# 浙 江 大 学

## 硕士学位论文文献综述

（专业学位）

文献综述题目： 复杂事件处理在应用性能监控中的应用


姓　名： 常文龙

学　号： 21451136

专　业： 软件工程

院　别： 软件学院

导　师： 孙建伶


日期：二零一六年 9 月

# 目　录

# 一、应用性能监控概述

软件的整个生命周期[1]包括问题定义、可行性研究、需求分析、开发测试阶段、维护。软件维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投入使用后，由于多方面的原因，软件不能继续适应用户的要求。要延续软件的使用寿命，就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。如何获取软件的错误，软件的改进点，我们可以将软件的使用过程监控起来。因此我们引入应用性能监控[2]。

绝大部分的应用系统都会提供对自身运行状况进行监控的功能，以方便用户更加深入和全面地了解系统的性能和当前的健康情况，更好的维护应用。

从性能监控系统测量的对象方面讲，需要监控的对象有：方法的时间响应，程序运行异常信息，业务数据的处理情况。对于系统资源的利用情况，如 CPU 利用率、内存利用率、网络带宽等。几乎所有的性能监控系统提供对上述这些监测数据的采集和可视化展示，但对这些收集过来的监控数据进一步分析，得出更有价值的监控信息的支持还很限，更不用说允许用户自主定义监控指标产生报警信息这一功能。比如，我们采集到一条 CPU 使用率的监控数据，CPU 某一时刻的使用率达到 95%，这条数据其实不能准确说明 CPU 的负载情况，但若在过去半小时内采集到的 CPU 使用率均是大于 90%，这时我们就需要生成报警信息通知到相关负责人。目前有少部分性能监控系统开始关注对收集的原始监控数据的分析，允许用户自主定义的监控指标，并对触发监控指标的情况生成报警信息。

近年来国内外也涌现出一些成熟的 APM 解决方案[3]，国内有 OneAPM、听云，国外有 NewRelic 、 APP Dynamic。他们提供丰富的可视化展示，托管之上

的应用，可以查看系统资源的利用情况，方法的执行时间调用栈和慢查询 sql 等大多数应用共同关注的性能监测点。对于不同应用的具体业务逻辑，不能提供有效的监控。小公司一般将应用托管在 APM 服务提供商提供性能监控系统上。大型公司大多选择自主研发适用于公司业务的性能监控系统。

## 二、复杂事件处理概述

复杂事件处理[4]技术作为一种实时数据处理技术，近些年来逐渐走入人们的视野。它的起源与主动型数据库系统中的触发器、数据流处理系统、发布订阅系统等都有联系；它以从传统数据库技术中独立出来，成为一个新兴的研究领域。在传统的数据库系统中，数据首先被存储起来，在用户查询后返回结果;而复杂事件处理技术则是先将查询存储起来，当符合查询条件的事件发生时，实时地通知该查询的来源，这就实现了数据的实时处理。

当今事件处理技术正快速成为今天信息社会处理信息的基础。这个技术出现在 20 世纪 90 年代末期，但是事件处理并不是新的东西。在 20 世纪 90 年代，有一些研究工程指定去开发事件处理新的原则，被叫做复杂事件处理(以下简称 CEP)。CEP 的概念最早是由斯坦福大学的 David C．Luckham 提出的[5]。主要用来研究如何从海量基本事件中抽取、组合、发现或者总结抽象出更高层、更复杂的事件。它是 1 种新兴的基于事件流的技术，将系统数据看作不同类型的事件，通过分析事件问的关系，建立不同的事件关系序列库，利用过滤、关联、聚合等技术，最终由简单事件产生高级事件或商业流程。可以为物联网、互联网计算、云计算中的上层应用提供具有复杂语义的发现、监测、预警等基础服务。

近年来 CEP 在实时数据流处理、商业活动监控、商业过程管理的自动化服务体系中扮演着越来越重要的角色。学术界和企业界对复杂事件处理领域的关注日益剧增。在 2000 年时已经有人关注基于事件的系统和对数据流的处理。但是从 2004 年开始关于复杂事件处理的研究报道才逐渐增多，也出现了一批相关开源项目和工具。现在国际上已经有了很多较成熟的复杂事件处理产品和项目。例如商业领域的产品有 BEA Systems、TIBCO Business Events、Alert CEP3 .0、IBM StreamBase[6]和 D EsperTech[7]等。。研究机构中具有代表意义的是 Cornell University 的 Cayuga[8]、University of California，Berkeley 的 Telegraph CQL 和 Brandeis University、Brown University 和 MIT 合作开发的流式系统 Aurora、Medusa 以及由此演化而来的 Borealis 等等[9]。

