeatUntil 型循环活动。RepeatUntil 型循环活动 A_j 被定义为子活动 A'_j 被循环执行,直到终止条件 C 成立为止。假设终止条件 C 成立的概率为 p , 则 RepeatUntil 型循环活动可表示为 $A_j = (A'_j \rightarrow A'_j \mid p)$ 。活动的平均执行次数的计算规则为 $V_j = \frac{1}{p} V_{j'}$ 。

(6) While 型循环活动。While 型循环活动 A_j 表示当条件 C 满足时执行其子活动 A'_j , 通常也是表示一个循环活动。假设条件 C 成立的概率为 p , 则 While 型循环活动可表示为 $A_j = (A'_j \mid p)$ 。活动的平均执行次数的计算规则为 $V_j = \frac{p}{1-p} V_{j'}$ 。

(7) ForEach 型循环活动。ForEach 型循环活动 A_j 表示次数确定的循环活动。假设子活动执行的起止计数分别为 a 和 b , 则 ForEach 型循环活动可表示为 $A_j = (A'_j \mid b - a)$ 。活动的平均执行次数的计算规则为 $V_j = (b - a + 1) V_{j'}$ 。

上述规则中,子活动 A_{jk} 或 A'_j 既可以是简单活动,也可以是组合活动。当 A_{jk} 或 A'_j 是组合活动时,可以采用递归方法求解,直至求得每一个简单活动的平均执行次数。子活动 A'_j 的执行或终止概率可以由组合服务流程的拥有者直接给出,也可以通过以往的执行数据估算而出。

4 应用示例

按照上述方法可以实现基于组合服务流程静态模型的动态性能定量预测。下面通过一个示例介绍该方法的应用。

网格计算是科学计算的重要技术,它通过组合网格中的不同计算节点完成大规模复杂计算任务。一个网格服务组合流程的 BPEL 描述文件如下:

```

< bpws:process >
< bpws:sequence >
  < bpws:invoke operation = "Identification"/ >
  < bpws:flow >
    < bpws:invoke operation = "DocumentTransfer"/ >
    < bpws:invoke operation = "DataTransfer"/ >
  < /bpws:flow >
< bpws:switch >
  < bpws: case condition prob = "0.4" >
    < bpws:invoke operation = "NumericalComputation"/ >
  < /bpws: case >
  < bpws: otherwise >
    < bpws:invoke operation = "LogicalComputation"/ >
  < /bpws: otherwise >
< /bpws:switch >
  < bpws:invoke operation = "DataSave"/ >
  < bpws:invoke operation = "Analyze"/ >
< /bpws:sequence >
< /bpws:process >
    
```

该组合服务流程由数据存储服务(DS)、计算服务(CS)和分析服务(AS)3个服务组成,其基本流程是:首先调用计算服务的 Identification 操作进行身份验证;然后同时调用计算服务的 DocumentTransfer 操作和数据存储服务的 DataTransfer 操作进行文件传递和数据传递;然后按照计算类型调用计算服务的 NumericalComputation 操作或 LogicalComputation 操作进行数值运算或逻辑运算,两种操作的执行概率分别为 0.4 和 0.6;之后调用数据存储服务的 DataSave 操作进行数据存储;最后调用分析服务的 Analyze 操作进行结果分析。由资源服务时间的计算规则可知,只有 invoke 类型的活动才是 QN 系统中的有效活动,因此上述流程省略了 BPEL 中其他与计算无关的活动和条件等描述。

组合服务流程所调用的成员服务(及资源消耗)的参数如表 1 所示,它们被定义在成员服务的 WSDL 描述文档中。

按照 3.2 节所述计算本例组合服务流程及成员服务资源系统的服务需求量。例如对于 BPEL 流程中的 Switch 活动,其平均执行次数的计算过程为: $V_{\text{NumericalComputation}} = p_{\text{case}} V_{\text{Switch}} = 0.4 V_{\text{Switch}} = 0.4 V_{\text{process}} = 0.4$, $V_{\text{LogicalComputation}} = (1 - p_{\text{case}}) V_{\text{Switch}} = 0.6 V_{\text{Switch}} = 0.6 V_{\text{process}} = 0.6$, 与其对应的资源开销分别是 CS: CPU 160, CS: Disk 120 和 CS: CPU 40, CS: Disk 50。依次类推,计算出所有原子活动(操作)的平均执行

表 1 BPEL 及成员服务的接口参数

WSDL 服务名称	WSDL 操作 A_j	资源 R_i	资源开销 S_{ji}
计算服务 (CS)	Identification	CS: CPU	6
	DocumentTransfer	CS: CPU	4
		CS: Disk	90
		Network	21
	NumericalComputation	CS: CPU	120
数据服务 (DS)	LogicalComputation	CS: Disk	160
		DS: CPU	2
		DS: Disk	90
	DataTransfer	Network	53
		DS: CPU	2
分析服务 (AS)	DataSave	DS: Disk	30
		AS: CPU	90
	Analyze	AS: Disk	40

