

主动式复杂事件处理方法的研究

耿少峰^{1,2}, 王永恒¹, 李仁发¹, 张佳²

(1. 湖南大学信息科学与工程学院, 湖南 长沙 410082; 2. 集美大学计算机工程学院, 福建 厦门 361021)

摘 要: 在 CPS 的应用背景下, 对传感器和控制设备产生的不确定性事件流进行分析和处理得出高层事件, 然后在此基础上引入适应性动态贝叶斯网络和并行马尔可夫决策过程模型来支持主动式的复杂事件处理。针对大型 CPS 中马尔可夫决策过程存在的状态数量巨大的问题, 引入状态划分和报酬分解的方法来进行并行优化。在模拟交通网络的环境中, 实验结果显示所提方法能有效地处理事件流, 并具有良好的可扩展性。

关键词: 信息物理融合系统; 大数据; 复杂事件处理; 马尔可夫决策过程

中图分类号: TP391

文献标识码: A

Research of proactive complex event processing method

GENG Shao-feng^{1,2}, WANG Yong-heng¹, LI Ren-fa¹, ZHANG Jia²

(1. College of Information Science and Engineering, Hunan University, Changsha 410082, China;

2. College of Computer Engineering, Jimei University, Xiamen 361021, China)

Abstract: Based on the preliminary analysis results of the indeterminate event stream that generated by the sensors and control purpose equipment of CPS, the adaptive dynamic Bayesian network and parallel Markov decision process model were used to support the proactive complex event processing. In order to resolve the vast state space issue of Markov decision process for large CPS, states partition and reward decomposition methods were proposed to parallel the decision making process. The experimental result based on the simulation of traffic network shows that proposed method can process event stream effectively and has favorable scalability.

Key words: CPS, big data, complex event processing, Markov decision process

1 引言

信息物理融合系统 (CPS, cyber physical system) 将智能计算、通信和控制技术深度融入到物理系统, 使计算资源与物理资源紧密结合与协调分配, 完成信息世界与物理世界的融合。它的应用领域非常广泛, 小到智能家庭网络, 大到工业控制系统, 覆盖了各种不同规模的应用系统。如何有效地处理 CPS 应用中所产生的大数据是人们目前需要面对的主要问题。当前在大数据研究领域, 对于历史数据的分析和处理已经进行了广泛深入的研究, 特别是在数据仓库和数据挖掘方面。常用方法之一是将大量的服务器组成集群或使用专业的云计算

平台 (如亚马逊的 AWS) 以及基于 Hadoop 的分布式计算框架。另外一种近期常见的计算平台是 Spark, 它突破了 MapReduce 的算法框架, 支持更灵活的数据处理方式, 性能也有了很大的提升。但这些方式并不适合于 CPS 的数据处理。因为 CPS 在工作时产生的是实时的数据流, 对这些数据流的处理往往有较高要求的时间约束。而 Hadoop/MapReduce 的计算框架仅适用于处理静态数据和历史数据。近期也出现了一些面向数据流的处理平台, 如 Spark streaming、Storm 和 Flink。这些平台能够高效地处理数据流, 但仅限于比较简单的过滤、聚合等操作, 无法直接支持面向 CPS 的复杂数据流处理。

收稿日期: 2016-04-07; 修回日期: 2016-06-26

基金项目: 国家自然科学基金资助项目 (No.61173036, No.61371116); 福建省教育厅基金资助项目 (No.JA14186)

Foundation Items: The National Natural Science Foundation of China(No.61173036, No.61371116), Foundation of Fujian Educational Committee (No.JA14186)

在 CPS 应用中,系统必须处理来自无线传感网、RFID 阅读器、GPS、社交媒体等各种各样设备的信号,这些信号可以被当成事件来处理。这些由设备直接生成的事件称为原始事件。一般情况下原始事件里的语义信息非常有限,在实际应用中,人们更加关注的是业务逻辑和规则等高级别信息。如在装有智能门禁的小区门口有车辆通行时,RFID 阅读器会进行读操作来生成原始的事件,但只有“车辆进入或离开小区”这样的复杂事件才是人们真正关心的,可以根据一些模式将许多原始事件结合在一起,从而得到对用户有意义的复杂事件。CPS 应用系统先将业务逻辑转换成复杂事件,再通过处理复杂事件来检测业务逻辑。复杂事件处理(CEP, complex event processing)^[1]用于处理数量巨大的简单事件,并从中得到有价值的高层次的信息。

在某些 CPS 的应用中,对象执行的操作可以改变系统的状态。当前主流的事件处理方法为响应式的,这意味着动作是由系统状态发生变化触发。主动式事件处理系统通过运用预测和自动决策技术^[2]减轻或避免将来不希望的事件发生。例如,在一个 CPS 的交通网络中,可以根据当前的状态和历史数据来预测道路的拥挤状态,然后采取一些行动以避免将来可能出现的拥堵。马尔可夫决策过程(MDP, Markov decision process)作为一个最佳决策过程中的随机动态系统,多年来人们已对其进行了深入的研究,并在多个领域中得到了广泛的应用。

目前,基于 MDP 实现主动式复杂事件处理还面临着巨大的挑战。首先是如何有效地处理由于误读、干扰等原因产生的事件的不确定性。当前的研究主要限于通过信息融合和推理来解决单个事件的不确定性,但缺乏对复杂事件不确定性的研究。另一个问题是如何进行高效准确的状态预测。尽管人们已经对数据流预测进行了大量的研究,但针对 CPS 产生的高速、海量数据流,预测模型和算法上都还面临挑战。同时,在决策支持方面,目前很少有文献涉及如何将 MDP 融入到主动式复杂事件处理中,以及如何应对在此过程中 MDP 所产生的巨大的状态空间。

