

服务流程中的数据流处理

陈姣娟 曹 健

(上海交通大学计算机科学与技术系 上海 200240)

摘 要 服务流程需要处理服务之间大量的异构数据的交互,不同的数据流处理方式直接影响了服务流程的执行效率。阐述了服务流程模型中的数据流表示模型、数据映射机制与数据流验证机制,论述了服务流程运行中的数据流调度、数据存储以及传输等数据管理问题,分析了数据流处理在服务流程中的应用情况。最后,结合现有的数据流研究进展,提出了数据流研究的展望。

关键词 服务流程,数据流

中图法分类号 TP39 **文献标识码** A

Dataflow Processing in Service Workflow

CHEN Jiao-juan CAO Jian

(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract Service workflow needs to deal with a large number of heterogeneous data interactions between services. Different data processing approaches can directly affect workflow execution efficiency. Some data modeling issues such as dataflow implementation models, data mapping techniques and dataflow validation approaches were firstly presented. Then, several data management topics during workflow execution were discussed, including dataflow schedule, data storage challenges and data transfer challenges. In addition, the current situation of dataflow processing in service workflow was analyzed. Finally, combined with the existing data flow related works, some future research directions of dataflow were pointed out.

Keywords Service workflow, Dataflow

1 引言

随着服务计算、云计算等新型计算模式的不断发展和成熟,互联网上的软件服务越来越多。通过工作流模型将多个软件服务组合起来,支持更为复杂的功能已经成为一种常用的手段^[1]。服务流程^[2],是一种在活动中调用已经存在的软件服务并通过定义软件服务之间的交互关系,将多个软件服务的功能进行组合,从而实现新的功能的工作流模型。目前,服务工作流的应用越来越普及,并与各种新型的计算模式融合起来形成新的工作流形态。例如,服务工作流可以被应用于科学计算领域,形成科学工作流,服务工作流与网格计算结合起来形成网格工作流,服务工作流与云计算结合起来形成云工作流等。

服务流程可以整合分布在互联网上的各式各样的服务,这些服务有着天然的分布式性质,流程涉及大量的数据交互,需要处理各种异构数据之间的转换传递。当把服务流程应用于科学计算^[3]或者云计算领域时,涉及的数据量往往十分庞大,异构数据之间的映射转换以及大量的消息传递使得服务流程的数据流处理显得更加复杂。因此,近年来关于数据流

的研究日益增多,如何处理数据流以提高服务流程的执行效率成为研究的热点。

数据流处理主要包含两部分:服务流程模型中的数据流模型的表示与服务流程运行中的数据流调度、数据传递的方式。其中数据流模型体现了流程模型中活动之间的数据依赖关系,它确保数据建模的正确性,解决一些异构数据之间的映射关系;而数据流调度、数据传递的方式,则是流程引擎运行时与服务之间的数据交互的策略,发送、接收数据的协议。

经过多年的工作流技术的研究发展,数据流模型的表示已经拥有了一个良好的基础,现有的一些数据流模型的实现,已经能够十分清晰地表示流程模型中各活动所赋予的输入、输出参数之间的转换、传递,体现活动之间的信息交互关系。当一些特定的应用涉及的数据结构十分复杂或者数据量十分庞大时,只需在原有的数据流模型上做一定的扩展,即可完成数据流的表示。

关于流程运行过程中的数据流处理目前存在很大的问题,因为在传统的业务工作流^[4]中,流程的中心是活动,强调的是活动之间的协同方式及对数据的逐步处理过程;数据是从属,是对流程的补充。即使后来产生的一些专用于服务流

到稿日期:2012-03-26 返修日期:2012-09-10 本文受国家自然科学基金(61073021,61272438),上海市科委项目(11511500102,10DZ1200200),上海交通大学医工交叉项目(YG2011MS38)资助。

陈姣娟(1988—),女,硕士生,主要研究方向为工作流建模,E-mail:chan@sjtu.edu.cn;曹 健(1972—),男,博士后,教授,主要研究方向为业务过程建模、Agent 技术与应用和协同信息技术与系统。

