

Список литературы

1. СМС рассылка 2014 [Электронный ресурс]. – Режим доступа: <http://www.smsdelivery.ru>.
2. SMS ЦЕНР [Электронный ресурс]. – Режим доступа: <http://smsc.ru>.
3. SMS Aero [Электронный ресурс]. – Режим доступа: <http://smsaero.ru>.
4. MainSms [Электронный ресурс]. – Режим доступа: <http://mainsms.ru>.
5. Little SMS [Электронный ресурс]. – Режим доступа: <http://littlesms.ru>.

**МУЛЬТИАГЕНТНАЯ СИСТЕМА.
ВЗАИМОДЕЙСТВИЕ АГЕНТА-СОБИРАТЕЛЯ
С БАЗОЙ ДАННЫХ**

Пестряев А.А., Воронова Л.И.

*Московский технический университет связи информатики,
Москва, e-mail: aleksandrpestr@rambler.ru*

В настоящее время в интернете появляется все больше форумов, социальных сетей и блогов. В них люди общаются, решают различные проблемы. Эти данные могут содержать информацию, которая неприемлема для детей и подростков или была бы интересна спецслужбам.

В этой связи активно развиваются технологии модерации пользовательских постов. В том числе поиск таких нарушений как, распространение спама, флуда, размещение постов и комментариев не по теме, но существуют и более серьезные нарушения, например, такие как разжигание межнациональной розни, обсуждение и организация антиобщественных мероприятий, пропаганда насилия и т.д. В связи с неуклонным ростом размещаемой пользователями информации выполнять модерацию в основном человеческими силами становится все труднее. Для решения этой проблемы существует ряд разработок как зарубежных, так и отечественных [1-3].

В рамках магистерской диссертации «Разработка мультиагентной системы (МАС) для сбора текстовой информации в социальной сети» [4] автором ведется работа по проектированию и реализации МАС, с учетом многоядерной архитектуры компьютера и использования КЭШ памяти. Мультиагентная система анализирует объем КЭШ памяти и подстраивается под возможности компьютера, а если система имеет несколько ядер, то разные агенты смогут работать на раздельных ядрах. Это совершенно новый подход в разработке системы, который ранее не применялся.

Наиболее распространенное определение МАС – это система, образованная несколькими взаимодействующими интеллектуальными агентами. Мультиагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента [5]. При этом важно, что интеллектуальный агент – программа, самостоятельно выполняющая задание, в течение длительных промежутков времени и взаимодействующая с другими агентами.

Применение агентного подхода значительно упрощает разработку программного обеспечения (ПО), поскольку новые агенты могут использовать в своей работе других агентов, путем наследования их функций и свойств или же просто посылая им свои запросы, а также упрощается процесс размещения ПО в условиях сети. Происходит это за счет автоматизации процессов перемещения программного кода, его установки и конфигурирования [6].

Разработанная автором модель МАС включает следующие компоненты.

Агенты: агент-собирающий, агент-исследователь, агент-диагностик, агент-очиститель; базы данных: временная БД, база знаний (БЗ), БД «грязных» стра-

ниц, на которых найдены слова или выражения из базы знаний.

Одним из главных агентов является поисковый робот, или агент-собирающий. Он посещает социальные сети, форумы и блоги, и найденные ссылки сохраняет во временную базу данных.

Временная база данных – это база, в которой хранятся ссылки собранные из интернета и подлежащие дальнейшей обработки другими агентами.

Оптимизация работы агента-собирающего и его взаимодействия с временной базой данных, использует многоядерность компьютера и КЭШ память. В статье на примере тестов показаны, особенности и «тонкие моменты», которые необходимо учитывать при разработке приложений, обрабатывающих большие объемы данных.

Для обеспечения терминологической однозначности ниже приведен ряд определений.

Кэш память – это сверхбыстрая память, являющейся буфером между контроллером системной памяти и процессором. В этом буфере сохраняются блоки данных, с которыми процессор работает в данный момент, тем самым значительно уменьшается количество обращений процессора к медленной системной памяти и увеличивая общую производительность процессора.

