



OSTIS-2013

(Open Semantic Technologies for Intelligent Systems)

УДК 004.272.43+004.272.32

ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ВЕКТОРНОГО ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ПРОЦЕССОРА

Вереник Н.Л., Татур М.М.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

nick.verenik@gmail.com

tatur@i-proc.com

В статье изложен подход к проектированию средств аппаратной поддержки вычислений при построении интеллектуальных систем на базе семантических сетей. В качестве научной основы принята концепция платформенной независимости верхнего логико-семантического уровня системы от ее аппаратной реализации. Рассмотрены основные этапы проектирования: разработка архитектуры векторного проблемно-ориентированного процессора, разработка программной модели и обоснование выбора элементной базы.

Ключевые слова: семантическая сеть; параллельный процессор; семантическая обработка информации.

ВВЕДЕНИЕ

В данной работе под семантической сетью понимается графовая структура, в которой определенным образом закодирована полезная информация. В целом, аппарат семантических сетей позволяет разрабатывать формальные алгоритмы процессов поиска и обработки информации посредством анализа и изменения состояний элементов и конфигурации семантической сети. Несмотря на широкую распространенность семантических сетей в качестве модели представления знаний, общепринятой теории, определяющей методы кодирования и переработки информации в семантических сетях, до сих пор не существует.

Современные интеллектуальные системы характеризуются большими объемами хранимой и перерабатываемой информации, многоуровневой иерархической структурой знаний, активным использованием метаинформации. Как следствие, резко возрастает сложность используемых семантических сетей, структура которых становится неоднородной и нерегулярной, а алгоритмы обработки информации основываются, как правило, на эвристиках, отличных от системы к системе. Все это обуславливает актуальность проблемы создания эффективной аппаратной платформы с параллельной архитектурой, ориентированной на класс задач семантической обработки.

Согласно ряду проведенных исследований

[Голенков и др., 2012], теоретическим фундаментом необходимой унификации может выступить концепция, основная идея которой заключается в платформенной независимости создаваемых методов, алгоритмов и программ семантической обработки информации (т.е. независимости верхнего уровня интеллектуальной системы) от их аппаратной реализации. При этом нижний уровень интеллектуальной системы представляется абстрактной графодинамической машиной (ГДМ), непосредственно осуществляющей обработку данных (рис. 1). Процесс обработки информации в ГДМ трактуется как графодинамический процесс, т.е. как процесс преобразования графовой структуры семантической сети, в ходе которого меняется не только состояние элементов структуры, но и ее конфигурация (появляются и удаляются вершины, изменяются связи между ними).

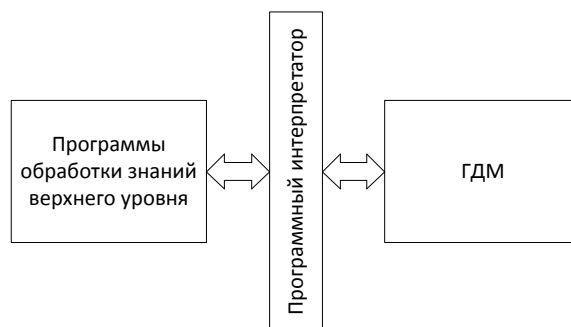


Рисунок 1 – Платформенно независимая интеллектуальная система

В качестве конкретной реализации абстрактной ГДМ может выступать любая вычислительная система с любой архитектурой, с любой аппаратной и программной платформами. Как следствие, их выбор будет влиять на трудоемкость программирования и производительность, которую они смогут обеспечить. Платформенная независимость достигается за счет программного интерпретатора, осуществляющего трансляцию программ семантической обработки верхнего уровня в формат команд конкретной ГДМ.

В настоящей работе рассматривается один из возможных вариантов реализации ГДМ с использованием оригинальной параллельной архитектуры.

