

На правах рукописи

Литвинов Владимир Геннадьевич

**ПРИМЕНЕНИЕ ТИПОВЫХ РЕШЕНИЙ ДЛЯ РАСПАРАЛЛЕЛИВАНИЯ
АЛГОРИТМОВ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ**

05.13.18 – Математическое моделирование, численные методы
и комплексы программ

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Самара – 2013

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)» на кафедре информационных систем и технологий.

Научный руководитель: доктор технических наук, доцент
Востокин Сергей Владимирович

Официальные оппоненты: доктор технических наук, профессор, заведующий кафедрой «Вычислительная техника» ФГБОУ ВПО «Самарский государственный технический университет»
Орлов Сергей Павлович

доктор технических наук, доцент, профессор кафедры технической кибернетики ФГБОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)»
Попов Сергей Борисович

Ведущая организация: Федеральное государственное унитарное предприятие «Государственный научно-производственный ракетно-космический центр «ЦСКБ-Прогресс»

Защита состоится ____ 2013 г. в ____ часов на заседании диссертационного совета Д212.215.05 ФГБОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)», по адресу: 443086 Россия, г. Самара, Московское шоссе, д. 34.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВПО «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)».

Автореферат разослан ____ 2013 г.

Ученый секретарь
диссертационного совета,
д.т.н., доцент

С. В. Востокин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы диссертации.

Роль высокопроизводительной вычислительной техники в области математического моделирования очень важна. Её применение позволяет сократить время решения задач моделирования, повысить размерность решаемых задач, а, в некоторых случаях, строить решения, которые принципиально невозможно получить на последовательной ЭВМ или аналитически.

Переход от численного метода к реализации программы для высокопроизводительных вычислений на данный момент остается самостоятельной научной и инженерной проблемой. Практика показывает, что для программиста средней квалификации разработка параллельного алгоритма решения задачи составляет большую трудность. Данная проблема выделена академиком Г.И. Марчуком как фундаментальное научное направление, называемое «проблемой отображения».

Одним из методов решения проблемы отображения является повторное использование имеющихся типовых решений (шаблонов или паттернов) для распараллеливания алгоритмов разного назначения. Метод берет начало в работах Э. Гамма, Р. Хелма, Р. Джонсона и Дж. Влиссидеса. В области типовых решений для параллельного программирования известны работы отечественных и зарубежных ученых: М.И. Коула, С. Макдональда, Д. Шмидта, П.К. Берзигиярова, В.Э. Малышкина и других. Описано около двух десятков типовых решений в области параллельного программирования. Однако задача разработки моделей типовых решений, удобных для понимания специалистами прикладных областей и для конструирования программ на их основе, сохраняет актуальность.

Большое значение имеют также стоимость и сроки разработки программных комплексов численного моделирования. Процесс разработки и отладки параллельных программ в настоящее время отличается более низкой автоматизацией, чем автоматизация отладки последовательных программ в имеющихся интегрированных средах. Поэтому необходимо встраивание системы параллельного программирования в известные процессы разработки программного обеспечения и интегрированные среды разработки, поддерживающие их. Общий подход в решении перечисленных проблем – построение формальных моделей вычислительных процессов.

В качестве важного приложения методов синтеза параллельных программ на базе типовых решений в диссертации исследованы методы построения интеллектуальных систем на базе нейронных сетей, предложенных С. Осовским, С. Хайкиным, А.Н. Горбанём, с целью разработки эффективных параллельных алгоритмов обучения гиперрадиальных нейронных сетей для решения задач прогнозирования. Также рассмотрены методы преобразования сеточных алгоритмов к параллельному виду с использованием типовых решений.

Цель диссертационной работы состоит в разработке имитационной модели представления типовых решений для построения параллельных алгоритмов численного моделирования и исследовании эффективности данного подхода при постановке численных экспериментов на высокопроизводительных системах.

Задачи исследования:

1. Анализ используемых в вычислительной практике типовых решений и методов их формального представления.

2. Разработка имитационных моделей вычислительных процессов для типовых решений, применяемых при распараллеливании алгоритмов численного моделирования, на основе графической процессно-ориентированной нотации Templet¹⁾.

3. Разработка параллельного алгоритма численного решения задачи прогнозирования с использованием оптимизируемой структуры гиперрадиальной нейронной сети на основе типового решения «портфель задач».

4. Проверка эффективности построенных моделей алгоритмов и программ на примерах решения практических задач математического моделирования.

5. Разработка программного комплекса численного моделирования на основе типового решения «портфель задач» для использования на высокопроизводительных вычислительных системах.

Научная новизна диссертационной работы:

1. Построены новые имитационные модели типовых решений «портфель задач», «конвейер», «метод параллельных направлений» с использованием нотации Templet, которые, в отличие от известных, наглядно представляют декомпозицию численного метода на типы данных и обрабатывающие их процедуры, а также описывают протоколы взаимодействия процессов. Данное представление типового решения позволяет выполнить автоматический синтез параллельной программы.

2. С использованием типового решения «портфель задач» разработан параллельный алгоритм обучения гиперрадиальной нейронной сети. В отличие от известных алгоритмов он позволяет одновременно с обучением выполнять оптимизацию структуры сети. Тем самым снижается вычислительная сложность на стадии прогнозирования и время обучения.