程的流程语言如 BPEL, 数据处理能力还是比较弱, 当交互的数据量十分庞大时, 仍然无法确保流程运行的效率, 甚至可能因为数据传输产生的错误, 导致流程无法顺利运行。如今的时代是信息的时代, 科学计算、云计算的快速发展对服务流程的数据处理能力提出了新的要求, 使得数据流处理的重要性甚至超越了控制流, 如何进行大数据量的正确传递并保证执行效率, 是研究数据流处理的大方向。

本文对服务流程中的数据流处理进行了综述。第 2 节介绍了数据流的表示, 它是数据流处理的基础; 第 3 节阐述了流程运行过程中的数据处理; 第 4 节给出了数据流处理在服务流程中的应用情况; 最后总结全文, 指出数据流的未来研究方向。

2 数据流表示

数据流的表示是数据流处理的首要问题, 数据流一般有 3 种表示模型^[5,6]: 隐含于控制流的隐式数据流; 使用公共数据存储的隐式数据流; 显式数据流。隐含于控制流的数据流实现模型, 利用控制流在活动之间传递数据, 灵活性不高^[6], 适合于传统业务 workflow, 而服务流程中的数据流基本上采用后两种实现模型。本节对这两种实现模型进行描述, 并给出相应的例子。另外, 在服务流程中, 一个服务的输出数据与另一个服务的输入数据的格式往往并不一致, 数据之间的映射也是数据流表示的一个问题。最后, 合理正确的数据流建模才能使流程正确运行, 数据流的验证也是数据流表示的一个必不可少的问题。本节对几种服务流程语言中的数据流表示进行了总结, 如表 1 所列。

表 1 几种服务流程语言的数据流表示

	WSFL	BPEL	BPML
表示模型	显式	隐式	隐式
数据类型	支持	支持	支持
数据映射	MAP	XSLT	XPATH, XQUERY, XSLT

2.1 显式数据流模型

在显式数据流模型中, 与控制流相似, 数据流作为流程模型的一部分被显式地定义, 这些数据转换显示了数据从一个活动到另一个活动的流动, 如图 1 所示。图 1 中, 一条显式表示的数据流表示了活动 C 对活动 A 的数据依赖, 当活动 A 完成时, Data 通过数据流作为 C 的输入数据传递给活动 C, 活动 B 完全不知道 A 与 C 之间的数据依赖关系。显式的数据流表示使得流程中数据的流动十分清晰、明确, 直接显示了活动之间的数据依赖关系, 然而数据流的独立显示使得流程模型的结构变得复杂。

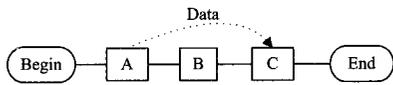


图 1 显式数据流模型^[5]

WSFL^[7]使用 dataLink 元素对数据流进行显式表示, dataLink 元素与表示控制流的 controllink 元素处于同一级。

2.2 使用公共数据存储的隐式数据流模型

使用公共数据存储表示的数据流模型, 通过操作流程模型中的公共数据来隐含地表示数据的流动, 如图 2 所示。图 2 中, 活动 A 产生 Data, 保存于流程的 data store 中; 活动 C 执行时, 从 data store 获取 Data 作为输入, 而不需知道是谁产

生 Data 的, 通过操作共同的数据 Data, 实现 Data 的隐式传递。相比于显式数据流模型与隐含于控制流的隐式数据流模型, 使用公共数据存储的数据流表示方法受到普遍的欢迎, 它没有显式数据流模型的臃肿, 表达能力也比隐含于控制流的数据流模型要强得多, 因此服务流程语言的数据流大都采用这种表达方式。

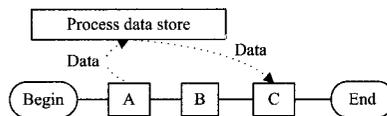


图 2 使用公共数据存储的数据流模型^[5]

已成为 Web 服务组合的事实标准的 BPEL, 使用“黑板”方式处理数据, 数据被活动共享, 数据流由赋值活动拷贝源数据变量到目的数据变量来隐式地体现, 这里的黑板即是一种公共数据存储。业务流程建模元语言 BPML^[8]中用上下文的概念定义流程或复杂活动的执行环境, 以及如何访问属性、如何交换消息等; 在同一个上下文中运行的活动通过上下文中定义的属性来交换信息。BPMN 作为 BPML 的可视化表现符号被创建, 是用于业务过程建模的一个标准的图形化标记^[9], 它不显式表示数据流, 数据由属性表示。BPMN2.0^[10]定义了元模型和执行语义, 数据模型中的 Data Associations 元素用来移动活动的输入输出等数据, 接收数据作活动的输入, 将活动的数据结果传递给其他数据来实现数据的流动。

