

Дипломная работа

28 мая 2013 г.

1 Введение

Today's scientific experiments typically involve running and refining a series of intertwined computational analysis and visualization tasks on large amounts of data. The complexity of these so-called analysis pipelines resulting in high costs for development and maintenance, the need for sharing knowledge encoded in these pipelines as well as hardware to execute them, and the need for re

За последних два десятилетия в научном сообществе компьютерное моделирование, названное eScience, стало незаменимой частью исследовательского процесса наравне с традиционными инструментами, такими как эмперические, основанное на экспериментальных наблюдениях, теоретическое моделирование. Компьютерное моделирование включает в себя

1.1 Мотивация

Мотивация

2 Базовые понятия и определения

3 Формальное описание модели актор-ориентированных исследовательских workflow

В данном разделе будут введены базовые понятия, используемые в данной работе, в том числе, понятия атомарного блока, составного блока, потока данных, автомата Мили соответствующего блоку и так далее.

3.1 Графы актор-ориентированные иерархических workflow (Actor-Oriented Hierarchical Workflow Graphs)

Граф связей workflow Граф связей workflow $W = (A, D)$ состоит из набора акторов A , представляющих задачи и набора связей, соединяющих акторы через порты. Акторы по своей сути являются пассивными, т.е. на набор входных данных выдают. Акторы передают друг другу токены данных через сообщающиеся порты.

Порты Для каждого актора $a \in A$ существует собственный набор портов $port(a)$, таких что каждый порт $p \in ports(a)$ является либо входящим, либо выходящим для этого актора, т.е. $ports(a) = in(a) \cup out(a)$

Связи Пусть для workflow $in(W) = \bigcup_{a \in A} in(a)$ будет набор всех входных портов, $out(W) = \bigcup_{a \in A} out(a)$ набором выходных портов соответственно. Тогда связь $d \in D$ будет направленным ребром $d = \langle o, i \rangle$

Обобщение и дробление workflow Обобщение и дробление workflow являются ключевыми примитивами в составлении workflow. Для обобщения workflow W следовало бы представить его в виде *составного актора* a_W , иными словами "*свернуть*". И наоборот, при дроблении, "*раскрывая*" актор, получить

3.2 Описание актора

Базовой логической единицей потока данных является атомарный актор. Т.е. тот, который задаётся только Недетерминированным конечным автоматом(FSM).

Состояние любой системы или подсистемы в любой определённый момент времени характеризует, как система(подсистема) будет реагировать на входные данные. Формально, под состоянием мы будем подразумевать всю накопленную системой информацию, которая влияет на поведение системы в текущий момент и в последующие.

Для примера рассмотрим простой актор типа Counter, который считает количество собственных запусков. У него разумеется есть состояние, и его реакция на входные данные из порта trigger зависит от того, сколько раз он уже сработал. Для подсчёта и зрания числа запусков он использует локальную переменную. И эту локальную переменную будем называть **переменной состояния**.

В этом случае количество возможных состояний определяется типом счётчика. Если, к примеру, это `int`, то возможно только 2^{32} состояний, если `double`, то 2^{64} . Если же тип счётчика `String`, то число состояний бесконечно. Но не смотря на то что число состояний может быть очень большим, логика переходов между ними достаточно проста.

Finite State Machines Автоматом называется система, выходы которой зависят не только от поступивших входов, то и от текущего состояния системы. Состояние системы - объединение всей информации о предшествующих входах, необходимое системе для выполнения шага. Состояние системы может быть обозначено переменной состояния $s \in \Sigma$, где Σ - это набор всех возможных состояний системы. Конечным автоматом называется автомат, для которого число состояний Σ конечно.

Внутренняя логика блока представляется моделью недетерминированного конечного автомата (Finite State Machine, FSM) Определение: конечным автоматом называется набор $M = (\Sigma, I, \Lambda, T, s_0)$, где

- S -набор конечных состояния,
- I – набор входных портов актора,

- O - набор возможных выходных портов актора
- $s_0 \in \Sigma$ - начальное состояние,
- $T : S \times 2^I \rightarrow 2^{S \times 2^O}$ отображение сопоставляющее каждому состоянию с наборов входных портов набор состояний с соответствующим набором выходных портов

Как уже было определено выше, у актора имеется **начальное состояние** s , т.е. то в котором модель находится перед запуском. Так же могут быть и **конечные состояния**. Визуально состояния соединены переходами, рядом с которыми указано, что происходит при срабатывании работе.