针对上述问题,本文提出一种主动式复杂事件处理的架构和方法。该方法以交通网络为背景,将树的匹配和缓冲链表相结合,高效地处理了大量由车辆产生的不确定事件流,然后将处理后的复杂事件提交给适应性动态贝叶斯网络(ADBN, adaptive

dynamic Bayesian network)来预测未来事件和系统状态,最后主动式响应器根据 ADBN 的预测结果,利用一种基于状态划分和并发操作的新型并行 MDP 方法进行分析,最终产生避免交通拥堵的决策。

2 相关研究

CEP 通过监测不断产生的数据流,根据每个事件发生的集合/序列识别复杂事件,然后对检测到的状态做出回应。Etzion 等^[3]在其著作中定义了复杂事件处理的基本概念及体系结构。一个事件处理 agent (EPA, event processing agent) 执行某些处理逻辑,从输入事件中提取出复杂事件。多个 EPA 连接在一起可以执行更复杂的层次化事件处理,这种网络称为事件处理网络。

在以前的主动数据库研究中,人们已经对 CEP 做了大量的工作。常用的方法都是基于固定的数据结构,如树、图、自动机和 Petri 网。这些方法很难灵活地对事件查询语言的实现进行优化,也很难根据应用需求的变化来支持查询语言的扩展。针对这些问题,Eugene 等^[4]提出了一种基于查询规划的复合事件检测方法(SASE)。SASE 基于有限自动机(NFA)和堆栈来检测复杂事件,并对大的滑动窗口和大的中间结果集进行了优化,实现了比较好的性能和可扩展性。近年来的很多工作围绕着对 SASE 的改进展开,AGRAWAL 等^[5]提出了改进的有限状态机模型来提供更丰富的查询能力,Zhang 等^[6]对 SASE 进行了改进,提供了对非精确时标的事件流的支持。

由于传感器噪声等因素,CPS 事件具有不确定性,这种不确定性通常使用概率来表示。处理概率性原始数据的 CEP 普遍采用的模型有隐马尔可夫模型、动态贝叶斯网络和条件随机域等。近年来也出现了一些基于有限状态机的概率复杂事件处理方法。文献[7]基于 NFA 设计了一种新的数据结构——链接实例队列(chain instance queues),并采用一种条件概率索引树来提高搜索性能。另外,文献[8]面向 NFA 提出一种优化方法,不仅能够计算复杂事件的概率,还能够对不可信网络产生的数据获取复杂模式的可信度。基于有限状态机的概率复杂事件处理,比较有代表性的研究项目是美国华盛顿大学的 Cascadia 系统^[9]和 Laha 系统^[10]。Cascadia 系统基于概率数据库中的可能世界模型(possible world model),实现了对概率事件流和历史概率事件的查

询，但没有在预处理中考虑流上相邻事件间的时间关联性。Laha 成功支持了关联概率事件流上的高效查询，并支持更复杂的事件代数操作，从而能支持更灵活的复杂事件处理。

主动式系统的研究已经持续了多年，如主动式运输管理系统^[11]、面向即时无线网络的主动型路由^[12]等。这些研究总体上代表了一种由响应式系统向主动式系统转变的潮流。在复杂事件处理这个领域，对主动式方法的研究还比较少。但当前 CPS 的发展已经对这种研究提出了需求，也提供了研究的基础条件。Engel 等^[13,14]提出了一种基于事件的主动式计算框架，该框架主要由复杂事件处理网络、预测器和基于马尔可夫决策过程的主动式响应器构成。该工作对传统的复杂事件处理 agent 进行了扩展，支持 2 种新型的 agent：预测 agent 用于预测系统状态；主动式响应 agent 用于计算合适的响应并执行。但它的应用背景不是面向 CPS，也没有考虑大规模的数据和状态的问题。

多年来，人们对 MDP 进行了广泛的研究和应用，产生了多个变种。对于大型 CPS 应用这个领域，最大的挑战在于巨大的状态空间造成的组合爆炸问题。对这个问题当前主要有 2 个研究方向。第一

类是根据问题领域中的信息来简化结构，如状态聚集的方法^[15]和时间聚集的方法^[16]等。此类方法的问题和具体的系统相关，无法找到普遍解决方法。第二类是近似计算的方法，如近似动态规划^[17]和选择性近似^[18]等方法。这类方法的关键是如何在优化过程中对值函数进行近似。当应用到大型 CPS 的事件处理领域，MDP 的规模更大且具备一些新的特征，其具体模型和大规模计算问题的解决仍然是一个挑战。

3 系统架构

本文方法的总体结构如图 1 所示。CPS 中各类传感器及 RFID 等设备产生的信号，汇聚在一起称为原始事件流。由于设备本身的精度及干扰等原因，这些事件不是完全精确的，称为概率事件。这些原始事件经过概率复杂事件处理引擎的处理，提取出有意义的高层事件，也就是复杂事件。其中，PEPA (probabilistic event processing agent) 为概率事件处理器^[19]，它负责处理由底层设备产生的原始事件流，并最终生成概率复杂事件。查询规划器根据复杂事件查询语句使用一个或多个 PEPA 来完成查询。多个 PEPA 组合在一起构成的网络称为概率