2.3 数据映射机制

传统的企业内部业务过程自动化的业务流程中所处理的数据的格式往往是一致的, 流程模型更注重数据相继被哪些活动处理, 数据如何随着控制流流动。而服务流程集成了各种服务来实现一个特定的功能, 服务可以由不同的组织提供, 各服务的数据格式及表现方式往往不尽相同, 如何将各种不一致的数据进行映射以实现数据的流动, 也是各种服务流程语言需要解决的问题。

大部分服务流程语言都提供了数据映射机制。WSFL 的 dataLink 元素里的 map 元素指定了源活动与目的活动需要传递的具体信息, 可以将数据的一部分传递给目的活动; BPEL2.0 中增加了 XSLT 来实现复杂变量的转换传递; BPML 在将一个数据赋予属性或者将数据结果映射到消息时, 都使用赋值语句, 赋值语句可以使用表达式, 动态建立数据或者从一个或多个属性中导出数据^[8]; BPMN 中的 Data Association 有一个可选的转换函数, 当 Data Association 被“执行”时, 数据被拷贝到目标, 拷贝的数据依赖于是否有转换函数^[10]。

总体来说, 数据的映射都可以由一个转换表达式实现, 可以是 XPath、Xquery、XSLT 表达式, 也可以是一些自定义的扩展函数, 但目前的这种数据映射转换都是由人工进行设置的, 过程十分繁杂且容易出错。如何在一个已经建立好控制流的流程模型上进行自动或者半自动的数据流映射, 是一个十分有意义且具有挑战性的课题。

2.4 数据流验证

一个完整的流程模型除了要有正确的控制流描述的行为信息。还需要正确的数据流描述的数据交互信息。模型结构的正确性并不代表一个流程模型能够正确执行并且完全满足需求, 数据流建模的错误可能导致流程错误运行甚至终止。

数据流验证就是在控制流完全正确的流程模型中验证数据流的正确性。

文献[5]首次提出了数据流验证的重要性,分析了 workflow 模型中的数据可能产生的错误,并将错误总结为 7 大类:

- (1)数据冗余:活动的输出数据未被任何后继活动使用。
- (2)数据丢失:处于并行分支的两个活动产生相同数据时,一份更新会丢失。
- (3)数据缺失:活动输入数据的来源未被指定。
- (4)数据不匹配:来源活动产生的数据和消耗活动需要的数据格式不匹配。
- (5)数据不一致:数据在流程运行时被外部应用更新,导致不一致的数据版本。
- (6)数据误用:数据流动方向与控制流冲突,导致后继活动无法正确读取数据。
- (7)数据不充分:数据在语法、语义上是否足以描述一个完整的活动。

文献[5]系统分析了数据流可能的错误类型,但并未给出具体的错误检测算法,因此有不少的研究在文献[5]的分析基础上,采用活动依赖和数据依赖的依赖分析方法^[11,12]以及基于时序逻辑和反模式集成控制流和数据流验证^[13,14]等技术,设计了各自的数据流模型及相应的数据流错误检测算法来检测某几种错误类型。

BPEL 是服务流程建模的事实标准,BPEL 中的变量有作用域的限制,BPEL 的这种语法上的特殊性,使得它的数据流验证更为复杂。文献[15]采用变量的定义,使用链信息和数据依赖关系^[16],针对单个 BPEL 流程中的数据流进行分析和错误检测,检测出引用未初始化变量、变量未被引用等错误。

3 服务流程运行中的数据处理

相比于传统的业务 workflow,服务流程中软件服务分布在互联网上,流程引擎需要将服务之间的数据传输进行调度,在数据密集型应用中,大数据的传输、存储甚至可能导致流程的运行失败。如何提高引擎系统的执行效率及可靠性,降低总体成本,是服务流程执行中数据流处理的重点,主要有两个方向的研究,一个方向是流程引擎的整体数据流调度方案,另一个是流程运行中数据的管理。

