РАЗДЕЛ XII. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Егоров. А.Н., Кузнецов В.А.

Решение прикладных задач в среде универсальной программной оболочки распараллеливания

*Государственный университет морского и речного флота имени адмирала С.О.
Макарова
(Россия, Санкт-Петербург)*

doi: 10.18411/lj-31-08-2017-37

idsp: 000001:lj-31-08-2017-37

Аннотация

В статье рассматривается универсальный подход к организации параллельной обработки данных, реализованный в программной оболочке распараллеливания (ПОР) и особенности решения прикладных задач в среде ПОР.

Ключевые слова: параллельные алгоритмы, высокопроизводительные вычисления.

В последнее время наблюдается резкое усиление внимания к использованию высокопроизводительных средств вычислительной техники. Это объясняется тем, что с одной стороны, растёт число пользователей персональных компьютеров, а с другой стороны, происходит концентрация мощных вычислительных ресурсов в центрах коллективного пользования и, как следствие, развитие инфраструктуры удалённого доступа с использованием средств телекоммуникации.

Сейчас нет необходимости в доказательстве того факта, что дальнейшее повышение производительности компьютеров только за счёт улучшения характеристик элементов электронных схем практически достигло предела, который определяется законами физики. Поэтому дальнейшее повышение производительности возможно лишь за счёт распараллеливания процессов обработки информации. Опыт показывает, что построение параллельных вычислительных процессов, назначение которых состоит в обеспечении максимальной производительности, является важной самостоятельной проблемой [1].

Известно, что перенос обычной программы на многопроцессорную вычислительную систему может не дать ожидаемого выигрыша в повышении производительности. Более того, в результате такого переноса программа может работать медленнее. На практике, написать эффективную параллельную программу, или приспособить уже имеющуюся последовательную программу для параллельных вычислений, достаточно трудоёмкая задача. Это объясняется тем, что переход к использованию многопроцессорных систем характеризуется принципиально новым содержанием.

Рассмотрим универсальный подход к организации параллельной обработки данных, реализованный в программной оболочке распараллеливания (ПОР) и особенности решения прикладных задач в среде ПОР.

Программная реализация оболочки предназначена для распараллеливания процесса вычислений в прикладных решениях, используя для этих целей различные аппаратные средства [2]. Каждому типу устройств соответствует свой обработчик. Поддержка прикладных решений реализуется в виде dll-библиотек, разработанных для каждого типа устройства, с которыми работает ПОР. Каждая библиотека должна содержать значения параметров, требуемых для настройки среды программной оболочки, таких как: размер массива бинарных данных задачи и результата её решения;

идентификатор задачи, и др. Так же необходимо реализовать ряд экспортных функций, обеспечивающих выполнение следующих операций:

1. Parser – проверка, обработка и инициализация пользовательской задачи (ПЗ);
2. Score – оценка производительности устройства;
3. Cut – декомпозиция ПЗ на подзадачи;
4. SimSim – решение подзадач и формирование результата ПЗ;

ПОР оперирует данными методами для организации параллельного выполнения прикладной задачи. Рассмотрим подробнее особенности использования dll-библиотек в процессе решения ПЗ.

При запуске программной оболочки выполняется анализ подключённых к компьютеру вычислительных устройств, с которыми поддерживает работу ПОР. На основе полученной информации создаются модули устройства (МУ), отвечающие за взаимодействие с аппаратными средствами. Следует отметить, что в большинстве случаев одному устройству соответствует один МУ, однако при необходимости их количество может быть больше. Например, для каждого ядра центрального процессора создаётся отдельный МУ. При реализации dll-библиотек необходимо учитывать эти особенности.

После создания МУ для каждого из них, используя метод «Score» соответствующей библиотеки, вычисляются оценки производительности для всех типов задач поддерживаемых программной оболочкой. В указанный метод передаются такие характеристики МУ как: объём памяти, количество и частота ядер, а также дополнительные параметры характерные данному устройству.

Оценка используется при балансировке нагрузки между параллельно работающими МУ. Расчёт оптимального значения крайне важен для достижения максимальной производительности расчётов [3, 4].

Отправка ПЗ на решение ПОР осуществляется с помощью конфигурационного файла, содержащего информацию о типе задачи и её условия, соответствующие данному типу. Программная оболочка считывает файл и для каждого типа устройства, используя метод «Parser», формирует набор данных задачи, формат и содержание которых может отличаться для различных аппаратных средств.