1. Подход к построению архитектуры проблемно-ориентированного процессора

При разработке программ низкого уровня для ГДМ на основе мультитядерных РС, кластеров или суперЭВМ с универсальными параллельными архитектурами возникает ряд проблем связанных со сложностью, нерегулярностью и иерархичностью семантических сетей. В частности, распределение задач между процессорами и организация их взаимодействия, распределение данных между блоками памяти, высокая вычислительная сложность и нерегулярность вычислений создают неразрешимые проблемы при распараллеливании алгоритмов и, в конечном счете, приводят к эффекту замедления роста производительности при увеличении числа процессорных элементов, сформулированному в виде закономерностей Густавсона-Барсиса [Gustafson, 1988] и Амдаля-Уэра [Amdahl, 1967]. Так, по опыту эксплуатации системы «Эльбрус» при решении задач с нерегулярной структурой графа вычислений, эффективность обработки достигает максимума при работе четырех-пяти. При подключении большего количества процессоров эффективность начинает падать из-за растущих затрат на обеспечение взаимодействия процессоров.

В качестве альтернативы в [Байрак и др., 2012], [Вереник и др., 2012a], [Вереник и др., 2012b] предложен подход к построению архитектуры проблемно-ориентированных параллельных процессоров, который может быть использован для создания ряда специализированных устройств, реализующих ГДМ, сравнимых по производительности с суперЭВМ, а по стоимости – с универсальными мультитядерными компьютерами. Суть подхода состоит в формальной трансформации исходного графа семантической сети к некоторому регулярному графу, что позволит эффективно распараллелить вычислительные алгоритмы семантической обработки данных. В качестве типовой архитектуры ГДМ предлагается использовать SIMD-архитектуру магистрального типа (Single Instruction Multiply Data) с локальной

оперативной памятью для каждого процессорного элемента (рис. 2). Данная архитектура широко применяется во многих вычислительных системах и известна как одна из наиболее технологичных и простых в использовании.

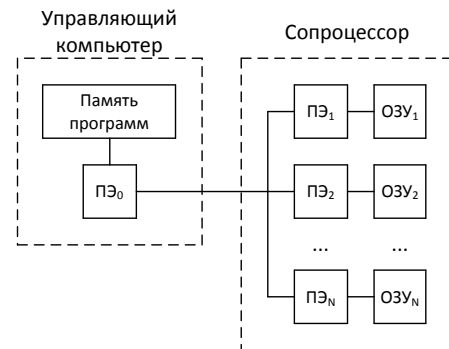


Рисунок 2 – Обобщенная блок-схема вычислительного комплекса

Фактически, знания интеллектуальной системы рассредоточены по множеству процессорных элементов (ПЭ) сопроцессора и имеют линейную регулярную структуру, соответствующую предложенному формату данных. Каждый ПЭ отвечает за доступ к единственному участку знаний системы, что позволяет исключить проблемы в организации межпроцессорного взаимодействия (устранение конфликтов доступа и т.п.). Функционально ПЭ сравним с простым компаратором, т.е. его аппаратная сложность минимизирована, чтобы позволить реализовать максимальное их количество на одном чипе.

Конструктивно вычислительный комплекс можно построить по одной из типовых схем, включающей в себя управляющий компьютер в качестве устройства управления и подключаемый сопроцессор, реализующий параллельные вычисления. Обмен данными между частями комплекса будет осуществляться по одному из стандартных типов интерфейсов.

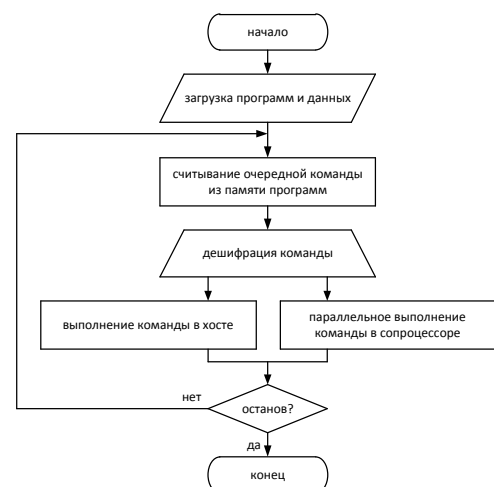


Рисунок 3 – Общий алгоритм функционирования вычислительного комплекса

В память управляющего компьютера записывается текст исполняемой программы (в терминах макрокоманд ГДМ), здесь же производится выборка команд, их дешифрация и передача для выполнения в сопроцессоре либо в хосте. На рис. 3 изображен общий алгоритм функционирования вычислительного комплекса.