Классическая теория конечных автоматов (Hopcroft and Ullman, 1979) различает два вида автоматов: **Автомат Мили** и **Автомат Мура**. В Автомате Мили, от входных значений, в отличии от Автомата Мура, выходное значение сигнала в котором зависит лишь от текущего состояния данного автомата.

4 The Need for Composing Models of Computation in E-science

5 Workflows and Hierarchy

Описание понятия актора(actor,atomic actor. ,composite actor (a.k.a. sub-workflow))

6 (Модели управления потоком)Models of Computation

Process Networks (PN)

6.1 Использование PN

- Тупики
-

Более подробно модель описывается через Kahn networks. Каждый актор запускается в отдельном потоке (thread) (или вычислительном узле), и все акторы запускаются в конкурирующем режиме.

SDF

6.2 Задачи

6.3 Использование SDF

Deadlocks

- Consistency of data rates
- The value of the iterations parameter
- The granularity of execution Более подробно в секции 3.3.

6.4 Тупики (Deadlocks)

6.5 Consistency of data rates

Все акторы внутри *однородного (homogeneous)* SDF принимают только по одному токenu из каждого входного порта и выписывают по 1 токenu в каждый выходной порт. В каждом цикле должно быть "сдерживающий" актор, так что dataflow-граф будет ациклическим, если убрать все "сдерживающие" акторы. Схему запуска можно определить статически, например, через топологическую сортировку графа.

6.6 Свойства SDF

SDF не связан с временными событиями. Для всех акторов, входящих в SDF, поглощение токенов из входных портов, выполнение вычислений операций и отправление токенов в выходные порты является атомарной операцией. Запуск композитного актора соответствует одной итерации содержащейся в нём модели по предварительно вычисленной схеме (schedule). При этой схеме выполнения рассчитывается так, чтобы, и при бесконечном числе итераций в модели не возникали тупики и накопление токенов.

6.6.1 Вычисление схем запуска

Вложенные друг в друга SDF модели , можно уflatten, Запуск последовательный, преимущественно на одном узле.

FSM

Конечные автоматы (Finite State Machines)

DDF

В этой модели управления вместо того, чтобы выдавать каждому актору отдельный тред, система управления запускает отдельный актор, когда необходимые удовлетворены его зависимости на входе. Dataflow делятся на два типа, dynamic dataflow (DDF) и synchronous dataflow (SDF). В случае DDF, система управления динамически определяет какой актор необходимо запустить в следующем, и следовательно составляет схему запуска(firing schedule) динамически во время работы. В случае SDF, система управления

- Сети Петри
- Анализ Сетей Петри
- Граф Карпа и Миллера
- Marked Graphs
- Однородные dataflow
- Обобщённые dataflow
- Модели Кана (Khan) для параллельных вычислений

9.0 Специальная лексика Dataflow - поток данных

10.0 Используемая литература

7 СЕТИ

7.1 Сети Петри

Сети Петри широко используются для моделирования и исследования динамических дискретных систем. И прежде чем рассмотреть частные случаи использования Сетей Петри, приведём описание их каноничной формы, согласно определению Петерсона [Pet81].

Сеть Петри представляет собой двудольный ориентированный граф, состоящий из вершин двух типов — *позиций* и *переходов*, соединённых между собой дугами. Вершины одного типа не могут быть соединены непосредственно. В позициях могут размещаться метки (маркеры), способные перемещаться по сети.

Простой сетью Петри называется набор $N = (S, T, F)$, где

1. $S = \{s_1, \dots, s_n\}$ - множество *позиций*
2. $T = \{t_1, \dots, t_r\}$ - множество *переходов* таких, что $S \cap T = \emptyset$.
3. $F \subseteq \mu S \times T \times \mu S$ - отношение *инцидентности* такое, что
 - $\forall \langle Q'_1, t_1, Q''_1 \rangle, \langle Q'_2, t_1, Q''_2 \rangle \in F : \langle Q'_1, t_1, Q''_1 \rangle \neq \langle Q'_2, t_2, Q''_2 \rangle \Rightarrow t_1 \neq t_2$;
 - $\{t | \langle Q'_1, t_1, Q''_1 \rangle \in F\} = T$

Условия в пункте 3 говорят, что для каждого перехода $t \in T$ существует единственный элемент $\langle Q', t, Q'' \rangle$, задающий для него входное мультимножество Q' и выходное мультимножество Q'' . Дадим определение входному и выходному мультимножеству.