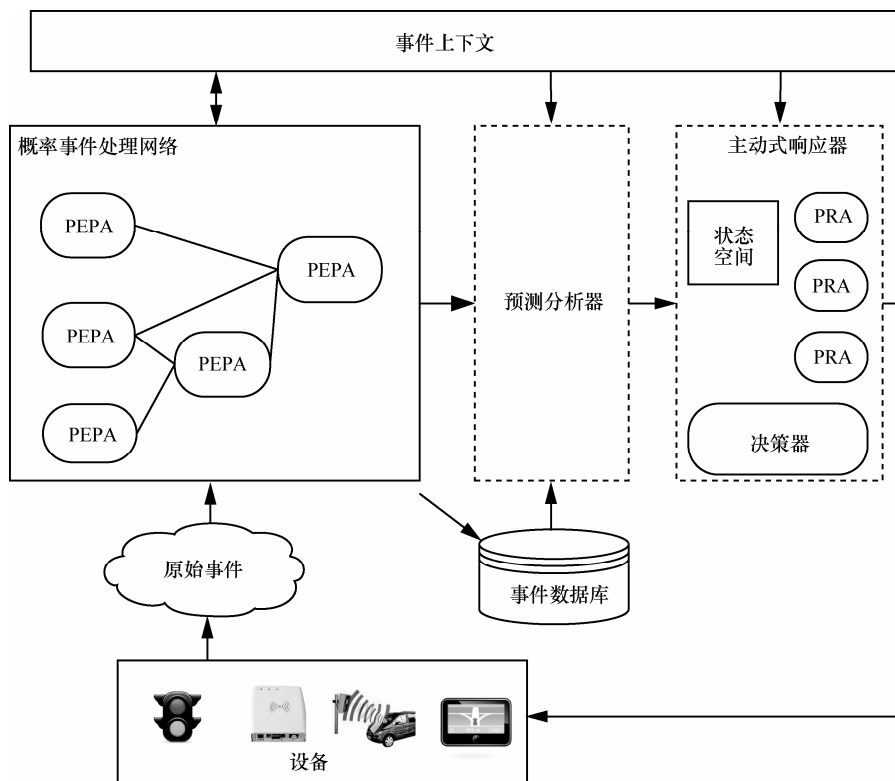


图 1 系统架构

事件处理网络 (PEPN, probabilistic event processing network)。本系统中预测分析器 (PA, predictive analytics) 的实现方法是基于 Pascale 等^[20]的工作, 通过使用一种适应性动态贝叶斯网络模型, 根据输入的复杂事件来预测道路网络中可能会发生拥堵的节点。复杂事件被存储在事件数据库中, PA 可以通过学习这些历史事件获得更准确的预测。主动式响应器根据 PA 预测出来的系统未来状态, 利用并行 MDP 产生避免或减弱这些状态发生的决策。PRA (proactive agent) 为执行器, 负责执行对应的决策动作。上下文可以从复杂事件中获取, 也可以从其他信息系统获取, 并影响系统的状态预测和控制。

定义 1 概率原始事件 CPS 事件流中的原始事件即在某个时间的一次发生 (一次信号读取)。概率原始事件表示为: $\langle ID, A, T, Pr \rangle$ 。其中, ID 为产生数据的设备标签号, A 为事件属性的集合, T 为事件发生的时标。Pr 为事件发生的概率值, 它代表由原始信号转换为确切的 CPS 事件的概率。

定义 2 概率复杂事件是由原始事件或者复杂事件按照一定的运算规则形成的事件。一个概率复杂事件表示为: $\langle E, R, Tc, Pr \rangle$ 。其中, E 是符合合成规则的概率原始事件串, R 为事件合成规则, Tc 为事件的时间跨度, Pr 为事件概率值。

事件类型由一组具有相同语义和结构的事件规则构成, 每个事件对象都是一个事件类型的实例。一个事件类型可以表示为原始或复杂事件。主要的复杂事件模式包括 ALL、ANY、COUNT 和 SEQ^[3]等。如在交通系统中, COUNT 模式用来计算规定时间内某一地点经过车辆的数量。把这些模式结合起来可以构建层次性的复杂模式。

4 主动式复杂事件处理

4.1 不确定性事件流的处理方法

由于 RFID 阅读器等设备存在漏读、误读和脏读等情况, 所以导致交给 PEPA 处理的数据存在不确定的可能性。这些不确定的原始数据被称为不确定事件。不确定事件流的复杂事件处理主要有 2 个挑战, 一是如何对大量高速的、实时的事件进行符合查询规则的检测; 二是如何计算由相关或不相关的不确定事件流组成的复杂事件的概率。本文提出了一种面向不确定性事件流的复杂事件处理方法 (ISCEP, indeterminate stream complex event processing) 来解决上述问题。该方法采用事件概率模型建

模, 并通过基于匹配树与各节点的缓冲链表相结合的方法来处理大量的事件流。

在大规模的 CPS 应用中经常需要处理分布式的事件流, 本文假设不同事件流上的事件实例是相互独立的。在单一事件流里, 一些在 SEQ 模式中的原始事件具有马尔可夫性。这意味着下一个事件发生的概率仅取决于序列中的当前事件, 和历史事件无关。如一辆汽车在 $i+1$ 时刻的位置和它在 i 时刻的位置有关, 而和 i 时刻之前的位置无关。一个具有马尔可夫性的事件序列被称为马尔可夫链。像 Kawashima 等^[9]做的工作那样, 使用条件概率表 (CPT) 来处理条件概率。除了这些马尔可夫链的事件, 可能还存在一些相互独立的原始事件。对于一个 SEQ 事件 $E=SEQ(e_1, e_2, \dots, e_m)$, 原始事件划分为 2 个集合 S 和 T 。S 包含相互独立的事件, T 包含一个或多个马尔可夫链。SEQ 事件的概率可以根据式(1)计算。

$$Pr(E) = \prod_{e_j \in T} Pr(e_j) \prod_{s_i \in S} (Pr(e_{i1}) \prod_{m=1}^{|s_i|-1} Pr(e_{m+1} | e_m)) \quad (1)$$

其中, S_i 是非独立集合 S 中的某一条马尔可夫链, e_1 表示马尔可夫链中的第一个事件, $Pr(e_{m+1}|e_m)$ 表示连续发生的事件是前后相关的, 它的值可以在 CPT 中获得。

事件概率模型中 PEPA 的输入是由概率原始事件组成的事件流。阅读器所处区域用大写字母表示, 时标用数字表示。如图 2 所示的阅读器在不同区域和时刻读取到由同一个设备产生的概率流。

SEQ	A ₀	A ₁	B ₁	B ₂	C ₃	D ₄	...
Pr	0.6	0.9	0.8	0.9	0.5	0.7	...