今天 CEP 主要集中应用在业务信息系统上，用来管理业务的更高等级类型的事件，但是我们能预测到 CEP 将应用于任何种类系统产生的事件类型。CEP 体现了构建应用程序的原则，嵌入 CEP 技术的 IT 系统使企业能够跟上信息的流动。CEP 的目标是让企业 IT 基础设施所有层的事件流的信息被发现，理解高层次的管理目标和业务流程的影响，并采取实时的行动。比如处理 RFID 技术所创建的事件[10]。CEP 采用的检测技术，如很多事件的复杂模型检测，事件流处理，事件关联和抽象，事件层次结构，如因果关系，成员，和定时的事件之间的关系。CEP 还可以补充和促进某些技术，如面向服务的架构(SOA)，事件驱动架构(EDA)[11]和业务流程管理(BPM)。

事件处理软件的发展，以及对 CEP 设计问题的更深刻的理解和对更快更智能的业务处理的需求，使得 CEP 变得越来越流行。现在已经广泛应用在金融、系统自动化交易、BAM、RFID、高级检测系统、欺诈检测等领域，虽然应用的

范围很广，但是他们都是独立存在的，相互之间并没有关联，而往往在一个环境下的不同系统或者同一系统下的不同环境之间存在着某种联系，比如业务活动和系统安全方面的关联等，在软件行业关于事件处理的发展空间很大，需要不断地深入研究。而且随着市场空间的扩大，多种实现之间出现的竞争，需要标准化来更好的为行业提供服务。

# 三、复杂事件处理相关技术

## 3.1 规则定义

规则定义的关键在于定义语言的选择。在复杂事件处理引擎中，定义规则的语言称为事件处理语言。事件处理语言是一种面向终端用户的高层语言。一般在设计复杂事件处理语言的时候，会遇到两个冲突的需求。一方面，要求它是一个可机器执行的语言，即需要支持复杂事件检测模型的高效创建和表达式的自动降解。另一方面，语言的语法和语义需要是高层的和直观的，以方面编程人员或者用户的事件表达式的书写。复杂事件处理引擎通常需要对这两个要求进行权衡设计合适的语言或者选择两种描述语言。

事件处理语言可以分成三个类型：组合操作表达式、数据流查询语言和产生式规则。组合操作表达式通过使用不同的组合操作符将单个事件进行组合并在此基础上进行表达式的嵌套来描述复杂事件。使用组合操作表达式的系统包括 IBM 主动中间件和 RuleCore[12]等。，数据流查询语言是对 SQL 结构化查询语言的拓展，首先将输入流中的事件转换成数据库中的关系，然后在这些关系上执行查询，最后将查询结果换成数据流。现有的主流的 CEP 引擎都采用数据流查询语言，

例如 StreamBase，Coral8，Aleri，Esper 等等。产生式规则指定了当特定状态到达时应该执行的相应的活动，产生式规则的推理过程与 CEP 处理事件的过程非常相似，采用产生式规则的系统有 Jboss Drools Fusion[13]，XChange 等。

## 3.2 事件检测模型

事件检测[14]是复杂事件处理的核心环节，该过程是发现能够构成复杂事件的所有基本事件的过程。复杂事件处理系统中主要的事件检测模型包括：自动机检测模型，Perri 网[15]检测模型、基于树的检测模型和基于图的检测模型。

(1)自动机检测模型

考虑到复杂事件的表述方式类似于正则表达式的表述方式，而正则表达式通常使用自动机来进行模型构造，因此事件的检测过程也可以采用自动机模型来描述。在自动机模型中，在构成复杂事件的简单事件到来时，自动机就会进入到另一个状态，即发生状态的跃迁，当自动机到达最后的可接受状态时，则表明一个复杂事件的成功匹配了。由于传统的自动机模型不能描述事件之间的约束条件(包括时间限制以及内容约束等)的能力，因此需要对自动机进行扩展，需要在逻辑谓词上增加时间限制或者内容约束的条件并且需要将这些约束信息使用额外的结构来存储。使用自动机模型匹配事件[16]的系统包括 ODE，SASE，Cayuga 等。