Контроллер памяти – цифровая схема, управляющая потоком данных к и от оперативной памяти.

Оперативная память – энергозависимая часть системы компьютерной памяти, в которой временно хранятся данные и команды, необходимые процессору для выполнения им операции. По быстроте действия оперативная память находится на втором месте после КЭШ памяти.

Простейшая схема взаимодействия процессора с КЭШ памятью и оперативной памятью представлена на рисунке 1.

Кэш память бывает первого, второго и третьего уровней. В мультиагентной системе используется только второй уровень. В КЭШ память второго уровня изначально передаются все данные, для обработки центральным процессором. На втором уровне из данных строятся цепочка инструкций, а на первом уровне «зеркально» строятся внутренние команды процессора, которые учитывают особенности процессора, регистры. В отличии от КЭШ памяти первого уровня, КЭШ память второго уровня для процессора имеет огромное значение, именно поэтому процессоры с наибольшим объемом КЭШа второго уровня показывают высокую производительность.



Рис. 1. Схема взаимодействия процессора с энергозависимой памятью

Все данные загружаются в КЭШ второго уровня, именно поэтому мультиагентная система напрямую работает с ней. На рисунке 2 показано, как процессор обращается сначала к КЭШ памяти первого уровня, а затем к КЭШ памяти второго уровня, и если данные не найдены, то к оперативной памяти.

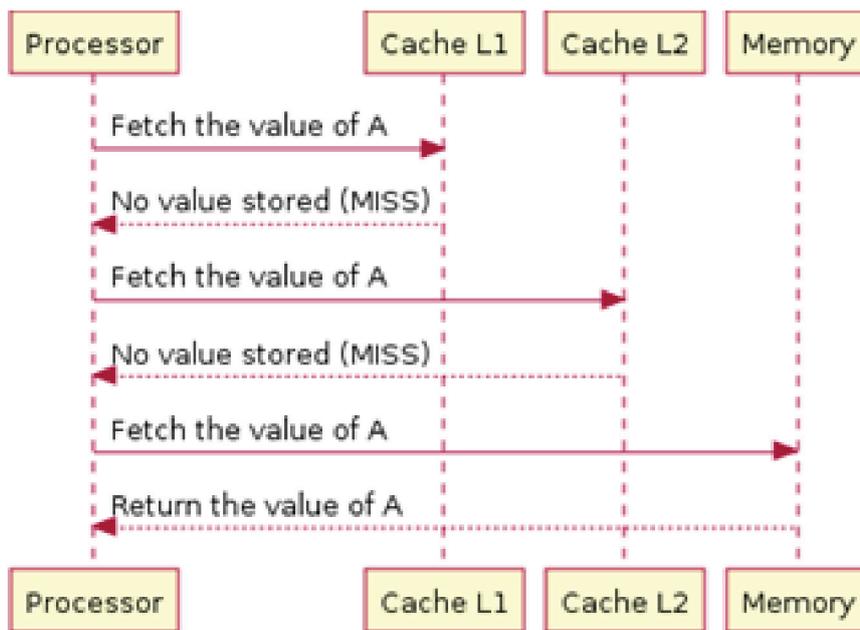


Рис. 2. Обращение процессора к КЭШ памяти

Впервые КЭШ памяти второго уровня появилась на процессорах i80486 и была равна 512 килобайт. На более современных компьютерах размер КЭШа больше, но даже при размере в 512 Кбайт процессор сможет обработать достаточно много информации.

Ниже приведены тесты на компьютере, который имеет 3 ядра и каждое ядро имеет 512 килобайт КЭШ памяти второго уровня.

Размер одного текстового символа равен 1 байт, 512 килобайт это 524288 байта. В рамках разработки мультиагентной системы было проведено исследование среднего размера ссылки в интернете, и выявлено, что средний размер равен 55 символов. Из вышеизложенного следует, что для загрузки целой таблицы в КЭШ память ее размер не должен превышать 9000 записей, но для поднятия быстродействия

разрабатываемой системы агент-собирающий будет обрабатывать несколько таблиц одновременно в разных потоках, поэтому размер таблицы будет ограничен 1000 записями.