图 2 概率原始事件流

EPA 的输出是概率复杂事件。例如表 1 是经过处理后得到的某个车辆可能的行驶路径, 其中省略了复杂事件的合成规则和时间跨度, 保留了 2 个主要的属性: 符合合成规则的概率原始事件串和复杂事件的概率。根据 EPA 的输出, PA 可以选择更可靠的结果作为输入, 这样在一定程度上可以避免因数据错误而导致错误的状态预测。

表 1 概率复杂事件输出流

序列	概率
A ₁ B ₂ C ₃	0.83
A ₁ B ₃ C ₄	0.36
A ₂ B ₃ C ₄	0.12

不确定性事件流的具体处理流程如图 3 所示。

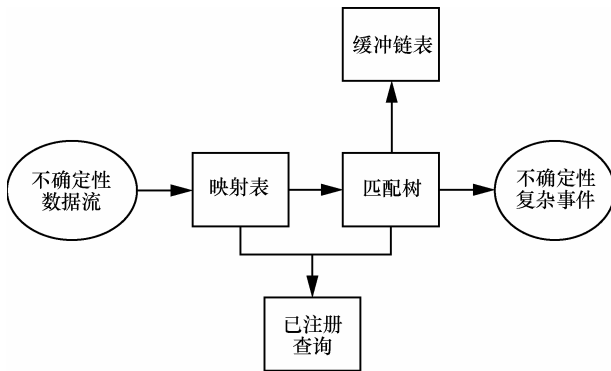


图 3 不确定性事件流处理方法

当查询被提交的时候，会注册相应的复杂事件。所谓注册就是在查询映射表中建立查询索引，同时建立用于流处理的匹配树。将交通网络划分为若干子区域，根据子区域和不同的时间段（如早晚高峰期）对应的已注册查询来构建查询映射表。当某一时段内某个区域发生事件的时候，就可以通过查询映射表迅速地找到已注册查询。如构造一个如表 2 所示的查询映射表。如果从输入流中读取到 8:00 区域 A 在发生事件，那么应该将原始事件发送到查询 1 对应的匹配树处理。

区域	时间	相关查询
	7:00~9:00	查询 1
A	17:00~19:00	查询 1、查询 2

B	7:00~9:00	查询 1

C	7:00~9:00	查询 1、查询 3

当处理实时数据流时，每个登记的查询都会建立一个匹配树。基本的建立规则在文献[21]描述。叶子节点是区域节点，中间节点是复合规则节点。在每一个节点的下方都会建立一个缓冲链表用以缓冲历史事件。如图 4 所示是查询 1 的匹配树构造结果，其中，SEQ 节点是由 B、C 区域事件节点组成，TSEQ 节点是由复合节点 SEQ 和 A 区域事件节点组成。

在执行查询时，首先获得概率简单事件流中的

每一个区域事件的相关查询集合。接着遍历每一个查询，把区域事件输送到查询对应匹配树中，按照自底向上的方式进行匹配。如果当前节点匹配成功就会在本节点的缓冲链表中加入匹配成功的记录，接着发送更新消息给父节点。父节点接到更新消息就会对所有子节点的缓冲链表进行匹配，若满足规则会在自己的缓冲链表中加入成功匹配的节点记录，然后继续向上级节点发送更新消息。如此迭代直到没有节点再接收到更新消息为止。最后对匹配树的根节点缓冲链表进行查看，如果有成功匹配记录，则自顶向下输出所有过程并返回结果集合，否则直接结束。具体过程如算法 1 所示。

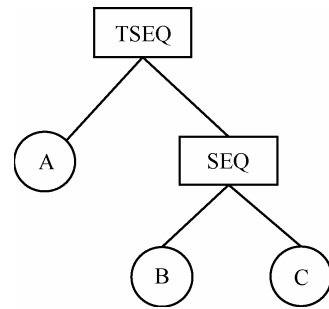


图 4 查询 1 的匹配树

算法 1 ISCEP 算法

输入 E 为概率原始事件流，QTree 为查询匹配树

输出 R 为概率复杂事件模式的实例集合
方法：

- 1) for 每个事件 E do
- 2) $query[i] \leftarrow MappingTable(E.A, E.id)$
- 3) for $query[i]$ 中每一个查询 do
- 4) $qtree \leftarrow InitQTree(query)$
- 5) for 每一个 QTree 的区域节点 do
- 6) if $E.A$ 在当前区域
- 7) then $Add(E)$;
- 8) $Update()$;
- 9) end for
- 10) while 节点 N 收到更新通知 do
- 11) if $Match(N, children)$
- 12) then $Add(N)$;
- 13) $Update()$;
- 14) end while
- 15) if QTree 根节点链表长度 > 0
- 16) then $R \leftarrow Output(QTree)$

```

17)     end for
18)   end for
19)   return R

```

算法中几个函数含义如下。

Add(): 事件匹配成功后, 将记录加入链表。记录包含新计算的复杂事件概率、产生此记录的事件集合和时标。

Update(): 节点缓冲链表有新记录加入后, 用此函数向节点的父节点发送更新消息。

Match(): 对子节点缓冲队列中的记录按照时标顺序由后往前进行遍历, 对时标限制和运算符限制进行匹配。

Output(QTree): 根节点缓冲链表有记录时候调用。因为成功匹配记录包含直接子节点的信息, 所以按照自顶向下的方式检索直到叶子节点, 就能得到组成当前查询结果的所有信息。

