

АНАЛИЗ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Статья посвящена анализу данных мониторинга типичного отдела научно исследовательского института, в результате которого выявлены закономерности работы персональных компьютеров (ПК) сотрудников и произведена оценка количества свободных ресурсов, которые можно задействовать для высоконагруженных научных вычислений.

Введение

Современные задачи физики высоких энергий, геномной инженерии, космологии и других наук требуют все больших вычислительных ресурсов [1, 2]. Для удовлетворения этих потребностей ученых бурно развиваются грид-технологии [3], на что тратятся огромные средства и человеческие ресурсы.

Вместе с тем, каждый научный сотрудник имеет у себя на рабочем месте персональный компьютер (ПК), зачастую с неплохими характеристиками. Возникает вопрос: насколько рационально используются ресурсы ПК? Ведь неполную загрузку можно расценивать как потерю ценных ресурсо-часов. Мы считаем, что человечество накопило огромные вычислительные ресурсы именно в секторе ПК и вопрос об их использовании для вычислений до сих пор остается открытым.

Для проверки данного предположения, в представляемой работе сделана попытка оценить ресурсы (и уровень их использования) накопленного парка ПК, которые можно объединять в единую инфраструктуру для решения различных вычислительных задач.

С технической точки зрения в качестве инфраструктуры может выступать многопользовательская вычислительная система, которая разворачивается на базе локальных и глобальных гетерогенных (неоднородных) компьютерных сетей.

Под неоднородной сетью понимается сеть произвольной топологии с различными каналами связи, состоящая из компьютеров различной конфигурации и мощности, на которых установлены различные операционные системы.

Вычислительная сеть предназначена для предоставления сервисов по обработке и хранению данных многих пользователей. Мы выбрали агентный подход и сервис-ориентированную архитектуру при разработке и моделировании такой системы. Поэтому исследуемая система далее именуется агентной вычислительной системой или сокращенно АВС.

Такая архитектура была выбрана с целью обеспечения:

- масштабируемости системы – компоненты системы взаимодействуют друг с другом удаленно, без необходимости знать месторасположение друг друга в системе (что собственно не нужно и только усложняет масштабирование – следовательно это лишнее), а исходя из их возможностей (доступных ресурсов, например);

- отказоустойчивости системы – система должна сохранять работоспособность даже при выходе из лада или отключения любого ее компонента;

- адаптивность (например, система может решать запускать только по одной копии задач или по две. это может сильно влиять на скорость вычислений при различных условиях – один вариант может быть лучше при высокой вероятности выключения любой из машин, другой – при не высокой).

Таким образом, задача утилизации вычислительных мощностей персональных компьютеров является крайне актуальной. Для её решения необходима разработка гибких, сервис-ориентированных, самонастраивающихся, отказоустойчивых программных систем, которые позволили бы

осуществлять менеджмент ресурсов персональной компьютерной техники с целью их использования в качестве вычислительных инструментов в том числе и для решения больших и сложных задач.

Такие попытки были сделаны достаточно давно, однако в основном нацелены на решение какой-то конкретной задачи [4, 5]. Особенностью подобных систем является то, что участники, которые передали в общее пользование свой персональный компьютер, не решают свои задачи. Они скачивают и устанавливают себе на компьютер программный модуль (одинаковый для всех), который получает от главного сервера определенную порцию данных для расчета и по окончании вычислений возвращает серверу результат. Таким образом, подобные системы нельзя назвать многопользовательскими и они не пригодны для решения поставленной задачи.

Подсистема мониторинга

Для проведения мониторинга ресурсов сети ПК была использована разрабатываемая нами мультиагентная система для распределенных вычислений. Идея системы состоит в следующем. На каждом компьютере, подключенном к АВС, при запуске программы будет создан контейнер, как ограниченное жизненное пространство нашей системы на конкретном компьютере. В этих контейнерах будут «жить» агенты, выполняющие разные задачи согласно своей роли. С нашей точки зрения наиболее подходящей платформой для реализации АВС является фреймворк JADE [6, 7]. JADE – высокоуровневая среда для разработки мультиагентных систем (МАС) и в ней уже реализованы основные аспекты, необходимые для существования МАС, такие как, например, менеджмент агентов или коммуникация между агентами. Поэтому разработчик, выбрав за основу платформу JADE, может сосредоточиться больше над разработкой логики жизни МАС, чем над техническими тонкостями её реализации.