Рассмотрим тесты, которые показывают, что если в 100 таблиц по 1000 записей вносить изменения в несколько потоков, то это окажется намного быстрее, чем работа с одной таблицей на 100000 записей.

На рис. 3 показана схема теста, в котором, агент, используя один поток, взаимодействует с базой данных, содержащей одну таблицу на 100000 записей. Агент производит запись 100000 строк по 55 символ, так как средний размер ссылки равен этому значению.

После работы агента появляется сообщение о количестве секунд затраченных на запись (рис. 4). На рис. 5 показана степень загрузки процессора (в %).

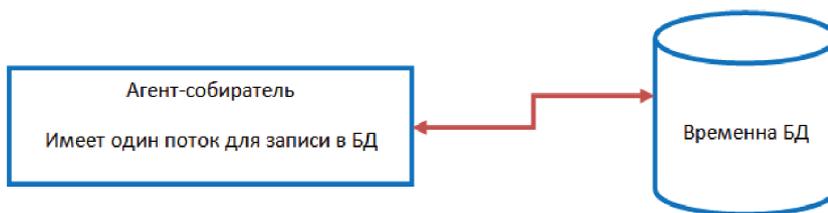


Рис. 3. Взаимодействие агента-собирающего с временной БД

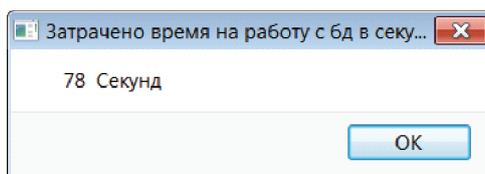


Рис. 4. Затраченное время на запись

Хронология загрузки ЦП

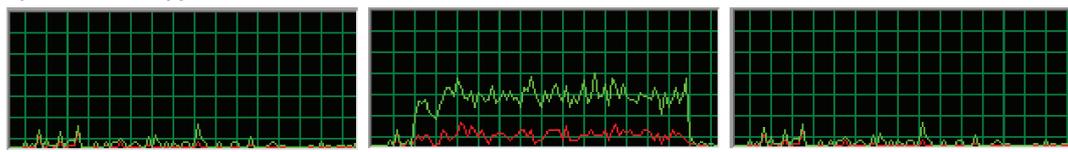


Рис. 5 Загрузка процессора

Следующий тест покажет, что если в БД создано 100 таблиц по 1000 записей, то это намного эффективнее работает, так как есть возможность запустить несколько потоков. На рис. 6 показана схема взаимодействия агента-сборателя с временной БД.

Агент имеет 6 поток для взаимодействия с БД. Каждый поток работает только со своей таблицей. Запускается 6 поток, так как это оптимально для данного компьютера. Результаты работы представлены на рис. 7 и 8.