设已注册的查询有 N 个, 表示为 Q_1, \dots, Q_N , 划分的区域有 M 个, 表示为 $D=D_1, \dots, D_M$, 函数 $f(D_i)$ 表示区域 D_i 对应的查询的集合。设一个新的查询 Q 对应 K 个区域, 表示为 $D_Q=D_{Q_1}, \dots, D_{Q_K}$, 其中, $D_{Q_i} \in D$ 。在没有优化的情况下, Q 的查询代价表示为

$$T(Q) = \sum_{i=1}^K T(Q_{D_i}), \quad Q_{D_i} \text{ 表示对应到区域 } D_{Q_i} \text{ 上的子查询。}$$

如果某个区域 D_{Q_i} 上的子查询和已注册查询匹配, 则其代价可表示为 $\text{find}(f(D_{Q_i})) + \text{merge}(f(D_{Q_i}))$, 其中, $\text{merge}(f(D_{Q_i}))$ 为合并子查询结果的代价, 而由于建立了索引, $\text{find}(f(D_{Q_i}))$ 可忽略不计。设已注册查询匹配率为 α , 则查询 Q 的区域匹配数为 $\lfloor \alpha K \rfloor$, 对应的区域集合为 D'_Q 。则 Q 的查询代价变为 $\sum_{D \in D \setminus D'_Q} T(Q_{D_i}) + \text{merge}(\{f(D_{Q_i}) \mid D_{Q_i} \in D'_Q\})$ 。由于 merge 合并结果的代价远小于直接查询, 因此当匹配率 α 的值比较大的时候, 算法的查询性能会有大幅度的提高。

4.2 基于并行马尔可夫决策过程的决策方法

针对 CPS 复杂事件处理系统的特点, 传统马尔可夫决策过程扩展如下。

定义 3 主动式复杂事件系统并发马尔可夫决策过程。一个主动式复杂事件系统并发马尔可夫决策过程定义为 $\langle I, S, \vec{A}, P, R, E_c, C, T_c \rangle$ 。其中, I 为处理动作的 agent 有限集合, S 为状态的有限集合, 包含一个特殊的初始状态 S_0 。 $\vec{A} = \times_{i \in I} A_i$ 为动作联合, 其中, A_i 为来自第 i 个 agent 的动作。 $P: S \times A \times S \rightarrow [0, 1]$

为状态转换函数, $P(s, \alpha, s')$ 代表在状态 s 执行响应 α 后转换到状态 s' 的概率, $R: S \rightarrow \text{Re}$ 为报酬函数 (Re 为实数)。对每个 $(s, \alpha) \in S \times A$, $\sum_{s' \in S} p(s, \alpha, s') = 1$ 。 E_c 为 CPS 的复杂事件, 复杂事件的变化影响系统的状态。 C 为上下文, 在不同的上下文中, agent 会采用不同的动作。 $T_c: S \times E_c \times S \rightarrow [0, 1]$ 为复杂事件对系统状态的影响。

系统从原始状态开始, 根据合适的策略选择一个或多个动作, 由动作 agent 执行。动作 agent 的操作引起 CPS 原始事件的变化, 原始事件的变化引起复杂事件的变化, 根据对复杂事件的综合分析, 确定系统转移到下一个状态的概率。马尔可夫决策过程的关键是寻找一个策略 $\pi: S \rightarrow A$, 它把每个状态映射到一个动作集合。最优的策略使报酬的总和达到最大。策略选择步骤基于式(2)。

$$v_\pi(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s' | s, \pi(s)) v_\pi(s') \quad (2)$$

策略更新步骤基于式(3)。

$$\pi^*(s) = \arg \max_{a \in A} (r(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) v_\pi(s')) \quad (3)$$

其中, $\gamma < 1$ 为衰减因子。经典的 MDP 中动作是顺序执行的, 最近有些研究提到并发 MDP 按功能对不同的动作进行分组, 这些分组后的动作并发执行且相互没有影响。MDP 模型针对一个状态有多个并发执行的动作, 它们在不同的位置执行相同的功能来协同把系统转向期望的状态。本文中的 MDP 方法基于以下观察结果。

1) 观察结果 1。大型 CPS 中的对象可以根据上下文进行划分, 每个组对应一个子状态, 而不单独考虑每个对象的状态。

例如在考虑交通拥堵问题时, 每个路口的车流量 (或拥堵情况) 是子状态, 所有路口的拥堵状况构成系统的总体状态。

2) 观察结果 2。大型 CPS 中某个位置对应的子状态依赖于其邻居位置的状态 (此处的邻居是指在一定范围内和当前节点相连的其他节点)。

例如某个路口的拥堵状态依赖于前一个时间段其邻居路口的拥堵状态。

3) 观察结果 3。在大型 CPS 中, 可以把状态节点分为一般节点和重点节点。重点节点是在主动式处理中要控制其状态的节点。如图 5 所示, 可以根

据重点节点对交通网络进行划分。对单个子网中重点节点的状态转换，只需调整其子网内邻居节点的状态来解决。

交通网络可以如图 5 划分成若干组。一个或多个拥堵程度高（由 PA 预测）的节点作为一个组的中心，所有到中心的距离小于阈值 D 的节点划分为一个组。假设有 n 个节点： $J=\{j_1, j_2, \dots, j_n\}$ ，本文把交通流量按拥堵程度分成 k 个级别： $CS=\{c_1, c_2, \dots, c_k\}$ 。MDP 中的状态可以表示为 $s_t = \langle s_{t1}, s_{t2}, \dots, s_{tm} \rangle$ ，其中， $s_{it} \in S$ 是节点 j_i 在 t 时刻处于拥堵状态， $s_i \in CS$ 。策略 $\pi_{it} = \{a_{it1}, a_{it2}, \dots, a_{itm}\}$ ，其中，动作 a_{itk} 的作用是在 t 时刻引导一些车辆改变它们的路径（如可以通过导航软件提示等方法），从而减少 k 组的拥堵程度。本文假设在交通网络中分组时，任意 2 组中心间的距离足够大，这意味着子动作间不存在依赖。因此上述分组的数据可以用分布式服务器处理，通过并行来提高系统性能。通过网络划分，把一个状态数为 n 的 MDP 转换成多个子 MDP，每个子 MDP 的联合状态数远小于 n ，从而大大降低了问题规模。