После инициализации задачи ПОР последовательно отправляет её на решение свободным МУ при этом, на основе оценки производительности устройства с помощью метода «Cute» формируются данные подзадачи для этого МУ. Следует отметить, что вызов указанного метода осуществляется в основном потоке работы приложения. В этой связи необходимо максимально уменьшить время обработки данного метода, чтобы не допустить значительного простоя свободных МУ, ожидающих получения подзадачи от ПОР. С этой целью можно перенести часть вычислений в метод «SimSim», который выполняет решение подзадачи на МУ параллельно.

Алгоритм, реализованный в методе «SimSim» должен использовать особенности аппаратного средства для повышения эффективности расчётов. Так при решении задачи восстановления криптографических ключей в среде ПОР производительность алгоритма для видеокарт компании NVIDIA, используя технологию CUDA, в 500 раз выше по сравнению с центральным процессором [2].

ПОР отслеживает работу МУ. Если один из них не смог решить отправленную ему подзадачу, например, при отказе оборудования, то она передаётся на решение другим МУ.

В случае нахождения результата решения ПЗ метод «SimSim» формирует соответствующие данные, которые ПОР отправляет пользователю.

Разработанный подход выполнения прикладных решений в интегрированной программной среде оболочки распараллеливания можно считать универсальным. Это означает, что практически любую задачу пользователя, требующую для повышения

производительности использования распараллеливания вычислений, можно решить в этой программной среде, реализовав необходимые библиотеки. ПОР является автоматизированной программной системой, которая использует весь потенциал подключенных к компьютеру вычислительных устройств, допускающих распараллеливание.

1. Камерон Х., Трейси. Х. Параллельное и распределенное программирование с использованием C++ — М: Вильямс. — 2004. — 672 с.
2. Егоров А. Н., Марлей В. Е., Кузнецов В. А., Назаргулов И. А. Модель распараллеливания вычислений для повышения эффективности восстановления доступа к данным в корпоративных сетях // Труды СПИИРАН. — 2015. — № 5 (42). — С. 37–55.
3. Metaweia M. A., Ghoneim S. A., Haggag S. M., Nassar S. M. Load balancing in distributed multi-agent computing systems // Ain Shams Engineering Journal. — 2012. — Vol. 3. — Is. 3. — Pp. 237–249.
4. Daryapurkar A., Deshmukh M. V. M. Efficient Load Balancing Algorithm in Cloud Environment // International Journal Of Computer Science And Applications. — 2013. — Vol. 6. — No. 2. — Pp. 308–312.

Носков К.М.

Тенденции и развитие нейросетевых технологий в диагностике сложных систем

*Нижегородский Государственный Технический Университет им. Р.Е. Алексеева
(Россия, Нижний Новгород)*

doi: 10.18411/lj-31-08-2017-38

idsp: 000001:lj-31-08-2017-38

Аннотация

Рассматривается вопрос диагностики сложных систем. Современные нейронные сети позволяют решать широкий круг задач. Наибольшее преимущество нейросетевые технологии получают в условиях априорной неопределенности, которая свойственна задачам диагностики трудноформализуемых систем.

Ключевые слова: нейронные сети, диагностика сложных систем

При создании сложных систем возникает необходимость их диагностики. Кроме общего состояния системы важно учитывать компоненты, из которых она состоит. Для решения задачи диагностики разрабатывают математические модели исследуемых систем, но в большинстве случаев такая задача является труднорешаемой из-за огромного количества нечетких параметров сложных систем и сложностью выявления закономерностей. Примером такой задачи может служить актуальная в настоящее время задача классификации биоценозов, решаемая совместно с ФБУН ННИИЭМ им. академика И.Н. Блохиной.

Искусственные нейронные сети представляют собой эффективный и гибкий метод имитации процессов и явлений и способны решать подобные задачи. Искусственная нейронная сеть подбирает скрытые внутренние параметры как отдельных нейронов, так и всей сети в целом, а также выделяет наиболее актуальные входные признаки и отбрасывает малоинформативные, некорректные признаки. Эта особенность позволяет нейронным сетям исправно работать с большим количеством неинформативных и избыточных данных.

Используя классические методы обработки данных иногда требуется запрограммировать вручную некоторые модели в символьном виде. Нейронная сеть конструируется указанием общей структуры и количеством входящих в ней элементов, а для её обучения необходимо предоставить достаточное количество тренировочных примеров. При наличии экспертных знаний по заданной задаче или по специальным требованиям нейронные сети позволяют подбирать нужный тип нейронов, структуру сети и задавать определенные характеристики.