(2)Petri 网检测模

Petri 网检测模型是一种比较常用的事件检测模型。心删网主要由库所、变迁、有向弧以及托肯等组成。其中库所主要包含输入库所和输出库所两个类型，输入库所用来对事件模式的成分进行建模，输出库所则为复杂事件模式进行建

模。除了这两种库所之外，还存在辅助库所来描述事件之间的依赖关系[17]。同样，托肯也包括两种类型：主要托肯以及辅助托肯。主要托肯起主导作用，托肯的相关内容取决于主要托肯所在的位置。Petri 网的基本思想是层次分析事件的逻辑结构和条件约束，逻辑结构使用库所以及变迁之间的连接关系来表述，而条件约束可以通过哨函数或者失效函数来描述相应的时间限制以及内容约束等。Petri 网天生的层次性结构以及并发的特性使得其很适合用来处理事件的检测。

(3)基于树的检测模型

该模型首先将需要检测的复杂事件以树的形式构造出来，其中叶子节点代表着构成复杂事件的简单事件，中间节点代表着每个层次上的复杂事件，根部节点代表了最终的复杂事件，如果成功到达根部节点，则说明了复杂事件的一次匹配成功。

此模型下的事件检测过程是一个递归的过程，当构成复杂事件的一个简单事件到来时，叶子节点就会接收此简单事件，并且通知给它的父亲节点，父亲节点接收下属节点传递的简单事件，根据预定义的规则匹配现有的事件元素，如果成功则将该事件向上通知给其父亲节点，以此类推，直至到达根节点，如果匹配成功，则说明一类复杂事件的发生，否则，则说明复杂事件没有发生。

一般情况下，基于树的检测模型通常有两种方式[18]，一类是自底向上的检测，另一类是自顶向下的检测。前者是一种自发的检测方式，当需要的事件到来时，子节点会通知父亲节点，触发其完成进一步的检测过程；而对于后者来说，是一种父亲节点主动询问子节点的方式，这种方式比较适合否定事件的检测。采用基于树的检测模型的复杂事件处理系统有Esper等。

(4)基于图的检测模型

此模型与上面提到的基于树的检测模型类似,首先需要用一个无环图来描述一个复杂事件。事件用节点来表示,而事件之间的规则可以用图的边来描述,一些时间限制以及参数约束的条件可以在节点中描述。当检测到相应的事件时,节点负责给出标记,当所有节点都有标记后,则说明了一个复杂事件出现了。采用基于图的匹配模型的复杂事件处理系统[19]有SNOOP以及TelegraphCQ等[20]。

## 3.3 其它相关技术

### 3.3.1 窗口技术

除了两个核心技术外,复杂事件处理引擎中经常要使用到的一个重要技术就是窗口技术[21]。在复杂事件处理过程中,时刻有大量的事件涌入到系统中,这些事件形成的事件流是没有边界的。如何在无边界的事件流上执行操作是复杂事件处理引擎需要考虑和解决的问题。而窗口技术就提供了一个很好的解决方案,它可以对事件流进行截取,因此可以利用窗口技术来指定事件流中需要处理的部分。

现有的系统中存在着不同类型的窗口,包括时间窗口和长度窗口。前者是以时间划定界限,例如限定一个操作只能处理上一个5分钟内的事件。后者则以事件的个数作为界限,例如限定一个操作只能处理最近到来的10个事件。

另外,根据窗口的移动方式,又可以进一步划分为下面几种窗口模型:

(1)固定窗口模型,该窗口是固定不动的。例如,可以使用固定窗口来处理"从2012年1月1日到2012年2月1日之间发生的事件"。

(2)界标窗口模型,该模型有一个固定的下界,上界会随着新的数据项的到来而前移。该模型可以来处理"从2012年1月1日以后发生的事件"。

(3)滑动窗口模型，该模型是最常见的模型。滑动窗口具有固定的尺寸，当新的数据项进入系统后，窗口的上界和下界都会同时前移。滑动窗15模型可以用来处理“最近接收的lO个事件”。