Основные сущности в JADE – это агенты и контейнеры. Каждый агент существует в рамках определенного контейне-

ра, а каждый контейнер – это отдельный процесс Java. Именно контейнер обеспечивает агенту доступ к сервисам, как к стандартным, так и к дополнительным – созданным разработчиками МАС на основе JADE. В свою очередь сервисы предоставляют агентам и другим сервисам функционал распределенной платформы.

Агенты используют функционал, который предоставляется сервисами (например, общение агентов происходит благодаря сервису передачи сообщений). Изначально платформа JADE уже содержит набор специальных сервисов и агентов, необходимых для функционирования распределенной МАС.

Следует отметить, что JADE предоставляет разработчикам специальный механизм для интеграции в систему собственных сервисов. JADE избавляет разработчика сервиса от реализации взаимодействия распределенных составляющих сервиса, что существенно облегчает решение вопроса масштабируемости системы. Гибкость интеграции сервисов в систему обеспечивает так называемая Архитектура распределенных скоординированных фильтров платформы. Эта архитектура позволяет сервисам реагировать на основные события платформы, а также в случае необходимости агенты могут напрямую взаимодействовать с сервисами.

Архитектура АВС

Архитектура разрабатываемой нами АВС сервисно-ориентированная. В частности, как один из сервисов, для сбора данных о наличии и использовании ресурсов была разработана и использовалась подсистема мониторинга. Она представляет собой совокупность специально разработанного нами сервиса ядра и набора программных агентов, которые собирают данные с машин, на которых они установлены. Эти данные агенты передают сервису мониторинга для дальнейшей обработки и хранения. Как планируется, в дальнейшем эти данные будут предоставляться по требованию другим сервисам или агентам.

Кроме того, особенность разрабатываемой АВС состоит в том, что в агенты обладают мобильностью и могут переме-

щаться от контейнера к контейнеру. За вычисление каждой задачи отвечает отдельный агент именуемый TaskAgent-ом. Это мобильный, целенаправленный агент, способный перемещаться между машинами системы и перемещать за собой код своей задачи. Цель этого агента – провести задачу из состояния "запланирована" в состояние "посчитана" за минимально возможное время (tcalc min). Первое состояние – задача и ее входные данные созданы и она готова к вычислениям, второе – есть валидный результат вычислений задачи. Причем этот агент отвечает не только за то, чтобы найти машину, на которой в определенный момент есть необходимое количество доступных вычислительных ресурсов, но и сам управляет процессом вычисления его задачи, а не делегирует эту роль каким-то другим компонентам системы. Это возможно благодаря тому, что все задачи, которые отправляются в систему, должны соответствовать заданному нами стандарту.

Таким образом, агенты путем переговоров делят доступные ресурсы и используют их для вычисления задач. Система состоит из контейнеров, в которых "живут" агенты. На каждой машине запущено ровно по одному контейнеру. Контейнер представляет собой отдельный процесс.

Многоуровневая архитектура системы показана на рис. 1.

Как видно из рисунка, она представляет собой иерархическую пятиуровневую систему. Самый базовый уровень – Аппаратный, который состоит из компьютеров и связывающего их сетевого оборудования. Второй уровень – Транспортный, обеспечивающий логическую связь между компьютерами. Третий уровень служит базой для построения агентной вычислительной сети. Им может быть любой фреймворк, который дает возможность работать с такими сущностями, как агенты и обеспечивать общение между ними. Следующий уровень системы содержит логику работы агентов, обеспечивает основные сервисы, такие как решение задач, хранение данных, интерфейсы пользователя, авторизацию и т. п. Разработка и выявление, а затем и оптимизация с помощью