3. Применённые в программах вычислительного комплекса алгоритмы, в отличие от известных систем параллельного программирования, позволяют инженеру-математику выполнить разработку численного метода в форме последовательной программы. Затем автоматически получить его параллельную реализацию, использующую, по выбору исследователя, MPI, потоки POSIX или Windows API. Далее провести вычислительные эксперименты на высокопроизводительной ЭВМ.

¹⁾ Востокин С.В. Templet – метод процессно-ориентированного моделирования параллелизма // Программные продукты и системы, 2012. №3. С. 9-12

На защиту выносятся:

1. Имитационные модели типовых решений «портфель задач», «конвейер», «метод переменных направлений», построенные с использованием процессно-ориентированной нотации Templet.

2. Параллельный алгоритм обучения гиперрадиальной нейронной сети, сочетающий обучение и оптимизацию структуры нейронов, построенный с использованием типового решения «портфель задач».

3. Результаты численного решения задачи прогнозирования с использованием параллельного алгоритма обучения гиперрадиальной нейронной сети с оптимизацией её структуры; результаты исследования эффективности распараллеливания численных моделей с использованием типовых решений.

4. Программный комплекс численного моделирования на основе типового решения «портфель задач» для решения ресурсоёмких задач на высокопроизводительных вычислительных системах.

Достоверность результатов работы.

Достоверность полученных в диссертации решений подтверждена корректными математическими выкладками, результатами вычислительных экспериментов на высокопроизводительной технике, а также соответствием результатов математического моделирования реальным процессам.

Научная и практическая значимость работы.

Полученные в диссертации имитационные модели типовых решений позволяют автоматизировать постановку вычислительного эксперимента на высокопроизводительной вычислительной технике инженером-исследователем, не являющимся специалистом в области системного программного обеспечения параллельных ЭВМ. Это сокращает время проведения вычислительного эксперимента, исключает возможные ошибки в системном коде управления параллельными вычислениями, расширяет круг специалистов, применяющих высокопроизводительную вычислительную технику для решения задач математического моделирования.

Параллельные численные методы обучения и оптимизации структуры гиперрадиальной нейронной сети, построенные на основе типового решения «портфель задач», позволили повысить качество решения задачи прогнозирования и сократить время обучения нейронной сети в ~7 раз на 8-ми ядерной системе.

Разработанный в диссертации программный комплекс позволяет в целом автоматизировать постановку вычислительного эксперимента на суперкомпьютере или кластерной системе посредством удобного управления запуском кода в удаленной системе и контроля его исполнения через web-интерфейс.

Связь диссертационной работы с планами научных исследований.

Диссертационная работа выполнена в рамках работ по проекту «Разработка комплекса технологий использования ресурсов суперкомпьютера «Сергей Королёв» в целях развития инновационной и научно-

образовательной среды университета» (мероприятия 3.3 Программы развития национального исследовательского университета).

Результаты диссертационной работы внедрены в научной деятельности факультета летательных аппаратов СГАУ и применялись при выполнении НИР для анализа многомерных динамических систем и процессов с использованием суперкомпьютера «Сергей Королёв», о чем имеется акт внедрения.

Апробация результатов работы.

Основные результаты работы были представлены на следующих всероссийских и международных конференциях: Всероссийской молодежной научной конференции с международным участием «X Королёвские чтения» (Самара, 2009); Международной конференции с элементами научной школы для молодежи «Перспективные информационные технологии для авиации и космоса (ПИТ-2010)» (Самара, 2010); Всероссийской молодежной научной конференции с международным участием «XI Королёвские чтения» (Самара, 2011); Шестой Международной конференции Института проблем управления имени В.А. Трапезникова РАН «Параллельные вычисления и задачи управления» (Москва, 2012); Научно-технической конференции с международным участием «Перспективные информационные технологии в научных исследованиях, проектировании и обучении (ПИТ-2012)» (Самара, 2012).

Публикации по теме диссертации. По теме диссертации опубликовано 9 работ, в том числе 2 статьи в журналах, рекомендованных ВАК РФ, 5 работ в материалах и трудах международных и всероссийских конференций, 2 тезиса доклада.

Авторский вклад. Все результаты, изложенные в диссертации, получены автором лично, либо при его определяющем личном участии. Автор принимал непосредственное участие в постановке задач, разработке моделей типовых решений и их реализации. Автор осуществлял проведение численных экспериментов, обработку, анализ и интерпретацию полученных результатов.

Структура и объем работы. Диссертация состоит из введения, трех глав, заключения и одного приложения. Общий объем диссертации __ страниц. Диссертация содержит __ таблиц, __ рисунков и список литературы из __ наименований.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертации, определена цель и сформулированы задачи работы, изложена постановка задачи диссертационной работы, научная новизна и практическая значимость полученных результатов. Приводятся основные положения, выносимые на защиту.

В первой главе приводится обзор актуальных проблем разработки параллельных программ и выполнено построение имитационных моделей типовых решений для распараллеливания численных алгоритмов.