Отличительной особенностью архитектуры является проблемная ориентация сопроцессора на операцию ассоциативного поиска. Операциям записи в этом устройстве соответствуют операции добавления, изменения либо удаления вершин или дуг графа, а операциям чтения – ассоциативный поиск данных по заданному шаблону. Отсутствие связей между ПЭ сопроцессора, а также организация локальной памяти обеспечивают линейную зависимость времени поиска от числа элементов ПЭ. Таким образом, достигается главная задача сопроцессора – обеспечение максимального быстродействия при выполнении операций ассоциативного поиска.

Из опыта проектирования сложных программно-аппаратных комплексов можно выделить следующие этапы разработки:

- разработка архитектуры и общих принципов функционирования;
- разработка программной модели и верификация архитектуры;
- выбор элементной базы, приобретение соответствующей аппаратной платформы, разработка макета;
- получение и оценка граничных технических характеристик;
- формирование ТЗ на разработку опытного образца; оптимизация технических решений под заданные технические условия.

2. Программная модель проблемно-ориентированного процессора

Прежде чем формулировать ТЗ на разработку аппаратного прототипа сопроцессора необходимо детализировать архитектуру и верифицировать алгоритмы вычислений. Для этого необходимо создать программную модель на РС, посредством которой выполнить ряд тестовых задач и проанализировать результаты выполнения.

Программная модель, в первую очередь, должна отражать совокупность функциональных возможностей (систему команд) вычислительного комплекса, которые прямо либо косвенно могут быть использованы программистом в ходе вычислений. Но также при разработке программной модели необходимо стремиться адекватно отразить ключевые блоки, межблочные информационные связи, форматы команд, способы размещения данных в памяти и доступа к ним. В ходе моделирования тестовая задача должна быть представлена так, как это ожидается в разрабатываемом вычислительном комплексе,

чтобы наиболее полно проверить принципы программного управления и ход параллельных вычислений, заложенные в архитектуру. В дальнейшем такая модель может стать основой программного эмулятора, позволяющего создавать и отлаживать прикладное программное обеспечение для вычислительного комплекса.

На рис. 4 представлена упрощенная диаграмма классов программы, позволяющей моделировать функциональные возможности и архитектурные свойства процессора. Ниже приведены ее основные особенности:

- текст исполняемой программы записывается в терминах макрокоманд процессора и хранится в отдельном текстовом файле;
- поддерживается программный интерфейс для интеграции интерпретатора с языка верхнего уровня;
- реализована возможность настройки атрибутов элементов графа (задавать как сам набор атрибутов, так и возможные принимаемые значения для каждого атрибута);
- реализована возможность настройки основных параметров процессора, таких как: размер локальной памяти, приходящийся на один ПЭ; количество ПЭ каждого типа; объем памяти системы и т.п.;
- все настройки системы хранятся во внешних файлах, позволяя легко управлять большим количеством различных конфигураций системы;
- существующая архитектура программной модели ориентирована на дальнейшую разработку программного обеспечения для разрабатываемого проблемно-ориентированного процессора.

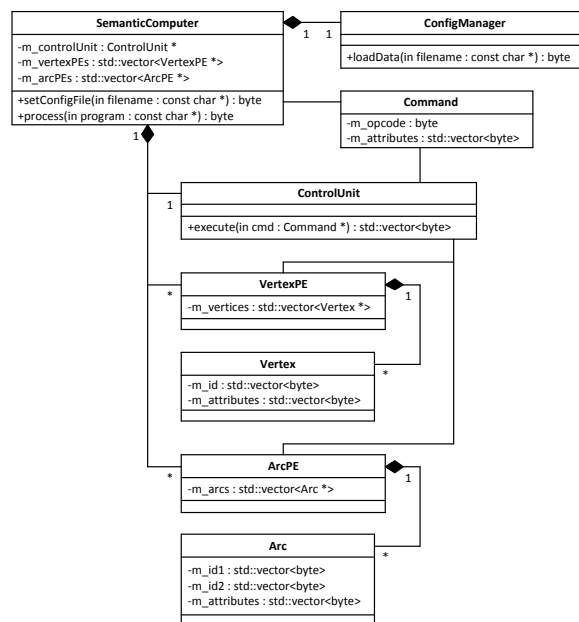


Рисунок 4 – Упрощенная диаграмма классов программной модели

3. ОБОСНОВАНИЕ ВЫБОРА ЭЛЕМЕНТНОЙ БАЗЫ И РАЗРАБОТКА МАКЕТА

В ходе этапа аппаратного проектирования в общем случае решаются следующие задачи:

- выбор элементной базы;
- выбор (приобретение или разработка) аппаратной платформы;
- отображение оригинальной архитектуры на аппаратной платформе (программирование «железа»).