(4)窗格和翻滚窗口模型，这两个模型是滑动窗口的变形。每当k个事件进入系统时，窗15的上界和下界都同时前移k个元素。它们两者的不同之处在于，窗格窗口的尺寸大于k，而翻滚窗口的尺寸小于等于k。实际上，翻滚窗口保证了每次窗口的移动窗口内的所有元素都发生了变化，但这不适用于窗格窗口。举个例子，假设需要计算上个星期的平均气温时，如果是每天中午测量一次，则可以采用窗格窗口模型；如果是每周日中午测量一次则可以使用翻滚窗口模型。

窗口技术[22]的主要难点在于窗口大小的确定。如果窗口过小，则很可能出现不正确的检测结果，而窗口过大则对内存有很高的要求，因此当前对于窗口技术的研究主要集中在窗口大小的设置上。

### 3.3.2 容错技术

在复杂事件处理过程中，很可能因为各种各样的错误而产生不正确的结果，这些错误包括时间的不同步，有损耗的通信以及事件的递交过程中产生的乱序等[23]。因此在复杂事件处理引擎中引入容错技术是非常有必要的。虽然大部分的引擎的做法是简单地忽略这些错误，但现在也有不少对于容错技术的研究。STREAM就提出了一个近似查询处理的方案。Wasserkrug S等则针对事件出现的不确定性提出了一个构造概率空问的机制，使用贝叶斯网络来确定事件出现的概率[24]。Romdhane R等人则对不确定事件进行建模，并且使用逻辑与概率结合的方法来识别事件[25]。针对事件之间可能存在的乱序现象，LiuM等人提出了两个应对策略，一个是使用了补偿机制的积极策略，另一个是保证所有事件有序的情

况下才进行输出的保守策略[26]。总体来说，现在对于复杂事件处理过程中使用的容错技术的研究还很有限,要使容错技术变得成熟并应用到复杂事件处理引擎上还需要一段时间。

### 3.3.3 预测技术

随着复杂事件处理技术的广泛应用,事件预测这个新的功能也逐渐加入到复杂事件处理引擎中来，特别是面对股票市场、金融行业等领域，预测技术可以很好地进行股票趋势分析，可以预防金融犯罪。通常事件预测需要与人工智能结合起来，利用机器学习技术来对历史事件进行学习并预测即将发生的事件。

Antunes M等人针对离散事件提出了一个基于贝叶斯的预测方法来支持事件的在线预测[27]。Rudin C 等人则使用关联规则与贝叶斯技术相结合的方法来预测新的事件[28]。此外，Tqth G 等人提出了一个复杂事件处理与预测分析技术结合起来的方案，对两者结合的关键部分给出了定义[29]。由于事件预测技术在复杂事件处理领域里属于一个新技术,因此目前的研究还比较少,但其应用价值决定了预测技术是未来复杂事件处理领域的一个研究热点。