3.1 数据流调度

文献[17]根据控制信息与数据的不同交换方式,将服务集成模型分成了 4 类:集中式控制流与集中式数据流(1C1D)、集中式控制流与分布式数据流(1CnD)、分布式控制流与集中式数据流(nC1D)、分布式控制流与分布式数据流(nCnD)。1C1D 模型概念简单,且容易实现,是目前最普遍的服务集成模型。BPEL 也属于 1C1D,BPEL 中隐式的数据流表示,使得 BPEL 引擎作为数据管理中心直接与服务的信息进行交换传递,当数据量十分庞大时,BPEL 引擎就成为了一个瓶颈,导致流程运行时间加长,甚至因为超时等待数据而使服务调用失败,从而导致流程运行失败。而集中式控制流与分布式数据流(1CnD)模型使得引擎不需直接参与服务之间的数据交互,数据直接在服务之间进行传递,在网络环境中充分利用了单一服务之间的带宽。文献[17]证明了 1CnD 在汇总成本和响应时间上大大优于 1C1D。除了 1CnD,也有人研究了 nC1D 及 nCnD 的实现,在分布式控制流模型中,控制信

息在单一服务之间进行传递,这种模型多出现于以“数据驱动”方式启动指令执行的数据流计算机架构中^[17]。相比于 1CnD,nC1D 及 nCnD 的实现更加复杂,有更多的挑战,如错误的处理、恢复、代码的分割等。文献[18,19]对这些问题进行了详尽的阐述。

总体而言,集中式的数据流调度方案使得大数据量的消息处理成为了流程引擎的一个瓶颈,如何实现分布式数据流的调度,甚至是分布式数据流与分布式控制流的调度,是研究服务流程,提高数据传输效率的一个大方向。

3.2 数据管理

当把服务流程应用于科学计算,如调用实现 k-means 聚类算法的软件服务时,涉及的数据量往往十分庞大,而且数据密集型应用流程的运行中数据处理的工作量远远大于了计算的工作量^[20],数据管理的重要性慢慢地体现出来。另外,近年来,云计算作为一种按使用量付费的新型服务交付和使用模式,成为科学界和工业界的研究重点,它使用户可以以按需、易扩展的方式获得公共可配置的计算资源,包括网络、服务器、存储设备、应用程序和服务^[21]。云 workflow 作为一种协同多个云服务实现更复杂功能的解决方案被提出,并且得到广泛的研究。云 workflow 可能涉及大量的计算密集型和数据密集型的大型云计算应用,云 workflow 系统的可靠性在很大程度上依赖于系统的一些数据管理问题(如数据的存储、存放、复制、传输等问题)的解决,云计算在带来了高性能计算和大量数据存储的同时,也带来了更多的挑战。

文献[22]详细阐述了科学云 workflow 系统的 3 个数据管理问题及相应的解决方法。

(1)数据存储问题:科学 workflow 运行时会产生大量中间数据,但并不是所有的中间数据都应该存储。由于数据被很多人共享,如何决定数据是否应该存储。该文认为记录数据依赖关系的元数据提供了中间数据产生过程,根据该信息可以在存储代价和产生代价之间进行权衡,以解决存储问题。

(2)数据存放问题:云中不同服务提供者的数据中心之间的数据传输需要成本,如何决定数据的存放位置以减少总体费用。该文使用聚类算法,将由一些共同活动处理的一些数据存放在同一个数据中心,以减少总体的数据移动费用。

(3)数据复制问题:数据的复制可以提高数据访问速度、减少数据在数据中心之间的移动并提高数据可靠性,如何决定数据应该复制多少份及存放地点。该文使用数据依赖关系决定复制数据的存放地点,采用数据使用率来决定数据的复制份数,以产生动态的数据复制策略。

除了上述问题外,文献[23]介绍了数据的预读取、数据的传输等问题:

(1)数据预读取问题:workflow 引擎可以根据数据依赖关系,得到需要的数据集以及期望的数据集访问顺序,根据这些信息,引擎可以异步地将一些数据预先读入活动附近的一些共享资源,以提高 workflow 执行效率^[24]。