Для описания типовых решений взяты алгоритмы, структуру которых можно описать ориентированным графом вида (1)

$$G_M = (P, C), P \subset N, C \subset N \times N, \quad (1)$$

в котором вершины P представляют собой процессы, а дуги C - каналы, по которым передаются сообщения между процессами. Для описания вычислительного процесса, порождаемого такими алгоритмами, применена темпоральная логика Лампорта. Состояние вычислительного процесса – это значения множества переменных (2), связанных с процессами и каналами

$$f \equiv (i \in P : a_i; (i, j) \in C : e_{i,j}, d_{i,j}), \text{ где } a_i \in \{0, 1\}, e_{i,j} \in \{0, 1\}, d_{i,j} \in \{0, 1\}. \quad (2)$$

Логические переменные a_i обозначают пассивное 0 или активное 1 состояние процесса i ; $e_{i,j}$ - пассивное 0 или активное 1 состояние канала (i, j) ; $d_{i,j}$ – направление передачи сообщений по двунаправленным каналам ($1 - i \rightarrow j$; $0 - j \rightarrow i$).

В начальном состоянии (3) некоторые каналы находятся в активном состоянии с передачей сообщений в направлении $i \rightarrow j$, все процессы находятся в пассивном состоянии:

$$I \equiv \forall i \in P : \neg a_i \wedge \forall (i, j) \in C : (e_{i,j} \wedge d_{i,j}) \vee (\neg e_{i,j} \wedge \neg d_{i,j}). \quad (3)$$

Действия, выполняемые каналами, описываются формулой (4)

$$A_C(i, j) \equiv \{e_{i,j} \wedge \neg e'_{i,j} \wedge ((d_{i,j} \wedge d'_{i,j}) \vee (\neg d_{i,j} \wedge \neg d'_{i,j}))\} \vee \{ \neg e_{i,j} \wedge e'_{i,j} \wedge ((d_{i,j} \wedge \neg d'_{i,j}) \vee (\neg d_{i,j} \wedge d'_{i,j}))\}. \quad (4)$$

Действия описывают соседние в истории вычислительного процесса состояния, переменные следующего состояния обозначены апострофом. Согласно формуле (4) состояние канала периодически меняется с активного на пассивное, а направление передачи данных с прямого на обратное.

Действия, выполняемые процессами, описываются формулой (5). Процесс становится активным, когда из смежного с ним канала поступает сообщение. При этом такой канал сам переходит в пассивное состояние. Среди нескольких каналов, готовых передать сообщение в процесс, выбирается произвольный канал. Когда процесс завершит работу и перейдет в пассивное состояние, по одному или нескольким каналам от него могут передаваться сообщения в другие процессы.

$$\begin{aligned}
A_p(i) \equiv & \{ \neg a_i \wedge a'_i \wedge (\exists j : e_{i,j} \wedge \neg e'_{i,j} \wedge \neg d'_{i,j}) \wedge \neg (\exists j : e_{j,i} \wedge \neg e'_{j,i} \wedge d'_{j,i}) \} \vee \\
& \{ \neg a_i \wedge a'_i \wedge (\exists j : e_{j,i} \wedge \neg e'_{j,i} \wedge d'_{j,i}) \wedge \neg (\exists j : e_{i,j} \wedge \neg e'_{i,j} \wedge \neg d'_{i,j}) \} \vee \\
& \left\{ a_i \wedge \neg a'_i \wedge \left(\bigvee_{k:(i,k) \in C} \neg e_{i,k} \wedge e'_{i,k} \wedge d'_{i,k} \right) \right\} \vee \\
& \left\{ a_i \wedge \neg a'_i \wedge \left(\bigvee_{k:(k,i) \in C} \neg e_{k,i} \wedge e'_{k,i} \wedge \neg d'_{k,i} \right) \right\} \vee \\
& \{ a_i \wedge \neg a'_i \}
\end{aligned} \quad . \quad (5)$$

С учетом выражений (4) и (5), действия, выполняемые моделируемым вычислительным процессом, можно представить в виде (6)

$$N \equiv \bigvee_{(i,j) \in C} A_C(i,j) \vee \bigvee_{i \in P} A_p(i). \quad (6)$$

Способ планирования вычислительного процесса характеризуется выражением (7)

$$F \equiv WF_f \left(\bigvee_{i \in P} (a_i \wedge \neg a'_i) \vee \bigvee_{(i,j) \in C} (e_{i,j} \wedge \neg e'_{i,j}) \right), \quad (7)$$

подразумевающим справедливую в слабом смысле стратегию планирования. Окончательно получаем модель вычислительных процессов для представления типовых решений (8):

$$\Phi_M \equiv I \wedge \Box [N]_f \wedge F. \quad (8)$$

По построению вычислительного процесса порядок смены состояний, удовлетворяющий формуле (8), легко воспроизвести дискретно-событийной имитационной моделью. Для этого организуется очередь каналов, находящихся в активном состоянии. На каждом шаге моделирования случайным образом выбирается активный канал. Далее он переводится в неактивное состояние, выполняется действие процесса и отправка сообщений из данного процесса.