次数,按式(8)即可计算出所有成员服务资源的服务需求量,如表 2 所示。

表 2 组合服务流程的服务需求量

操作 A_j	平均执行 次数	资源 R_i	服务需求 量 D_i
Identification	1	DS: CPU	4
DocumentTransfer	0.4	DS: Disk	120
NumericalComputation	0.6	CS: CPU	82
LogicalComputation	1	CS: Disk	154
DataTransfer	1	AS: CPU	90
DataSave	1	AS: Disk	40
Analyze	1	Network	74

服务需求量 $D = \sum D_i = 564$, $\bar{D} = D/7 = 80.6$, $D_{\max} = 154$ 。按照式(1) - (4)即可计算 BPEL 组合服务流程及其成员服务在不同负载类型下的吞吐量和响应时间的临界值,从而实现基于组合服务流程静态模型的动态性能定量预测。例如在开放系统中:流程实例发生速率 $\lambda = 9$ 个/小时(或 0.0025 个/秒),则每个实例调用其成员服务的响应时间 R 最长为 917 秒,最短为 706 秒,系统每小时最多能完成的流程实例数 T 为 23 个(或每秒 0.0065 个);而当流程实例发生速率 λ 增加到 21 个/小时(或 0.0058 个/秒)时,每个实例调用其成员服务的最长响应时间增加到 5548 秒,最短响应时间则增加到 1064 秒。又如在封闭系统中,当实例间隔时间 $Z = 50$ 秒,流程实例总数 $K = 3$ 个时,每个实例调用其成员服务的响应时间 R 最长为 863 秒,最短为 712 秒,系统每小时最多能完成的流程实例数 T 最多为 14 个(或每秒 0.0039 个),最少为 11 个(或每秒 0.0032

个);而当流程实例数 K 增加到 8 个以上时,系统每小时最多能完成的流程实例数 T 最多都为 23 个。

图 3 和图 4 分别是系统的吞吐量和响应时间的临界值在封闭状态下随流程实例总数 K 、实例间隔时间 Z 变化以及在开放状态下随流程实例发生率 λ 变化的曲线图。可以清晰地看到,同一组合服务流程模型在不同的实例执行模式下具有不同的动态性能,即使在同一模式下,随着系统中运行实例数量的变化,也会产生不同的性能。

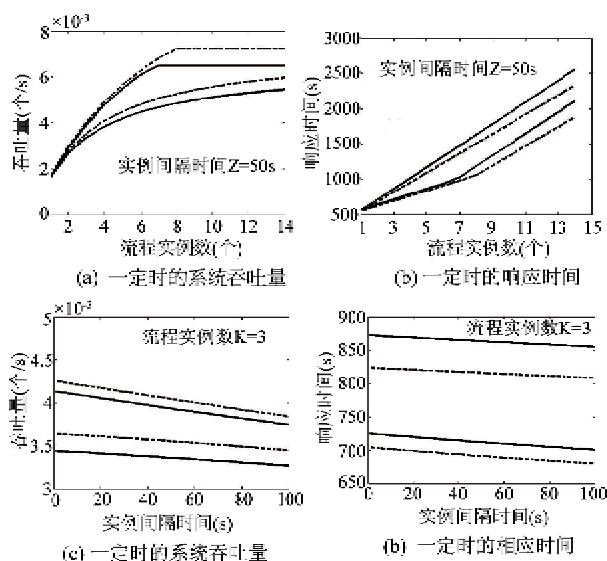


图 3 封闭负载下的系统吞吐量和响应时间

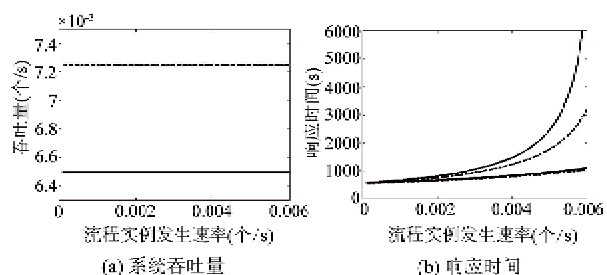


图 4 开放负载下的系统吞吐量和响应时间

当流程实例数超过系统的最大吞吐量时,例如本例中流程实例发生率超过 23 个/小时时,就会产生实例滞留,从而影响组合服务流程的执行。在 3.1 节中曾指出,服务需求量最大的资源及其对应的成员服务是整个组合服务流程的性能瓶颈。这就为组合流程的优化提供了指导:由于服务需求量与访问次数和资源的的服务时间成正比,因此,用户在优化组合流程时,可以选择服务时间短的成员服务,或者修改组合服务流程模型以减少对服务时间长的成