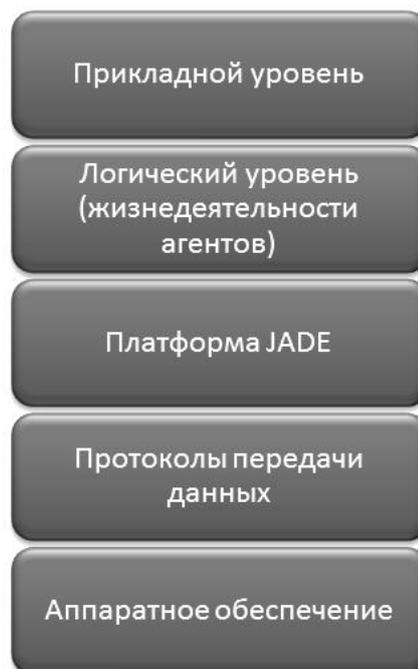


Рис. 1. Архитектура вычислительной системы

моделирования, основных алгоритмов работы агентов этого уровня как раз и является целью данного исследования. Последний, пятый уровень, является сервисным с точки зрения потребителя (пользователя) и дает возможность получить услуги по расчету необходимых задач.

Техническая реализация системы

В нашем случае менеджментом контейнеров и агентов полностью занимается платформа JADE, выступающая отдельным слоем в архитектуре (см. рис. 1).

Реализация агентов

Технически агент реализован как отдельный поток, к нему нет прямого доступа, но на него могут влиять агенты (но только путем отправки ему сообщений) и окружающий его мир – например, настройки, ограничения или доступные свободные ресурсы машины, на которой он находится в данный момент. Составной частью каждого агента есть специальные компоненты, инкапсулирующие в себе и предоставляющие агенту логику взаимодействия с системой. Структура агента с учетом этих компонентов показана на рис. 2.



Рис. 2. Компоненты агента

Такими компонентами являются: AgentsManager – менеджер агентов, PlatformManager – менеджер платформы. ServicesManager – менеджер сервисов. Менеджер агентов предоставляет агенту возможность создавать или удалять других агентов (но удалять он может только созданных им агентов). Менеджер сервисов позволяет агенту регистрировать в системе свои сервисы, причем агент может регистрировать и deregистировать свои сервисы в любой момент работы системы.

Стоит заметить, что агент может выполнять все действия, предоставляемые менеджерами и без их помощи, напрямую отправляя сообщения агентам, ответственным за менеджмент агентной платформы. Но выделение этих логических компонентов в архитектуре системы упрощает ее разработку и поддержку. Тем более если учитывать, что эти менеджеры доступны всей иерархии агентов.

Обмен агентов сообщениями

Агенты взаимодействуют друг с другом исключительно посредством отправки сообщений друг другу. Они обмениваются сообщениями посредством протокола MTP (message transport protocol), который передает данные поверх популярных протоколов передачи данных таких как tcp/ip, bluetooth и т. д. Все сообщения агентов – сообщения стандарта FIPA-ACL (agent communication language). За переда-

чу сообщений между агентами отвечает сервис передачи сообщений (MTS или message transport service), работающий по стандарту FIPA MTS. Сообщения, которыми обмениваются агенты, оформлены согласно их онтологиям. То есть представлены в виде структурированных данных, которые можно валидировать и извлекать знания из них (ontology and content languages).

Стоит отметить, что участником вычислительной сети может стать клиент, не обязательно использующий платформу JADE. Главное условие – реализация стандартов, по которым взаимодействуют компоненты системы (агенты).

Результаты мониторинга

Нами проводился мониторинг одного из сегментов локальной сети Института прикладной физики НАН Украины, в который входит одиннадцать компьютеров. Их характеристики приведены в табл. 1.

Таблица 1. Характеристики персональных компьютеров локальной сети

№ комп.	Тактовая частота, ГГц	Оперативная память, Мбайт
1	2 009	515
2	2 010	768
3	1 802	516
4	2 328	1030
5	2 933	758
6	1 808	449
7	1 600	3123
8	2 202	515
9	1 808	448
10	1 596	3136
11	1 800	1032

На каждом компьютере был автоматически запущен агент мониторинга, который каждые 10 секунд передавал сервису мониторинга данные о загрузке процессора и памяти. Чтобы набрать достаточно данных для статистики мы наблюдали за исследуемой локальной сетью ровно неделю, чтобы охватить рабочие и выходные дни. Понятно, что в нерабочее время и в нерабочие дни машины выключены и простаивают, мониторинг показал нулевую загрузку машин.