Для представления типового решения далее требуется определить сообщения, передаваемые в каналах и методы процессов, запускаемые при их обработке. Для этого применена графическая нотация Templet. Модели процессов Templet напоминают блок-схемы алгоритмов рис.1-3. Однако имеется несколько начальных вершин, соответствующих портам. Не делается различий между процедурой и принятием решения. Поток управления может пройти через вершину-метод, при этом метод может не выполняться, если не соблюдены некоторые условия. Данные свойства позволяют удобно описать выбор метода обработки поступающих в процесс сообщений. В отличие от блок-схем точно задан контекст запуска метода. Определено, какие сообщения пересылаются между процессами и их порядок. Это позволило провести анализ непротиворечивости графического описания алгоритма. Семантика исполнения модели по построению гарантирует отсутствие

состояния состязания (race condition) при реализации алгоритма в общей памяти и состояния тупика (deadlock). Хотя для моделей вида (8) возможна остановка в незавершенном состоянии, данная ситуация более проста, чем состояние тупика.

Типовыми схемами решения многих переборных и оптимизационных задач являются схемы «управляющий - рабочие». Рассматривается модель одной из общих схем управления вычислениями из данного класса: «вычисление с портфелем задач». Это схема с активными рабочими, реализующая управление pull-типа. Здесь иллюстрируется описание более сложного протокола взаимодействия процессов с несколькими типами сообщений (по сравнению с исходными (4) протоколами «запрос-ответ»).

На рис. 1 показана композиция процессов схемы «портфель задач», соответствующая графу (1). Обработка заданий выполняется по инициативе рабочих процессов $w_1, \dots, w_N$, что видно из направления каналов $p \rightarrow p_1: \text{Link}, \dots, p \rightarrow p_N: \text{Link}$ от рабочих к управляющему процессу.

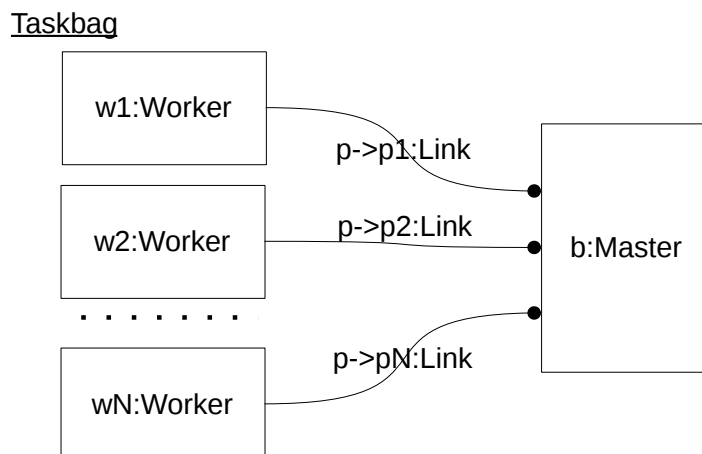


Рис. 1. Композиция процессов при вычислении с портфелем задач

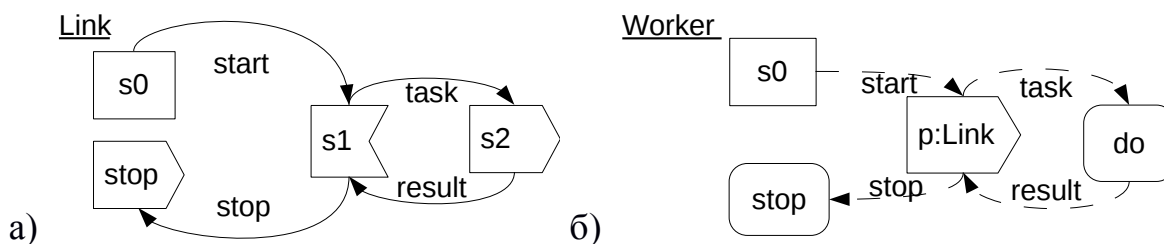


Рис. 2. Протокол канала (а) и рабочий процесс (б) алгоритма вычисления с портфелем задач

Роль рабочих процессов рис. 2 (б) заключается в периодическом получении задания от управляющего процесса в сообщении `task` и отправки результата его обработки `result`. Обработка выполняется в методе `do`. Цикл обработки начинается по инициативе рабочего с отправки сообщения `start` и заканчивается при получении от управляющего процесса сообщения `stop`. Если потребуется, в методе `stop` можно выполнить операцию очистки.

На рис. 3 показан фрагмент управляющего алгоритма, реализованного в процессе Master, относящийся к порту p_N . Методы процесса get, put, if_task и start реализуют специфические для решаемой задачи операции: get – извлечение очередной задачи; put – сохранение результата; if_task – проверку наличия задачи для обработки. Метод start может использоваться для инициализации.

Методы if_active, wait_N, stop_N, waiting_N являются универсальными для произвольного численного метода. Метод if_active возвращает истинное значение, если имеются активные рабочие процессы. То есть, имеются каналы, не находящиеся в состоянии «доставлено сообщение start или result», что в программной реализации определяется простым подсчетом при поступлении и отправке сообщений. Метод wait_N устанавливает признак, что канал N находится в состоянии ожидания появления заданий, метод waiting_N возвращает значение данного признака. Метод stop_N используется при отправке сообщения stop рабочему для начала очистки.