Определение: *Входное и выходное мультимножества мест и переходов*

Пусть задана сеть $N = (S, T, F)$.

1. Если для некоторого перехода t имеем $\langle Q', t, Q'' \rangle \in F$, то будем обозначать
 - $t = Q' = \langle (s, n) | (t, n) \in s \bullet \rangle, t \bullet = Q'' = \langle (s, n) | (t, n) \in \bullet s \rangle$
2. И соответственно
 - $s = \langle (t, n) | (s, n) \in t \bullet \rangle, s \bullet = \langle (t, n) | (s, n) \in \bullet t \rangle$

Будем говорить, что $\bullet t$ - входные, а $t\bullet$ - выходные позиции для перехода t . Таким образом, согласно определению, справедливо:
 $\forall t \in T : \langle \bullet t, t, t\bullet \rangle \in F$.

Позиция s называется инцидентной переходу t , если $s \in \bullet t$ или $s \in t\bullet$.

Сети Петри имеют удобную графическую форму представления в виде графа, в котором места изображаются кружками, а переходы прямоугольниками. Места и переходы, причем место s соединяется с переходом t если $(s, n) \in \bullet t$ и t соединяется с s если $(s, n) \in t\bullet$ для некоторого натурального числа $n \in N$. Здесь число n называется кратностью дуги, которое графически изображается рядом с дугой. Дуги, имеющие единичную кратность, будут обозначаться без приписывания единицы.

Само по себе понятие сети имеет статическую природу. Для задания динамических характеристик используется понятие маркировки сети $M \in \mu S$, т.е. функции $M : S \rightarrow N_0$, сопоставляющей каждому месту целое число. Графически маркировка изображается в виде точек, называемых метками (tokens), и располагающихся в кружках, соответствующих местам сети. Отсутствие меток в некотором месте говорит о нулевой маркировке этого места.

Определение: *Маркированная сеть Петри* Маркированной сетью Петри называется набо $\Sigma = (S, T, FM_0)$, где

1. (S, T, F) - сеть;
2. $M_0 \in \mu S$ - начальная маркировка.

Работа маркированной сети Петри управляется наличием или отсутствием маркировочных токенов. Сеть Петри срабатывает переход t , в процессе которого с каждой мультипликативного входа $(s, n) \in I(t)$ снимается по n токенов с каждого, где n - соответствующая кратность дуги и каждому мультипликативного выходу $(s, m) \in O(t)$ прибавляется m токенов. Для любого положения количество токенов не может быть отрицательным, поэтому переход не может сработать, если количество токенов на входах меньше требуемого количества токенов на выходе.

Определение: *Правило срабатывания переходов* Пусть $\Sigma = (S, T, FM_0)$ маркировочная сеть.

1. Переход $t \in T$ считается возбуждённым при маркировке $M \in \mu S$ если $M \geq I(t)$;

2. переход t , возбуждённый при маркировке M , может сработать, приводя к новой маркировке M' , которая вычисляется по правилу: $M' = M - I(t) + O(t)$. Срабатывание перехода обозначается

ПРИМЕР

7.2 Анализ сетей Петри

Сети петри могут быть использованы для моделирования конкурирующих систем. К примеру, сеть процессов с общей памятью.

Композиционный подход к построению сетей Петри предполагает возможность построения более сложных сетей из менее сложных составляющих. Для этого вводятся точки доступа, которые позволяют объединять простые сети путём синхронизации событий и состояний (переходов и мест).

Обычно в сетях Петри считается, что если при одной и той же маркировке возбуждено несколько переходов, то может сработать любой, но только один из них. Это ограничение не является принципиальным и может быть снято. При применении сетей Петри для целей управления позициям сопоставляются операции (действия), а переходам — условия, при выполнении которых возбужденные переходы срабатывают, активизируя соответствующие операции. При этом попадание меток в позицию ассоциируется с началом операции, а удаление метки — с ее окончанием. При использовании такого предположения считают, что любая операция не может быть повторно начата до ее завершения. Для описания таких процессов могут применяться только безопасные сети петри, т. е. такие сети, в которых при любой начальной маркировке μ невозможно ни через какую последовательность выполненных переходов получить такую маркировку μ' с количеством токенов в положении больше единицы.

Возможно сделать сеть Петри быть безопасной добавляя дуги, обеспечивая Безопасность

Список литературы

- [1] [GG] Gray L., Griffeth D. The ergodic theory of traffic jams // J. Stat.