(2)数据传输问题:数据的传输可能因为服务器的超载、瞬态网络失败等原因而失败或者超时,如何解决问题需要看错误发生的源头,比如访问输入数据失败时,可以在 workflow 规划期间从数据复制目录或者在 workflow 运行期间的回撤机制中寻找一个新的数据源,目的服务器暂时超载时可以进行重试。

4 数据流处理的应用

BPEL 中的数据流处理比较简单,流程引擎作为数据管理中心,直接从公共变量中读取数据,将其作为服务的输入参数,调用服务并获得服务输出的数据,保存于公共变量中,其中的 Web 服务调用采用 SOAP 的基于值的数据传输方式,实现与 Web 服务的数据交互。当传递的数据量十分庞大时,流程引擎的数据流处理成为一个瓶颈。

因此,文献[25]提出了 BPEL 的一种扩展‘BPEL data transitions (BPELDT)’,实现了 1CnD 的服务集成方式。它通过数据依赖关系显示连接各服务来显示地定义数据流,采用一种新的 Web 服务框架 Data-Gray-Box Web Service (DGB)[26],仍使用 SOAP 消息实现服务请求,请求信息包括输入数据的访问路径信息和输出数据的存储地点信息。而具体数据的传递使用新的一个数据层来处理,该数据层能够根据 SOAP 中的输入输出数据的访问信息,透明地调用合适的专用的数据传播工具进行服务之间的数据传输,使得 BPEL 引擎不再负责具体的数据传输。

近几年,云计算相关技术的出现,使得越来越多的计算资源如存储设备、软件程序等以云服务的形式发布在网络上,从而提供高性能的计算与大容量的存储。云工作流作为协同多个云服务的交互实现复杂功能的解决方案成为了科学界与工业界研究的重点,而保证每一个云工作流实例执行的高效性、可靠性,是云工作流系统的重要任务。云工作流在多个云服务之间传递大量的数据,而网络环境与云服务都是不可控的,云服务的失效、数据传输的错误,都会影响工作流系统的运行。

因此,为了充分利用云计算的优势、降低总体成本,针对上一节中阐述的一些数据管理问题,有很多机构进行了大量的研究,并提供了一系列的技术加以解决[20]。比如,为了减少数据传输的时间,可以采用数据并行、数据异步实时传输、数据传输率控制等技术;为了减少数据存储不足产生的错误,可以采用数据及时清除技术。

针对上述技术,也有一些研究提出了特定的算法,以提高系统的运行效率以及减小运行成本。比如文献[27]提供了一个数据传输率的控制框架,即由 workflow 规划师指定数据何时何地、以多少速率进行传输,以提高传输效率;Dong Yuan 等人也做了一系列的研究,文献[28,29]利用数据产生过程的描述,建立中间数据依赖图,设计了减少总体费用的中间数据存储算法;文献[30]设计了基于依赖分析的聚类数据存放策略来减少流程运行中的数据移动次数。

总的来说,云工作流的数据流处理应用主要包括如何设计数据的传输以极小化流程的最大完工时间;根据云上服务的交付使用模式,如何设计数据在云上的存储,以减少总体费用;如何根据延迟、网络带宽、系统负载等因素在分布式环境下选择合适的数据副本;如何及时地清除不再使用的数据等等。云计算的潜力是无限大的,而且涉及运行效率的因素也十分繁多,这类研究还将持续进行。

5 数据流研究的展望

如今的时代是信息的时代,工作流的发展也从传统的业

务工作流发展到处理大量数据的科学工作流。服务流程与科学计算、云计算的结合,使得数据流的处理成为了服务流程的重点。各种各样的云计算标准、云计算平台的提出,使得云服务之间的数据协同显得更加重要。

由于如今的应用涉及的数据量越来越大,数据的传输、存储成为了瓶颈,因此,数据流处理仍然面临着许多有待解决的挑战,针对数据流的研究还将持续进行,总结起来主要包括以下方面:

数据流的利用:在服务流程的运行中,数据源信息记录了数据之间的交互及数据的产生过程,如何充分利用这些数据源信息,实现服务流程的挖掘、预测流程运行结果等,是十分有意义的。