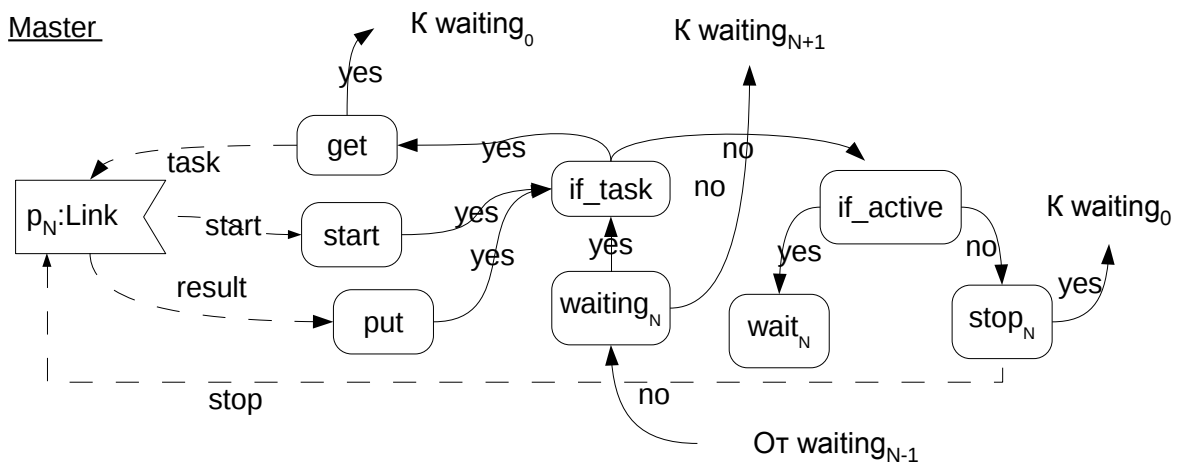


Рис. 3. Фрагмент диаграммы управляющего процесса для алгоритма вычисления с портфелем задач

Остановка вычислений и очистка начинаются, если нет задач и нет активных рабочих процессов. Когда сформирована и отправлена очередная задача, происходит поиск ожидающего рабочего процесса. Если такой процесс найдется и есть еще одна задача для обработки, то выполняется ее формирование и отправка рабочему. В противном случае управляющий процесс либо заканчивает обработку сообщения и ждет ответа от рабочих, либо начинает остановку вычислений и очистку.

Также в первой главе рассмотрено построение модели типовых решений «метод переменных направлений» и «конвейер», что демонстрирует адекватность рассмотренного подхода для моделирования вычислительных процессов, возникающих в задачах численного моделирования.

Во второй главе приводится обзор численных методов прогнозирования, и решена задача прогнозирования временного ряда с использованием гиперрадиальной нейронной сети. Для данной сети строится

последовательный алгоритм обучения, затем по последовательному алгоритму с использованием типового решения «портфель задач» синтезируется параллельный алгоритм. Показаны результаты численного моделирования.

Принципы функционирования гиперрадиальной нейронной сети рис. 4 описываются следующими выражениями:

$$v_j^{(k)}(\mathbf{X}_p) = x_{p,j} - c_j^{(k)}, \quad (9)$$

$$z_i^{(k)}(\mathbf{X}_p) = \sum_{j=1}^N (q_{i,j}^{(k)} \cdot v_j^{(k)}(\mathbf{X}_p)), \quad (10)$$

$$u^{(k)}(\mathbf{X}_p) = \sum_{i=1}^N (z_i^{(k)}(\mathbf{X}_p))^2, \quad (11)$$

$$f^{(k)}(\mathbf{X}_p) = \exp\left(-\frac{1}{2} \cdot u^{(k)}(\mathbf{X}_p)\right), \quad (12)$$

$$y(\mathbf{X}_p) = w^{(0)} + \sum_{i=1}^K (w^{(i)} \cdot f^{(i)}(\mathbf{X}_p)), \quad (13)$$

где N - количество выходных параметров сети, K - количество нейронов сети, $\mathbf{X}_p$ - p -й входной вектор для обучения, $\mathbf{W}$ - вектор весов выходного слоя, $\mathbf{C}^{(k)}$ - вектор центров k -ой радиальной функции, $\mathbf{Q}^{(k)}$ - матрица диаметров k -ой радиальной функции.