Особенности элементной базы коренным образом будут отражаться на технических характеристиках разрабатываемого устройства. На сегодняшний день имеет смысл рассматривать следующие варианты элементной базы для аппаратной реализации сопроцессора.

1. FPGA или DSP (локальная память реализуется во внутренней блочной памяти). Подход имеет ограничения на объем внутренней блочной памяти (по сути, объем хранимой семантической сети), зато время доступа к памяти минимально, т.к. легко можно обеспечить доступ каждого ПЭ к своей локальной памяти. Данный вариант является наиболее простым при макетировании проблемно-ориентированных процессоров.

2. FPGA или DSP + внешние ОЗУ для реализации локальной памяти. Вариант не имеет принципиальных ограничений на объем локальной памяти, но время доступа к внешней памяти возрастает. Кроме того, для каждого внешнего ОЗУ необходим контроллер доступа к памяти, соответственно возрастают массогабаритные характеристики и энергопотребление устройства.

Замечание 1. По ряду очевидных причин не указан вариант разработки оригинальной элементной базы для реализации проблемно-ориентированных процессоров.

Замечание 2. Промежуточные и комбинированные конструктивные варианты, такие как каскадное включение FPGA или DSP, в работе не рассматривались из соображений простоты изложения.

Заключение

При создании прикладных интеллектуальных систем на основе семантических сетей существует проблема распараллеливания алгоритмов обработки информации при использовании многопроцессорных систем с классическими архитектурами. Предложен подход к решению проблемы на основе оригинальной проблемно-ориентированной архитектуры. Научным фундаментом выступает концепция платформенной независимости, при которой верхний уровень интеллектуальной системы разрабатывается независимо от нижнего (аппаратного) уровня.

В качестве аппаратной платформы

интеллектуальной системы предлагается разработать графодинамическую машину с архитектурой SIMD типа. Алгоритмической основой для такой архитектуры является предложенный механизм преобразования произвольной семантической сети в классический граф с линейной регулярной структурой. Полученный граф позволяет легко распараллелить вычислительные алгоритмы.

Рассмотрены этапы разработки перспективного вычислительного комплекса, включая создание программной модели проблемно-ориентированного процессора, а также краткое обоснование выбора элементной базы для создания макета.

Библиографический список

[Amdahl, 1967] Gene M. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities / Gene M. Amdahl // AFIPS Conference Proceedings (30). – pp. 483–485.

[Gustafson, 1988] John L. Gustafson. Reevaluating Amdahl's Law / John L. Gustafson // Communications of the ACM 31(5). – pp. 532–533.

[Байрак и др., 2012] Параллельные процессоры для построения интеллектуальных систем / С. А. Байрак, Д. Н. Одинец, М. М. Татур, Ф. Филипов, М. Мунос // Открытые семантические технологии проектирования интеллектуальных систем : материалы II Междунар. научн.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 135–140.

[Вереник и др., 2012a] Вереник Н. Л. Концепция построения графодинамической машины с SIMD-архитектурой / Н. Л. Вереник, М. М. Татур // Информационные технологии и системы 2012 (ИТС 2012) : материалы международной научной конференции, БГУИР, Минск, Беларусь, 24 октября 2012 г. / редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2012. – С. 178–179.

[Вереник и др., 2012b] Вереник Н. Л. Разработка проблемно-ориентированных процессоров семантической обработки информации / Н. Л. Вереник, Е. Н. Сейткулов, М. М. Татур // Электроника инфо. – 2012. – № 8. – С. 95–98.

[Голенков и др., 2012] Голенков В. В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В. В. Голенков, Н. А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем : материалы II Междунар. научн.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2012. – С. 23–52.

BUILDING INTELLIGENT SYSTEM ON VECTOR ASIP

Nick L. Verenik, Mikhail M. Tatur

*Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

nick.verenik@gmail.com

tatur@i-proc.com

The article considers one of the approaches to design of computing hardware support while building applied intelligent system on semantic networks. The concept of platform independence of system logic-semantic high-level from their hardware implementation is proposed as a scientific basis. The primary design phases are given: vector ASIP development, programming model development, electronic components choice reasoning.