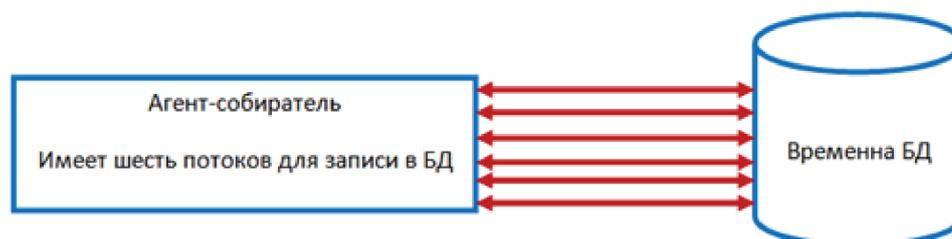


Рис.6. Взаимодействие агента-сборателя с временной БД

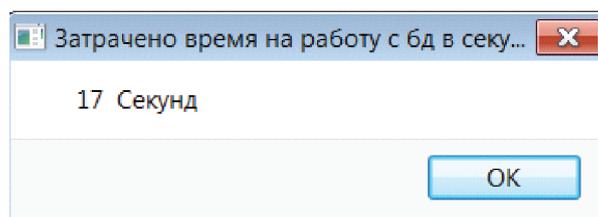


Рис. 7 Время, затраченное на запись

Хронология загрузки ЦП

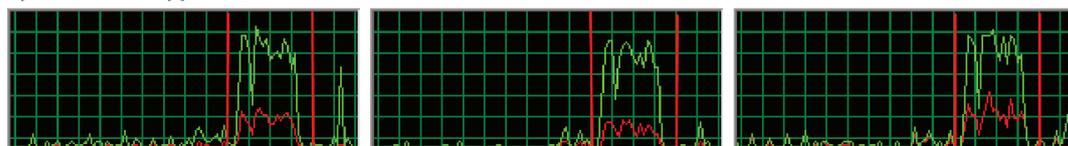


Рис. 8. Загрузка процессора

Из рис. 4 и 7 сразу видна разница в скорости выполнения записи, в 4.5 раза быстрее происходит запись в «малые» таблицы, чем в одну большую. Если посмотреть на рисунок 5, то у второго ядра загрузка, в среднем, составляет 45 процентов, это немного для данного компьютера. Однако, если сравнить данные на рис.8, то у всех 3 ядер, во время записи в таблицы, загрузка 90 процентов, что говорит об оптимальном выборе количества потоков и размере таблиц для данного компьютера.

В разрабатываемой мультиагентной системе во временной базе данных будет создано много таблиц по 1000 записей, для того, чтобы разные потоки могли работать с разными таблицами, тем самым ускоряя работу системы.

Выводы

При разработке приложения, обрабатывающего большие объемы данных, необходимо учитывать составляющие компьютера, в частности КЭШ память и количество ядер. Учитывая данные параметры, можно оптимизировать работу программы в 4-5 раз, что даст возможность обработать больше информации за меньший промежуток времени.

Список литературы

1. CleanTalk – Защита от спама для блогов и форумов [Электронный ресурс] – Режим доступа <http://cleantalk.ru>.
2. Comment E-Mail Verification – WordPress > Comment E-Mail Verification « WordPress Plugins [Электронный ресурс] – Режим доступа <http://wordpress.org/extend/plugins/comment-email-verify>.
3. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. – 877 с.: ил.
4. Пестряев А.А., Воронова Л.И. Анализ поисковых роботов и выбор функций для своего робота // Материалы V Международ-

ной студенческой электронной научной конференции «Студенческий научный форум» [Электронный ресурс] – Режим доступа www.scienceforum.ru/2013/183/2549 (дата обращения: 23.01.2014).

5. Википедия мультиагентная система – <http://ru.wikipedia.org/>
 6. Трунов А.С., Воронова Л.И., Воронов В.И. Разработка методов распределения для высокопроизводительных вычислений в многочастичных системах. – Международный журнал прикладных и фундаментальных исследований. 2013. № 10-2. С. 192-194.

ПАРАЛЛЕЛЬНЫЙ РАСЧЕТ СИСТЕМЫ N-ЧАСТИЦ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ MPI И CUDA

Пилипчак П.Е., Трунов А.С.

Филиал Российского государственного гуманитарного университета, Домодедово, Московская область, e-mail: thelastrainbow@yandex.ru