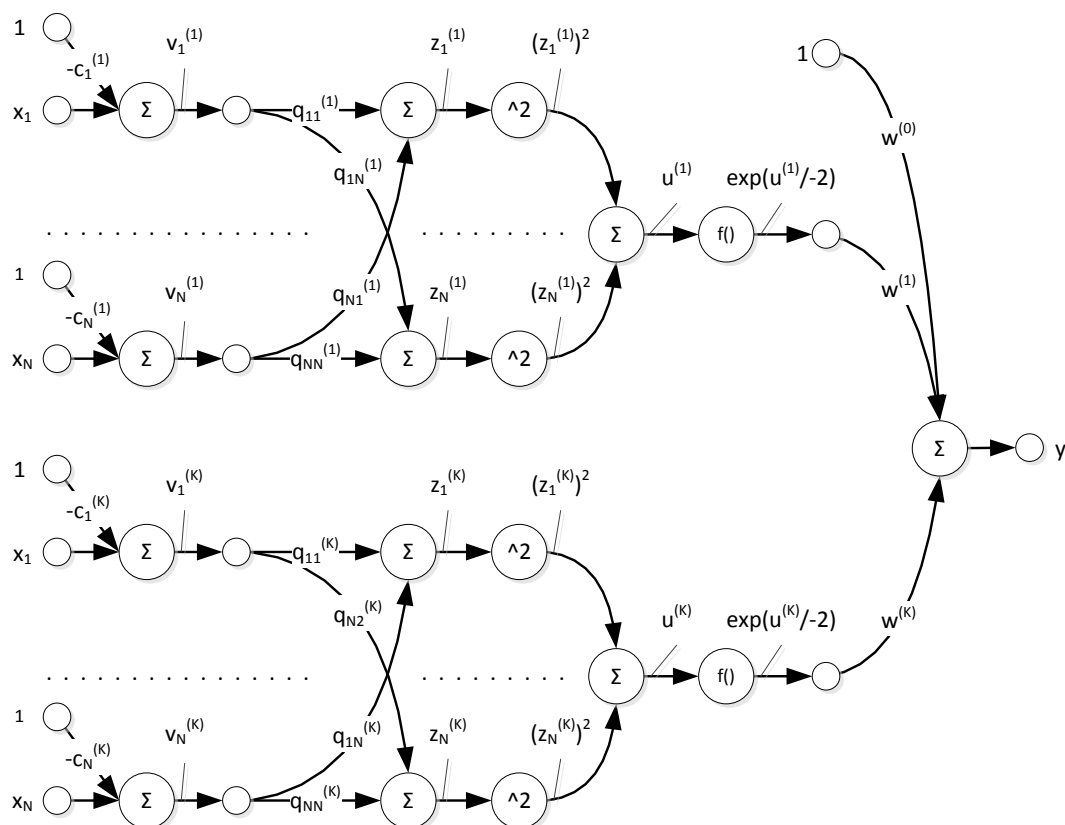


Рис. 4. Гиперрадиальная нейронная сеть

На основании выражений (9)-(13) вводится функция ошибки нейронной сети, используя Евклидову метрику:

$$E(\mathbf{X}, \mathbf{D}) = \frac{1}{2} \cdot \sum_{p=1}^P \left(y(\mathbf{X}_p) - d_p \right)^2, \quad (14)$$

где P - размер обучающей выборки, d_p - p -е ожидаемое выходное значение для обучения.

Для функции ошибки составляющие градиента равны:

$$\frac{\partial E(\mathbf{X}, \mathbf{D})}{\partial c_j^{(k)}} = \sum_{p=1}^P \left((y(\mathbf{X}_p) - d_p) \cdot w^{(k)} \cdot f^{(k)}(\mathbf{X}_p) \cdot \sum_{i=1}^N (z_i^{(k)}(\mathbf{X}_p)) \cdot q_{i,j}^{(k)} \right), \quad (15)$$

$$\frac{\partial E(\mathbf{X}, \mathbf{D})}{\partial q_{i,j}^{(k)}} = - \sum_{p=1}^P \left((y(\mathbf{X}_p) - d_p) \cdot w^{(k)} \cdot f^{(k)}(\mathbf{X}_p) \cdot z_i^{(k)}(\mathbf{X}_p) \cdot v_j^{(k)}(\mathbf{X}_p) \right), \quad (16)$$

Подбор количества базисных функций, каждой из которых соответствует один скрытый нейрон, является основной проблемой, возникающей при создании и обучении сети. Вследствие невозможности априорного определения точного количества скрытых нейронов применяются адаптивные методы, которые позволяют добавлять и удалять их в процессе обучения.

Для определения стратегии формирования структуры сети в последовательном алгоритме был применен подход каскадного наращивания сети. Подбор структуры сети при таком подходе происходит параллельно с ее обучением путем добавления на каждом этапе обучения одного скрытого нейрона.

Таким образом, определение структуры сети и реализацию алгоритма ее обучения можно трактовать как выполнение подбора оптимальной архитектуры искусственной нейронной сети.

Каждый нейрон старается адаптировать веса своих связей, чтобы быть востребованным для выполняемого сетью отображения данных. На начальном этапе формируется сеть, состоящая только из входных узлов и выходного нейрона и двух скрытых нейронов. Количество входов зависит исключительно от специфики решаемой задачи и не подлежит модификации. Скрытые нейроны добавляются в сеть по одному. Каждый добавляемый нейрон подключается ко всем входам сети, и его центр устанавливается в месторасположение выборки данных, на которой возникает максимальное отклонение, а диагональные элементы матрицы радиусов уточняются по формуле

$$q_{i,i}^{(k+1)} = \frac{1}{|x_{p,i} - c_i^{(l)}|}, \quad (17)$$

где $k+1$ – индекс добавленного нейрона, l – индекс ближайшего нейрона, p – индекс выборки данных с максимальным отклонением, остальные элементы $\mathbf{Q}$ принимаются равными нулю. Далее производится уточнение коэффициентов сети гибридным алгоритмом.