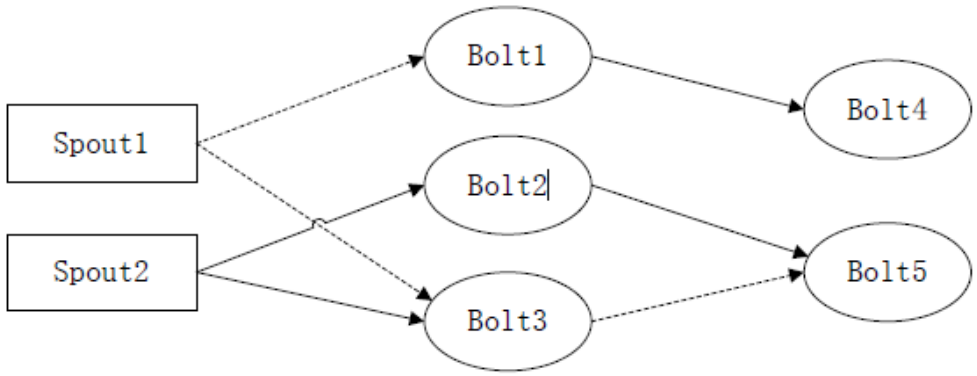
## 四、分布式实时流处理框架

Storm[30] 是一个分布式实时流处理平台，运行在 Storm 集群上的应用被称为拓扑结构（Topology）[31]。

### 4.1 Storm Toplogy

一个 Storm 的拓扑结构是整个计算的数据流图，其中的每个元素被称为

Spout，或 Bolt，中间通过数据流（Stream）传输。数据流是没有边界的数据元组（Tuple）序列，一个数据元组是一串任何类型的事件对象。数据流的输入来源是 Spout 节点，可以从外部应用程序读取数据，如股票交易数据、传感器测量值或程序的日志信息。数据流被 Bolt 节点消费，Bolt 作为逻辑计算节点，处理数据流，可以发送到下一个 Bolt 节点进行进一步的逻辑计算。Bolt 的处理逻辑可以是多样的，如对数据流进行过滤，聚集，连接等复合计算，也可将数据流存储到数据库。下图是拓扑结构的一个例子。




Storm Topology 图

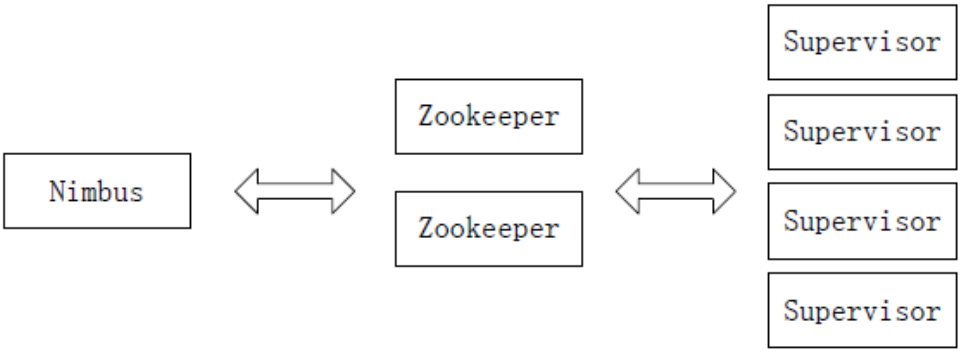
Topology 可以认为是 Storm 运行计算逻辑的蓝图，在运行时，topology 的每一个组件将运行一些任务（Task）。同一个组件中的每个任务执行相同的逻辑代码，但是使用其他线程执行在不同实例。一个任务从输入队列接收一组事件，进行处理，也可发送新的 tuple 作为事件输出流。数据流可以根据流分组（Stream Grouping）策略进行分发，如 shuffling grouping 随机分发数据流，all grouping 广播式发送，每个 Bolt 都将接收到一份数据流，field grouping 按 tuple 的字段分组。

总的来说，Topology代表一个完整的业务处理逻辑图，主要由Spout和Bolt组成，数据流在Spouts和Bolts之间流动。Storm集群可以同时运行多个Topologies。一旦Topology被提交，将永远运行，直到手动杀死进程。提交的Topology运行在

Storm集群上拥有多个worker进程。每个worker运Topology的一部分tasks，在运行过程中workers和tasks的数量不可以被改变。每个Bolt任务从输入队列接收数据，并发送数据给下一个Bolt。而Spout只能发送数据。任务节点之间的交互都通过消息中间件完成。

## 4.2 Storm Cluster

Storm集群包括一个主节点和多个工作节点。主节点Nimbus运行守护进程，负责接收提交的topologies，给工作节点布置任务，分配代码及故障检测。每个工作节点都会运行一个守护进程Supervisor，监听工作，控制工作进程的启动与销毁。此外，Zookeeper是针对大型分布式系统的可靠协调的开源系统，提供配置维护、命名服务、分布式同步、组服务等功能。




Storm Cluster 图

由于Nimbus和Supervisors是无状态，快速可恢复的，Storm依靠Zookeeper存他们的配置和状态，使其具有很好的健壮性。Storm编程模型有利于开发者建立topology，专注于应用逻辑，而不需要关心组件的分布、交互与底层网络传输。

# 参考文献


[1] 顾明.软件工程中几种常用软件生命周期模型的简介[J].计算机时代,2003(1).

[2] 陈丽,关少珊.应用性能管理概述[J].科技信息(学术研究),2008 年 02 期.

[3] 宋义华,班孝明.应用性能管理分析与研究[J].软件服务,2014.6(10):60-62.

[4] Luckham,David "The Power of Events:An Introduction to Complex Event Processing in Distributed Enterprise Systems", Addison Wesley Publishers, 2002.