数据驱动流程:现在的很多科学计算应用都是数据驱动的,如何实现完全以数据流为中心的服务流程,是研究数据流的一个大方向。服务流程的数据流约束体现了各服务的输入数据及输出数据之间的交互关系,根据数据的依赖分析,可以得到各服务之间的依赖关系,从而可以决定服务流程的流程模型。

云工作流的数据处理:云计算作为一种新的服务的交付和使用模式被提出,并被广泛研究,计算能力和存储能力可以作为服务被使用,如何解决云服务交互过程中的大数据量的数据存储、传输等问题,最大限度地利用云计算的优势,提高流程运行速率及可靠性,将成为今后研究的重点。

结束语 本文主要对服务流程的建模、运行阶段中的数据流处理进行了大致的介绍,包括不同的数据流模型的实现方式、数据流模型的映射机制与验证机制、流程运行中数据流的调度和一些数据管理方面的挑战。由于服务流程与科学计算、云计算的结合,使得流程涉及的数据量越来越大,数据流的处理越来越重要,甚至超越了控制流。数据的传输、存储成为了瓶颈,直接影响着流程运行的效率,甚至可能导致流程运行失败。可以预见,随着云计算发展的成熟,将会涌现更多、更完善的数据流处理的方法,从而可以更好地利用云计算的优势。

参 考 文 献

- [1] 倪晚成,刘连臣,吴澄. Web 服务组合方法综述[J]. 计算机工程,2008,34(4):79
- [2] Ganesarajah D. Web service workflow [EB/OL]. <http://www.doc.ic.ac.uk/~dg197/project>,2001-06
- [3] Akram A, Meredith D, Allan R. Evaluation of BPEL to scientific workflow[C]// Cluster Computing and the Grid (CCGrid). IEEE Computer Society,2006:269-274
- [4] van der Aalst W M P, van Hee K M. 工作流管理-模型、方法和系统[M]. 王建民,闻立杰,等译. 北京:清华大学出版社,2004:29
- [5] Sadiq S, Orłowska M, Sadiq W, et al. Data flow and validation in workflow modeling[C]// Fifteenth Australasian Database Conference (ADC). volume 27 of CRPIT, Dunedin, New Zealand, 2004:207-214
- [6] 李伟刚,王文斌,沈钧毅. 产品开发过程中的数据流建模与分析[J]. 计算机学报,2008,31(10):1804
- [7] Leymann F. Web Services Flow Language (WSFL 1. 0) [EB/