Процесс обучения и формирования структуры сети завершается, если результат функционирования сети считается удовлетворительным с точки зрения допустимого отклонения, или достигнуто ограничение по количеству нейронов. В противном случае структура сети повторно расширяется и производится уточнение коэффициентов.

Параллельный алгоритм обучения нейронной сети и подбора оптимального количества нейронов скрытого слоя основан на использовании метода грубой силы. Он заключается в одновременном обучении нескольких сетей, различающихся случайным положением центра нейрона добавленного в скрытый слой. На каждой итерации производится выбор лучшей сети по критерию достижения минимального значения допустимого отклонения.

Применение типового решения «портфель задач» рис. 1-3 для распараллеливания состоит в определении методов `get`, `put`, `if_task`, состояния портфеля задач, собственно задачи и результата её вычисления.

В портфеле задач для управления вычислениями хранятся: n – текущее количество нейронов в скрытом слое; N – максимальное число нейронов в скрытом слое; m – текущее количество запущенных задач; r – число решенных задач; M – максимальное количество запущенных задач (выбирается в 2-3 раза большим числа рабочих процессов); Σ — значение ошибки, достижение которой останавливает процесс обучения; коэффициенты обученной нейронной сети с n нейронами Φ_n (включают вектор центров, матрицу диаметров и вектор весов выходного слоя); Φ_{n+1} — коэффициенты обученной нейронной сети с $n+1$ нейронами; ε – значение ошибки для обученной сети с $n+1$ нейронами.

Задача определяется коэффициентами Φ'_n и положением центра добавляемого нейрона d' . Результат её вычисления — коэффициенты Φ'_{n+1} и значение ошибки ε' . Как и в последовательном случае, при решении задачи производится уточнение коэффициентов сети Φ'_{n+1} гибридным алгоритмом.

В начальном состоянии (start) определяются постоянные алгоритма N , M , Σ ; переменные $m:=0$, $r:=0$, $n:=2$, $\Phi_n:=\Phi_2$ (параметры сети с двумя нейронами), Φ_{n+1} — не определены, $\varepsilon \gg \Sigma$.

Проверка наличия задачи (if_task). Возможна выборка очередной задачи на обработку, если одновременно выполняются следующие условия: не достигнута заданная ошибка обучения $\varepsilon > \Sigma$; остались не выданные задачи $m < M$.

Алгоритм извлечения задачи (get). Присвоить $\Phi'_n := \Phi_n$; определить случайное положение центра нейрона d' . Увеличить счетчик запущенных задач $m:=m+1$.

Алгоритм помещения результата вычисления задачи (put). Если $\varepsilon' < \varepsilon$, то определить $\varepsilon := \varepsilon'$ и $\Phi_{n+1} := \Phi'_{n+1}$. Увеличить счетчик решенных задач $r := r+1$. Если $r = M$, то закончено обучение слоя. В этом случае, если $n < N$, переходим к следующему слою: $n := n+1$, $m := 0$, $r := 0$; определяем ε так, что $\varepsilon \gg \Sigma$.

Алгоритм завершается, когда будут обучены все сети с числом нейронов до N включительно, либо будет достигнуто заданное значение ошибки обучения $\varepsilon \leq \Sigma$ на $n < N$ числе нейронов скрытого слоя.

В проведённых вычислительных экспериментах обучение сети выполнялось на выборках котировок фондовых рынков. В качестве базы для экспериментального вычисления ускорения брался последовательный алгоритм с последовательным добавлением нейронов в сеть. Эксперимент показал, что незначительные модификации последовательного алгоритма позволили добиться сокращения времени вычисления примерно в 7 раз на восьмизерной системе.

Численные эксперименты также показали хорошую достоверность методов прогноза на примере числовых рядов биржевых котировок. Знак тренда котировок был достоверен на интервале до 9-ти дней, а значение тренда с точностью до 2% был достоверен в течении 5-ти дней.

Отметим, что данный алгоритм является распределенным по построению и может исполняться как в системах с разделяемой, так и в системах с распределенной памятью. Метод прогноза и техника его распараллеливания являются универсальными и могут применяться для решения подобных задач с использованием типового решения «портфель задач».

В третьей главе описываются программный комплекс численного моделирования на основе типовых решений и результаты его применения для ряда задач математического моделирования.

Алгоритм обучения нейронной сети реализован на языке C++ с использованием библиотеки CLAPACK. Для типового решения «портфель задач» разработаны среды многопоточного исполнения на основе Windows API, POSIX и MPI, позволяющие проводить высокопроизводительные вычисления на системах с общей и распределённой памятью.

Оценка снижения трудоёмкости при построении параллельных численных методов на основе типовых решений получена измерением объемов ручного и автоматического кодирования.

Показано, что программный комплекс, наряду с преимуществами в скорости разработки и возможностью автоматически запускать исходную последовательную программу на разных высокопроизводительных вычислительных системах, обеспечивает достаточно эффективное распараллеливание кода. В вычислительных экспериментах по распараллеливанию задачи нелинейной динамики при использовании типового решения «метод переменных направлений» на рабочей станции получено ускорение 2,28 и эффективность 57,1%; на узле суперкомпьютера — ускорение 3,36 и эффективность 42%. Для задачи о распространении плоской световой волны на рабочей станции получено ускорение 1,91 и эффективность 95,5%; на узле суперкомпьютера — ускорение 4,92 и эффективность 61,5%. В задаче моделирования фазовых траекторий динамических систем на суперкомпьютере с использованием типового

решения «портфель задач» достигнуто 120 кратное ускорение при вычислении 160 траекторий на 20 восьмиядерных узлах (эффективность 75%).