[5] Etzion O,Niblett P.Event processing in action[M].Manning Publications Co.,2010.

[6] Arasu,A.,Babu,S.,Widom,J.:The CQL continuous query language:semantic foundations and query execution.The VLDB Journal-The International Joumal on Very Large Data Bases 15(2),121-142(2006).

[7] http://esper.codehaus.org/.

[8] Demers,A.,et a1.:Cayuga:A general purpose event monitoring system.CIDR(2007)

[9] D.J.Abadi,Y.Ahmad,M.Balazinska,U.C,etintemel,M.Cherniack,J.-H.Hwang,W.Lindner,A.Maskey,A.Rasin,E.Ryvkina,N.Tatbul,Y.Xing,and S.B.Zdonik.The design of the borealis stream processing engine.In Proc.CIDR,pages 277-289,2005.

[10] 孙林超,陈群,康庄庄.分布式 RFID 复杂事件处理关键技术的研究[J].计算机工程与应用,2011,47(22):105-109.

[11] Brenda M.Miehelson.Event-Driven Architecture Overview. Patricia Seybold Group.February 2006.

[12] rulecore.com[EB/OL].http://www.rulecore.corn/.

[13] Bali M.Drools JBoss Rules 5.0:Developer's Guide[M].Packt Publishing,2009.

[14] 王建,董广智,柳军飞.支持复杂事件处理的业务流程建模研究[J].计算机工程与设计,2012,33(6) .

[15]Esparza J,Nielsen M.Decidability issues for Petri nets[J].Petri nets newsletter, 1994,94:5-23.

[16] Nagy K A.Distributing Complex Event Detection[Z].2012.

[17] 魏永超,陈立军.数据流上复杂事件处理系统 Eagle 的设计与实现[C]第二十五届中国数据库学术会议论文集(一).2008 .

[18] 张昕.基于复杂事件处理的金融软件系统实现及改进[D].浙江大学,2007.

[19] 郑明秀,付春常,杨明根.复杂事件描述语言事件表达式的研究[J].计算机技术与发展,2012,22(7):113-115.

[20] Dittrich K R,Gatziu S,Geppert A.The active database management system manifesto:A rulebase of ADBMS features[M].Rules in Database Systems, Springer, 1995,1-17.

[21] Cugola G,Margara A.Processing flows of information:From data stream to complex event processing[J].ACM Computing Surveys,2012,44(3):15.

[22] Paton N W,D I Az O.Active database systems[J].ACM Computing Surveys (CSUR),1999,31(1):63-103.

[23] O'Keeffe D,Bacon J.Reliable complex event detection for pervasive computing. Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems,Cambridge,United Kingdom,2010.New York,NY,USA:ACM,2010:73-84.

[24] Wasserkug S,Gal A,Etzion O,et a1.Efficient Processing of Uncertain Events in Rule-Based Systems.Knowledge and Data Engineering,IEEE Transactions on. 2012, 24(1):45-58.

[25] Romdhane R,Boulay B,Bremond F,et a1.Probabilistic Recognition of Complex Event.Proceedings of the 8th international conference on Computer vision systems,Anipolis, France,2011.Berlin,Heidelberg:Springer,2011:122-13l.

[26] Liu M,Li M,Golovnya D,et a1.Sequence Pattern Query Processing over Out-of-Order Event Streams.Proceedings of the 2009 IEEE International Conference on Data Engineering,Shanghai,China 2009.Washington,DC,USA:IEEE,2009:784-795.

[27] Antunes M,Turkman M A A,Turkman K F.A Bayesian Approach to Event Prediction. Journal of Time Series Analysis.2003,24(6):631-646.

[28] Rudin C,Letham B,Salleb-Aouissi A,et a1.Sequential Event Prediction with Association Rules.MIT web domain,2011.

[29] Tóth G,Fülöp L J,Vidács L,et a1.Complex event processing synergies with predictive analytics.Proceedings of the Fourth ACM Intemational Conference on Distributed Event-Based Systems.New York.Cambridge,United Kingdom, 2010. NY, USA:ACM.2010:95-96.

[30] Marz N.Storm:Distributed and fault-tolerant realtime computation [Z]. OSCON , 2012.

[31] Schultz-M O Ller N P,Migliavacca M, Pietzuch P.Distributed complex event processing with query rewriting[C].2009.