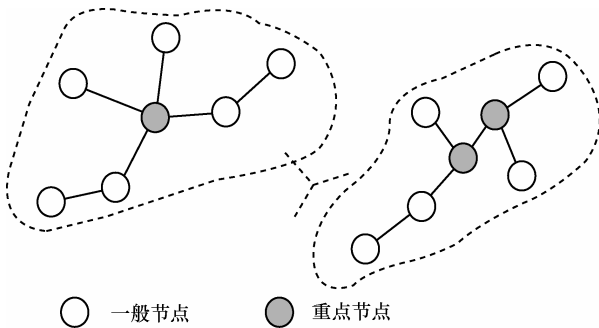


图 5 基于重点节点的网络划分

算法 2 并行调度方案

输入 已划分的交通网络分组 $V=\{v_1, v_2, \dots, v_k\}$ ，

分布式系统中服务器节点 $C=\{c_1, c_2, \dots, c_n\}$

输出 网络分组在处理节点中的分配方案

方法：

- 1) 计算 $v_{k,s}, \forall v_k \in V$;
- 2) 根据 $v_{k,s}$ 对 V 排序;
- 3) 设置 MaxHeap;
- 4) for $i \leftarrow 1$ to $|\text{MaxHeap}|$ do
- 5) Pop(v_i);
- 6) $c_{\min} \leftarrow \text{Min}(C.cost)$;
- 7) Assign(c_{\min}, v_i);
- 8) $c_{\min}.cost \leftarrow c_{\min}.cost + S_{it}(v_i)$

9) end for

其中，分布式系统为同构体系结构，并且服务器节点数目 n 小于子网分组数 k ， $v_{k,s}$ 是分组 v_k 中所有节点拥堵程度总和。根据 $v_{k,s}$ 的值对 V 进行排序，放入堆 MaxHeap 中。遍历 MaxHeap，通过 Assign(c_n, v_k) 将当前分组任务分配给 $cost$ 最小的服务器，随后对服务器的 $cost$ 值进行修正。此算法实现了分布式服务器的负载均衡，相对于单节点，处理性能会有大幅度的提高。

根据观察结果，在前述 MDP 模型中加入新的变量 G ，代表子状态节点的网络结构。 $G=(V,E)$ ，其中， V 为节点的集合， E 为边的集合。一个边 (i,j) 存在表示节点 i 影响节点 j 的状态。

定义 4 邻居状态节点。对于任意状态节点 $i \in V$ ，其邻居状态节点 $N(i)=\{j \in V | d_{ij} \leq k\}$ ，其中， d_{ij} 为 i 和 j 之间的路径长度， k 是阈值。定义 4 表明一个节点的状态只受邻居节点状态的影响。

通过定义邻居节点，和重点节点相关的联合状态数进一步缩小，从而进一步降低问题的规模。

定义 5 状态转换分解。设有 n 个状态的子节点，对应的动作也分为 n 个子动作，则根据前面的观察结果，有

$$p(s' | s, a) = \prod_{i=1}^m P_i(s'_i | s_i, a_i) \quad (4)$$

其中， s 表示当前状态， s' 表示下一个状态。

定义 6 报酬分解。设有 n 个状态的子节点，对应的动作也分为 n 个子动作，则根据前面的假设，有

$$r(s, a) = \sum_{i=1}^n r_i(s_i, a_i) = \sum_{i=1}^n \sum_{k=1}^{|N(i)|} r_{i,k}(s_k, a_k) \quad (5)$$

定义 6 意味着整体报酬被分解为若干个组的报酬。每一组的报酬又进一步分解为所有邻居节点的报酬。这样每个节点的报酬可以采用单机多处理器的并行方式来计算。

在交通 CPS 中，当一个动作 a_i 被执行，节点 i 处的交通流量为

$$\Delta_i = S_i - \sum_{j_k \in Pa_out(j_i)} S_k \alpha_{ki} \beta_{ki} p(j_k) - \sum_{j_m \in Pa_in(j_i)} S_m \alpha_{im} \beta_{im} p(j_m) \quad (6)$$

其中， $Pa_out(j_i)$ 是尝试减少节点 i 处流量的节点集， $Pa_in(j_i)$ 是尝试增加节点 i 处流量的节点集， α_{ki} 是 S_k 流向节点 i 的比例。假设并不是所有车辆都会按

本文的策略来改变行驶路线，那么 β_{ki} 是 S_k 受动作 a_i 影响的比例， $P(j_k)$ 是在节点 k 处的车辆根据 a_i 改变路线的概率，它可以从历史数据得到。整体的状态事务概率可以用 $P(j)$ 和 Δ_i 计算。

根据定义 6，可以定义平均流量和其中一组 g_i 的方差为

$$\bar{g}_i = \frac{\sum_{j=1}^{|g_i|} s_{ij}}{|g_i|} \quad (7)$$

$$D(g_i) = \left[\frac{\sum_{j=1}^{|g_i|} (s_{ij} - \bar{g}_i)^2}{|g_i|} \right]^{\frac{1}{2}} \quad (8)$$

本文的目标是把所有拥堵状态最小化。因此，定义欧式距离子动作的报酬函数和整体报酬函数为

$$R(S, a, S') = \sum_{i=1}^{|g|} \left[(\bar{g}_i - \bar{g}'_i)^2 + (D(g_i) - D(g'_i))^2 \right]^{\frac{1}{2}} \quad (9)$$