Реализованные в диссертационной работе типовые решения входят в состав программного комплекса Templet Web, выполненного в виде web-сервиса для автоматизации параллельных вычислений и размещенного по адресу <http://templet.ssau.ru>.

В заключении сформулированы основные выводы и результаты, полученные в работе:

1. Построены имитационные модели типовых решений «портфель задач», «конвейер», «метод параллельных направлений» в процессно-ориентированной нотации Templet. Показано, что данная техника моделирования позволяет наглядно представить декомпозицию параллельного алгоритма на процедуры и обрабатываемые ими типы данных, а также представить протоколы взаимодействия процессов, абстрагируясь от конкретной реализации параллельного алгоритма.

2. Разработан параллельный алгоритм обучения гиперрадиальной нейронной сети с использованием типового решения «портфель задач». Продемонстрирована возможность оптимизации количества нейронов скрытого слоя в структуре нейронной сети. Получено сокращение времени обучения нейронной сети в ~ 7 раз на 8-ми ядерной системе без снижения точности прогноза.

3. Разработан и передан в эксплуатацию на суперкомпьютер «Сергей Королев» Самарского государственного аэрокосмического университета программный комплекс численного моделирования на основе типового решения «портфель задач».

4. В экспериментах на высокопроизводительной технике продемонстрировано снижение трудоемкости с сохранением эффективности вычислений при распараллеливании численных алгоритмов с использованием имитационных моделей типовых решений, рассмотренных в диссертации.

Основные результаты диссертации опубликованы в следующих работах:

Статьи в изданиях, рекомендованных ВАК:

- 1) Востокин, С.В. Программный комплекс параллельного программирования Graphplus templet [Текст] / С.В. Востокин, **В.Г. Литвинов**, А.Р. Хайрутдинов // Вестник Самарского гос. техн. ун-та. Серия: Физ.-мат. науки. 2011. №4(25). С. 146-153.
- 2) Востокин, С.В. Применение комплекса параллельного программирования Graphplus templet в моделировании [Текст] / С.В. Востокин, **В.Г. Литвинов**, А.Р. Хайрутдинов // Программные продукты и системы. 2012. №3(99). С.12-16.

Статьи в других изданиях:

- 3) **Литвинов, В.Г.** Нейросетевой аппроксиматор на основе радиально-базисных функций с масштабирующей матрицей [Текст] / **В.Г. Литвинов** // Всероссийская молодежная научная конференция с международным участием «X королевские чтения». Самара. 2009. С.293.
- 4) **Литвинов, В.Г.** Экстраполяция поведения фондового рынка искусственной нейронной сетью на основе радиально базисных функций [Текст] / **В.Г. Литвинов** // Тезисы докладов XXXVI самарской областной студенческой научной конференции, Самара. 2010. С.307-308.
- 5) Солдатова, О.П. Методы построения оптимальной структуры HRBF нейронной сети для решения задачи прогнозирования [Текст] / О.П. Солдатова, **В.Г. Литвинов** // Перспективные информационные технологии для авиации и космоса (ПИТ-2010). Труды Международной конференции с элементами научной школы для молодежи. Самара. 2010. С.252-254.
- 6) **Литвинов, В.Г.** Разработка комплекса типовых решений параллельных вычислений на основе модели «Граф Плюс» на суперкомпьютерных системах [Текст] / **В.Г. Литвинов** // Тезисы докладов XXXVII Самарской областной студенческой научной конференции. Часть I. Самара. 2011. С.275.
- 7) **Литвинов, В.Г.** Моделирование динамики пространственно-распределенных систем типа «реакция-диффузия» с использованием паттерна «портфель задач» на суперкомпьютере [Текст] / **В.Г. Литвинов** // Всероссийская молодежная научная конференция с международным участием «XI Королёвские чтения». Самара, СГАУ. 2011. С.293.
- 8) Востокин, С.В. Визуальное моделирование параллельных алгоритмов в процессно-ориентированной нотации Templet [Текст] / С.В. Востокин, **В.Г. Литвинов**, Д.Д. Макагонова, А.Р. Хайрутдинов // Параллельные вычисления и задачи управления: Тр. Шестой Международн. конф-ции. М.: ИПУ РАН, 2012. Том 1. С.260-269.
- 9) Востокин, С.В. Представление алгоритмов высокопроизводительных вычислений [Текст] / С.В. Востокин, **В.Г. Литвинов**, Д.Д. Макагонова, А.Р. Хайрутдинов // Перспективные информационные технологии в научных исследованиях, проектировании и обучении (ПИТ-2012): Тр. научно-технической конференции с международным участием. Самара. 2012. С.74-76.

Формат 60x84 1/16. Усл. печ.л. 1,00
Тираж 100 экз. Заказ №____

Отпечатано с готового оригинал-макета. Типография СГАУ
443086, г. Самара, Московское шоссе, 34