员服务的调用次数。在本例中,服务需求量最大的是计算服务的硬盘资源 CS: Disk, 据此,可以选择其它的计算服务以降低最大服务需求量,从而提高组合服务流程的吞吐量。假设新选择的计算服务 CS' 的 NumericalComputation 操作对硬盘 CS: Disk 的占用量降为 120, 则系统对 CS: Disk 的服务需求量降低为 138。系统的吞吐率和响应时间如图 3 和图 4 中虚线所示,显然响应时间越短越好(对应图上曲线位置越低越好),系统吞吐量越大越好(对应图上曲线位置越高越好)。本例中,组合服务流程经过优化后,最大吞吐量可以提高到 26 个每小时(或 0.0072 个每秒)。

5 结论

在考虑服务状态和资源能力限制的情况下,组合服务流程的多个实例在执行过程中会在被调用成员服务上形成类似于 QN 系统中的排队现象,从而产生由交互引起的动态性能变化问题。现有研究中,针对组合服务流程动态性能优化的两阶段评估方法尚缺少这方面的考虑,并且动态问题只有在流程执行过程中才通过调度的思想进行优化,往往会产生其它更为复杂的问题。本文将吞吐量和响应时间引入到组合服务流程性能评估的流程级指标中,并借助排队网络系统的相关结论,提出了基于 BPEL 组合服务流程静态模型的动态性能定量预测方法。

在由成员服务和组合服务流程实例构成的排队网络系统中,吞吐量和响应时间不仅反映了成员服务资源的利用率,还反映了一定时间内组合服务流程的可执行能力。通过计算吞吐量和响应时间的临界值,不仅可以定量预测组合服务流程的动态性能,还能揭示组合流程的性能瓶颈节点,从而为流程优化提供指导。

参考文献

- [1] Chandrasekaran S, Miller J A, Silver G S, et al. Composition, performance, analysis and simulation of web services. *EM-Electronic Markets*, 2003, 13(2): 56-64
- [2] IBM, BEA Systems, Microsoft, et al. Business process execution language for web services version 1.1. <http://www.ibm.com/developerworks/library/specification/ws-bpel>: IBM, 2007
- [3] Zeng L Z, Benatallah B. Quality driven web services composition. In: *Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, 2003*. 411-421

- [4] Liu Y, Ngu A H H, Zeng L Z. QoS computation and policing in dynamic web service selection. In: Proceedings of the 13th International Conference on World Wide Web, New York City, NY, USA, 2004. 66-73
- [5] 蒋运承, 汤庸. 服务组合的质量估计模型. 小型微型计算机系统, 2006, 27(8): 1519-1525
- [6] Mohabey M, Narahari Y, Mallick S. A combinatorial procurement auction for QoS-aware web services composition. In: Proceedings of 2007 IEEE International Conference on Automation Science and Engineering, Scottsdale, Arizona, USA, 2007. 716-721
- [7] Claro D B, Albers P, Hao J K. Selecting web services for optimal composition. In: Proceedings of the 2nd International Workshop on Semantic and Dynamic Web Processes, Orlando, Florida, USA, 2005. 32-44
- [8] 郭于明, 孙延明, 郑时雄. 基于遗传算法的制造网络服务质量优化. 华南理工大学学报(自然科学版), 2007, 35(8): 59-64
- [9] 陈晶, 潘勋. 基于双重服务质量约束的网格资源调度策略. 计算机集成制造系统, 2008, 14(8): 1571-1578
- [10] Lazowska E D, Zahorjan J, Graham G S, et al. Quantitative System Performance: Computer System Analysis Using Queuing Network Models. Englewood Cliffs: Prentice-Hall Press, 1984. 15-28
- [11] Christensen E, Curbera F, Meredith G, et al. Web services description language (WSDL) 1.1. <http://www.w3.org/TR/wsdl>: W3C, 2001

Quantitative prediction of dynamic performance of BPEL-based composite service process

Jiang Feng^{* **}, Fan Yushun^{*}, Zeng Sen^{***}

(^{*} Department of Automation, Tsinghua University, Beijing 100084)

(^{**} IT Department, Agriculture Bank of China, Beijing 100005)

(^{***} Department of Automation, Guilin Air Force Academy, Guilin 541003)

Abstract

To bridge the current quality of service (QoS) research gap on model dependency and dynamic performance of composite service processes, this paper establishes two dynamic performance indices, i. e. Throughput (T) and Respond time (R), and represents the quantitative approach for dynamic performance prediction of composite service processes using the theories of queuing network (QN) system. Specifically, a bound computation is employed to business process execution language (BPEL) based composite service processes. The case study indicates that the proposed method is effective for dynamic performance prediction and can be used for optimization of composite service processes.

Key words: business process execution language (BPEL), composite service, dynamic performance, performance prediction, queuing network