最终，最优策略可以按式(10)计算。

$$\pi^*(St) = \arg \max_a \sum_{S'} P(s, a, s') V(s') \quad (10)$$

5 实验结果与分析

SUMO 是一个开源的微观道路交通仿真模拟器^[22]，它可以检测通过相应区域车辆的数量和每辆车的位置。如图 6 所示，本文用 JADE(Java agent development framework)框架对 SUMO 进行扩展来支持多 agent 系统，同时使用 SUMO 的 TraCI 接口来获得感应圈变量和车辆位置的变量，用它们来模拟 RFID 和 GPS 阅读器。基于这个框架，agent 控制器用智能算法控制车辆在 SUMO 中移动，从中获取事件数据和执行相应的动作。在实验中，在地图里设置了 80 个路口和 8 万辆汽车。为了更接近真实的交通系统，定义了一组规则：每辆车都有一个家的位置和办公室位置，车辆 V_i 在家和办公室之间运行的概率是 p_i 。这些车辆还有相应概率去超市、医院等其他地点。此实验模型可以满足 PCEP 对交通网络的划分、状态和动作的分解等操作，并且可以并行的生成决策方法。本文把 4 台至强 E3 处理器和 16 GB 内存的服务器当作数据处理服务器。另一台有 4 GB 内存的 PC 运行 SUMO。

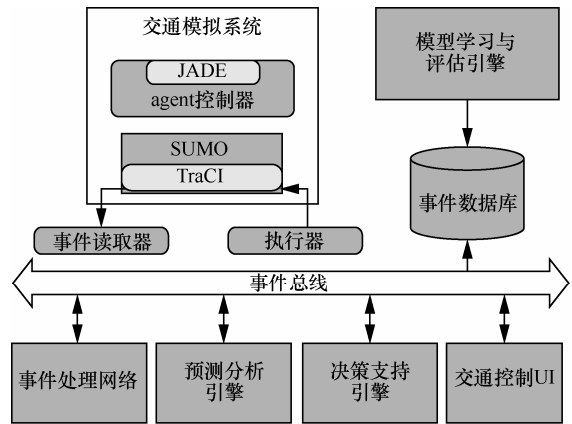


图 6 系统实现框架

首先来评估 ISCEP 关于多查询的处理性能。本文对经典的 RCEDA 方法^[23]进行了类似本文提出的概率计算方式的拓展，称之为 PRCEDA (probability RCEDA)。通过 PRCEDA 与本文提出的 ISCEP 进行比较。假定所有的注册查询都是二叉树且 Query Depth 不超过 4 层，从图 7 中可以观察到，随着注册查询数目的加大，ISCEP 的时间消耗增长幅度相比 PRCEDA 要小很多。因为 PRCEDA 以轮询方式查询匹配树，随着注册查询数目的增加，查询效率会越来越低。使用查询映射表的 ISCEP 轮询次数要少很多，所以查询效率下降较少。

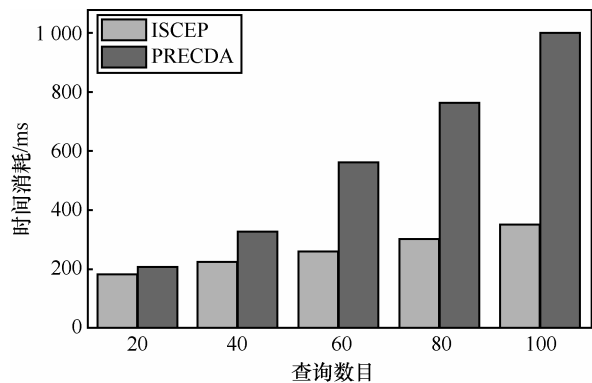


图 7 查询处理性能

下一个实验评估该系统的平均拥堵度，结果如图 8 所示。本文把拥堵程度划分为 11 个等级，其中，“0”表示无拥堵，“10”表示拥堵程度最高。从图中可以观察到使用本文方法比让汽车自主行驶时平均拥堵度明显降低。原因是本文方法发现节点可能出现拥堵时，会主动引导后续车辆向轻负荷节点行驶。同时，当车辆总体拥堵程度比较高的时候，本文方法带来的拥堵下降也更多。原因是当总

体车流量比较大时，系统执行的动作会影响更多的车辆，带来更多的拥堵下降。

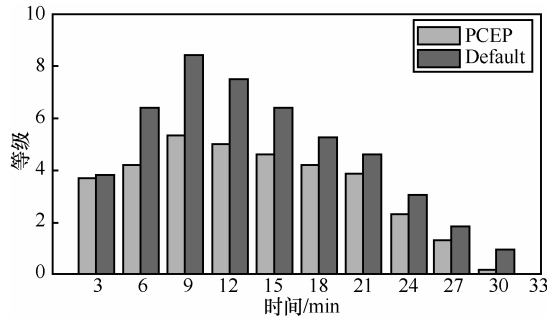


图8 拥堵程度比较

最后的实验中采用不同数量的服务器和不同大小的数据来测试本文方法的性能。重点节点的数量固定在20，其结果如图9所示。结果表明，本文方法通过网络的划分、邻居节点定义及状态和动作分解，可以在较大规模的MDP问题中在可接受的时间内形成决策。从图中可以看到，生成决策的平均消耗时间随着车辆数量的增加成正比。原因在于车辆数量的增加需要执行更多的子动作，因此增加了算法的复杂性。通过本文方法还可以发现随着车辆数量的增加，4个服务器的性能更好，而且生成决策所需的时间增长相对较慢。这说明系统具备更好的可扩展性，在一定范围内可以通过增加新的计算资源来支持更大规模的交通网络。