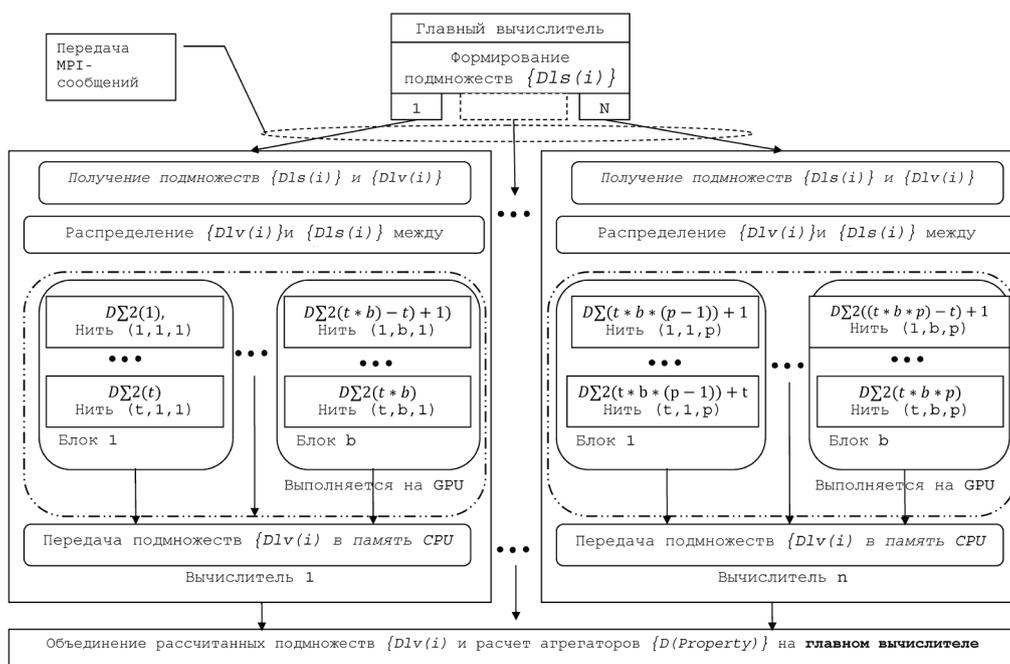
В настоящее время существует множество программных комплексов позволяющих решать задачи компьютерного моделирования с применением высокопроизводительных вычислений. Большинство таких комплексов решают задачи по моделированию систем относящихся либо к классу с дальнедействующими потенциалами, либо учитывающими только короткодействующие ковалентные взаимодействия [1-3].

Принципиальное отличие систем, являющихся предметом исследования данной работы, состоит в том, что это полимеризующиеся системы с многоча-

стичным взаимодействием, объединяющим несколько видов взаимодействий: двухчастичные вклады (дальнедействующие ионные и близкодействующие отталкивательные) и многочастичные (двух и трехчастичные ковалентные взаимодействия)[6-8]. Моделирование полимеризующихся систем является нетривиальной задачей, требующей учета особенностей ионного и ионно-ковалентного взаимодействия частиц разных типов в расплаве. Это существенным образом усложняет постановку задачи распределения.

На основе концептуальной модели и тщательного анализа программного кода локального МД-приложения [6], автором разработана модель неоднородных дескрипторов, обеспечивающая возможность распределения и распараллеливания расчетов для коррелированных многочастичных систем, на основе «свертки» детализации физического описания объектов [5].

Предметом данной работы является разработка метода позволяющего применять вычислители на основе графических процессоров для расчетов системы N-частиц, что даст возможность распределить расчет дескрипторов между миллионами нитей и серьезно сократить время КМ. На рисунке представлен алгоритм расчета дескрипторов с использованием центрального и графического процессора.



Алгоритм расчета дескрипторов с использованием центрального и графического процессора

Передание подмножеств дескрипторов каждому вычислителю производится с использованием технологии MPI, а параллельный расчет реализуется на графических процессорах с применением технологии CUDA.

На главном вычислителе формируются подмножества дескрипторов {Dls(i)}, {Dlv(i)} с мощностью $k=N/p$, где N – количество дескрипторов, p – количество вычислителей выполняющих расчет. Сформированные подмножества передаются вычислителям через интерфейс передачи сообщений MPI. Полученные подмножества дескрипторов распределяются центральным процессором между нитями, реализующими расчет на графическом процессоре, где индекс t – количество нитей в блоке, b – количество блоков. На каждой нити графического процессора, с использо-

ванием технологии CUDA, рассчитываются значения элементов одного двухчастичного агрегатора $D\Sigma 2(i)$. На основе результатов расчета подмножеств двухчастичных агрегаторов обновляются значения элементов дескрипторов {Dlv(i)}. Полученные подмножества дескрипторов {Dlv(i)} передаются на центральный процессор. После завершения расчетов всеми вычислителями рассчитанные подмножества одночастичных векторных дескрипторов объединяются на главном вычислителе для расчета агрегатора {D(Property)}.

Проведенные компьютерные эксперименты, результаты которых представлены в таблице показывают существенное сокращения времени проведения КМ за счет совместного использования технологий CUDA и MPI.