OL]. <http://www.homes.doc.ic.ac.uk/~hf1/phd/papers/to-read/WSFL.pdf>

- [8] Arkin A. Business Process Modeling Language (BPML) [EB/OL]. Working Draft 0.4, BPML, March 2001 (cf. <http://www.bpmi.org/>)
- [9] Weske M, Vossen G, Puhmann F. Workflow and Service Composition Languages [M] // Bernus P, Mertins K, Schmidt G. Handbook on Architectures of Information Systemst, Springer, 2006;369-390
- [10] Business Process Model and Notation (BPMN) Version 2.0 [EB/OL]. (visited 5th October 2010). <http://www.omg.org/spec/BPMN/2.0/Beta2/PDF>
- [11] Sun S X, Zhao J L, Nunamaker J F, et al. Formulating the Data Flow Perspective for Business Process Management [J]. Information Systems Research, 2006, 17(4); 374-391
- [12] Sundari M H, Sen A K, Bagchi A. Detecting Data Flow Errors in Workflows: A Systematic Graph Traversal Approach [C] // 17th Annual Workshop on Information Technology & Systems (WITS-2007), 2007
- [13] Trčka N, van der Aalst W, Sidorova N. Analyzing Control-Flow and Data-Flow in Workflow Processes in a Unified Way [R]. CS 08/31. Eindhoven University of Technology, 2008
- [14] Trčka N, van der Aalst W, Sidorova N. Dataflow anti-patterns: Discovering dataflow errors in workflows [C] // Conference on Advanced Information Systems Engineering. 2009; 425-439
- [15] Yang Xue-hong, Huang Jun-fei, Gong Yun-zhan. Static Data Flow Analysis and Anomalies Detection for BPEL [C] // International Conference on Test and Measurement. 2009; 18-21
- [16] Zheng Yong-yan, Zhou Jiong, Krause P. Analysis of BPEL Data Dependencies [C] // Software Engineering and Advanced Applications, 2007. 33rd EUROMICRO Conference. 2007; 351-358
- [17] Liu D, Law Kincho H, Wiederhold Gio. Analysis of Integration Models of Service Composition [C] // Proceedings of Third International Workshop on Software and Performance. 2002; 158-165
- [18] Nanda M G, Chandra S, Sarkar V. Decentralizing execution of composite web services [C] // Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications. 2004; 170-187
- [19] Binder W, Constantinescu I, Faltings B. Decentralized Orchestration of Composite Web Services [C] // Proceedings of the International Conference on Web Services, ICWS'06. 2006; 869-876
- [20] Pandey S, Buyya R. Scheduling and management techniques for data-intensive application workflows [C] // Data Intensive Distributed Computing: Challenges and Solutions for Large-scale Information Management. 2009
- [21] Mell P, Grance T. The NIST Definition of Cloud Computing [OL]. <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>
- [22] Yuan Dong. Data Management in Scientific Cloud Workflow Systems [C] // The first CS3 PHD Symposium. 2010; 58-61
- [23] Deelman E, Chervenak A. Data management challenges of data-intensive scientific workflows [C] // IEEE International Symposium on Cluster Computing and the Grid (CCGrid'08). 2008; 687-692
- [24] Chervenak A, Deelman E, Livny M, et al. Data Placement for Scientific Applications in Distributed Environments [C] // 8th Grid Computing Conference. 2007; 267-274
- [25] Habich D, Richly S, Grasselt M, et al. BPELDT- data-aware extension of BPEL to support data-intensive service applications [C] // Emerging Web Services Technology. vol. II, 2008; 111-128
- [26] Habich D, Preissler S, Lehner W, et al. Data-grey-box web services in data centric environments [C] // Proceedings of the 2007 International Conference on Web Services (ICWS 2007). 2007; 976-983
- [27] Park S M, Humphrey M. Data Throttling for Data-Intensive Workflows [C] // IEEE International Parallel and Distributed Processing Symposium (IPDPS 2008). 2008; 14-18
- [28] Yuan D, Yang Y, Liu X, et al. A cost-effective strategy for intermediate data storage in scientific cloud workflows [C] // 24th IEEE International Parallel & Distributed Processing Symposium. 2010; 1-12
- [29] Yuan D, Yang Y, Liu X, et al. A data dependency based strategy for intermediate data storage in scientific cloud workflow systems [J]. Concurrency and Computation: Practice and Experience, 2010
- [30] Yuan D, Yang Y, Liu X, et al. A data placement strategy in scientific cloud workflows [J]. Future Generation Computer Systems, 2010, 26(8); 1200-1214

(上接第 13 页)

- [4] Kyriazis A, Mathioudakis K. Enhance of fault localization using probabilistic fusion with gas path analysis algorithms [J]. Journal of Engineering for Gas Turbines and Power, 2009, 131(5); 51601-51609
- [5] Jones J A, Harrold M J, Stasko J. Visualization of test information to assist fault localization [C] // Proceeding of the 24th International Conference on Software Engineering. 2002; 467-477
- [6] Jones J A, Harrold M J. Empirical evaluation of the tarantula automatic fault-localization technique [C] // Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering. 2005; 273-282
- [7] Liblit B, Naik M, Zheng A X, et al. Scalable statistical bug isolation [C] // Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). 2005; 15-16
- [8] Schach S R. Object-oriented classical software engineering [M]. Beijing: China Machine Press, 2007; 490-193
- [9] Liu C, Yan X, Fei L, et al. SOBER; statistical model-based bug localization [C] // Proceedings of the 13th ACM SIGSOFT Symposium on Foundations of Software Engineering. 2005; 286-295
- [10] Renieris M, Reiss S P. Fault localization with nearest neighbor queries [C] // Proceedings of the 18th International Conference on Automated Software Engineering. 2003; 30-39
- [11] Zeller A. Isolating cause-effect chains from computer programs [C] // Proceedings of the 10th ACM SIGFOFT Symposium on Foundations of Software Engineering. 2002; 1-10
- [12] Haykin S. Neural networks-A comprehensive foundation [M]. Beijing: Tsinghua University Press, 2001; 484-508