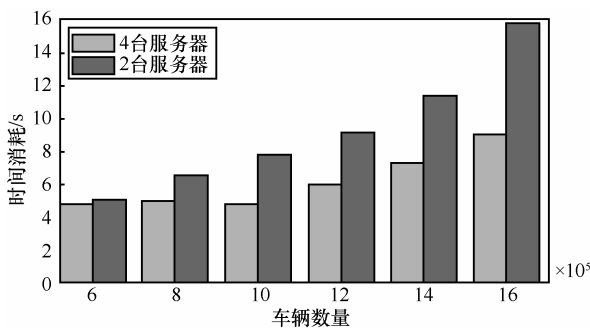


图9 不同服务器数量的性能

6 结束语

本文提出了一种对CPS设备产生的不确定性事件流进行处理分析，然后在基于ADBN的预测结果上引入并行MDP模型来控制大规模交通拥堵的主动式复杂事件处理方法。实验评估表明，该方法能够有效地减少节点的拥堵程度，并且具有良好的可扩展性。但在并行MDP中，子状态和子响应的

空间随着节点和车辆的增加依然会变得巨大。今后的工作重点在于研究如何进一步减少子状态的空间数量，从而提高系统的整体性能。

参考文献：

- [1] LUCKHAM D C. The power of events: an introduction to complex event processing in distributed enterprise systems[M]. Boston, Addison Wesley, 2002.
- [2] ENGEL Y, ETZION O. Towards proactive event-driven computing[C]//The Fifth ACM International Conference on Distributed Event-Based Systems. 2011: 125-136.
- [3] ETZION O, NIBLETT P. Event processing in action[M]. Manning Publications, 2010.
- [4] WU E, DIAO Y, RIZVI S. High-performance complex event processing over streams[C]//2006 ACM SIGMOD International Conference on Management of Data. 2006: 27-29.
- [5] AGRAWAL J, DIAO Y, GYLLSTROM D. Efficient pattern matching over event streams[C]//SIGMOD Conference. 2008: 147-160.
- [6] ZHANG H, DIAO Y, IMMERMANN N. Recognizing patterns in streams with imprecise timestamps[J]. PVLDB, 2010, 3(1): 244-255.
- [7] XU C, LIN S, LEI W. Complex event detection in probabilistic stream[C]//The 12th International Asia-Pacific Web Conference. 2010: 361-363.
- [8] KAWASHIMA H, KITAGAWA H, LI X. Complex event processing over uncertain data streams[C]//The Fifth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. 2010: 521-526.
- [9] KAWASHIMA H, KITAGAWA H, LI X. Complex event processing over uncertain data streams[C]//3PGCIC. 2010: 521-526.
- [10] WELBOURNE E, KHOUSSAINOVA N, LETCHNER J, et al. Cascadia: a system for specifying, detecting, and managing RFID events[C]//The 6th International Conference on Mobile Systems, Applications, and Services. 2008: 281-294.
- [11] DOLEV S, KOPEETSKY M, SHAMIR A. RFID authentication efficient proactive information security within computational security[J]. Theory of Computing Systems, 2011, 48(1):132-149.
- [12] KUNZ T, ALHALIMI R. Energy-efficient proactive routing in MANET: energy metrics accuracy[J]. Ad Hoc Networks, 2010, 8(7): 755-766.
- [13] ENGEL Y, ETZION O, FELDMAN Z. A basic model for proactive event-driven computing[C]//The 6th ACM International Conference on Distributed Event-Based Systems (DEBS'12). 2012: 107-118.
- [14] ENGEL Y, ETZION O. Towards proactive event-driven computing[C]//The Fifth ACM International Conference on Distributed Event-Based Systems, DEBS. 2011: 125-136.
- [15] JIA Q S. On state aggregation to approximate complex value functions in large-scale Markov decision processes[J]. IEEE Transactions on Automatic Control, 2011, 56(2):333-344.
- [16] SUN T, ZHAO Q C, LUH P B. Incremental value iteration for time

aggregated Markov decision processes[J]. IEEE Transactions on Automatic Control, 2007, 52(11):2177-2182.

- [17] POWELL W B. Approximate dynamic programming: solving the curse of dimensionality[M]. New York, Wiley-Inter Science, 2007.
- [18] JIA Q S, ZHAO Q C. Strategy optimization for controlled Markov process with descriptive complexity constraint[J]. Science in China, 2009, 52(11):1993-2005.
- [19] WANG Y H, CAO K N, ZHANG X M. Complex event processing over distributed probabilistic event streams computers and mathematics with applications[J]. Computers & Mathematics with Applications, 2012, 66(10): 1808-1821.
- [20] PASCALE A, NICOLI M. Adaptive Bayesian network for traffic flow prediction[C]//Statistical Signal Processing Workshop (SSP). IEEE, 2011: 177-180.
- [21] 王永恒, 杨圣洪. 分布式 RFID 数据流的复合事件检测方法[J]. 计算机应用研究, 2011, 28(11): 4177-4179.
WANG Y H, YANG S H. Complex event detection method for distributed RFID systems[J]. Application Research of Computer, 2011, 28(11): 4177-4179.
- [22] BEHRISCH M, BIEKER L, ERDMANN J, et al. Sumo - simulation of urban mobility: an overview[C]//The Third International Conference on Advances in System Simulation. Barcelona, Spain, 2011: 63-68.
- [23] WANG F, LIU S, LIU P, et al. Bridging physical and virtual worlds: complex event processing for RFID data streams[C]//The 10th International Conference on Extending Database Technology (EDBT2006). 2006: 588-607.

作者简介:



耿少峰 (1981-), 男, 河南台前人, 湖南大学博士生, 主要研究方向为 CPS、复杂事件处理等。



王永恒 (1973-), 男, 河北霸州人, 湖南大学副教授, 主要研究方向为物联网、数据挖掘等。



李仁发 (1957-), 男, 湖南宜章人, 湖南大学教授、博士生导师, 主要研究方向为 CPS、嵌入式系统和无线传感网等。

张佳 (1980-), 女, 河南新乡人, 集美大学讲师, 主要研究方向为物联网、嵌入式系统等。