На рис. 2 показан график загруженности (С, % от максимального) центрального процессора 1-го компьютера от времени (Т, сек), откуда видна степень неравномерности работы владельца, что отражает типичную ситуацию.

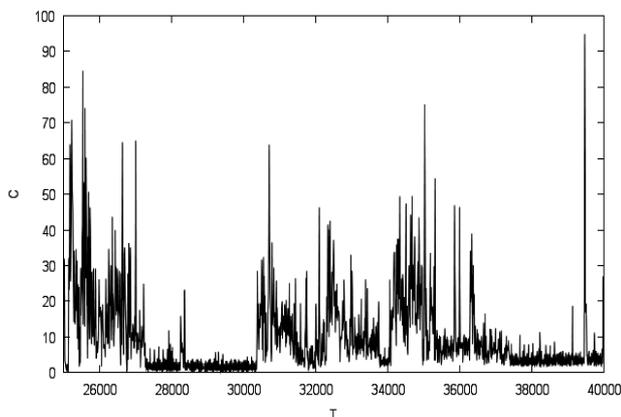


Рис. 2. Характерный график загруженности одного из персональных компьютеров

На рис. 3 представлено относительное время загрузки того же 1-го компьютера в другом виде. Здесь показано количество (К, ед) измерений, попавших в определенный интервал загрузки (Р, %), который разбит на 10 интервалов по 10 %.

Подобное распределение загруженности процессора за время работы пользователя наблюдалось для большинства компьютеров. Полученные данные позволяют определить среднюю загруженность первой машины во время работы пользователя $\eta = 18,80\%$. Однако, для расчета этой характеристики брался интервал времени, когда машины были включены, а интервалы, когда машины были выключены, не учитывались.

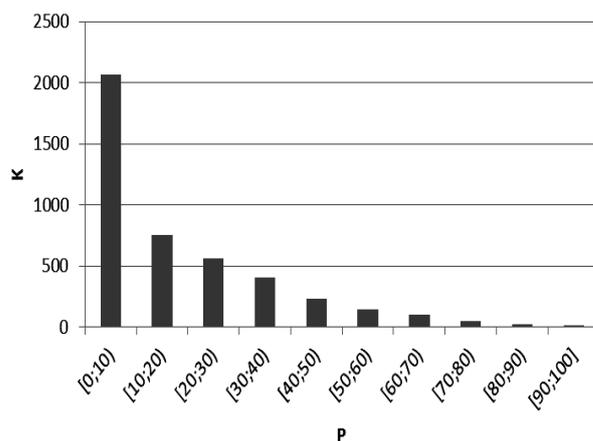


Рис. 3. Относительное время загрузки 1-го компьютера

Аналогичные результаты мониторинга по всем компьютерам сети приведены в табл. 2.

Таблица 2. Статистические характеристики загрузки компьютеров

№ компьютера	Средняя загруженность процессора включенного компьютера, %	Средняя загруженность процессора за все время, %
1	18,80	0,72
2	16,93	0,36
3	37,84	1,26
4	2,01	1,95
5	41,68	1,86
6	28,62	0,47
7	35,57	0,80
8	3,970	0,19
9	10,81	0,50
10	8,70	3,91
11	8,80	0,31
Среднее	19,43	1,12

Во второй колонке приведена средняя загруженность процессора компьютера

с учетом только того времени, когда он был включен. А в третьей колонке приведена средняя загрузка процессора за весь период наблюдения, включая и те моменты, когда компьютер был выключен (в момент, когда компьютер был выключен, считалось, что загрузка его процессора равна 0 %).

Таким образом, видно, что значительная часть ресурсов попросту простаивает и может быть использована для решения широкого спектра вычислительных задач.

Как видно из второй колонки табл. 2, относительно более «слабые» ПК сети загружены, как правило, в большей степени. Это означает, что ПК с такими характеристиками представляют меньший интерес для построения распределенной вычислительной сети не только с точки зрения своих относительно малых ресурсов, а еще с точки зрения большей загрузки.

Перспективы

В наши дни для решения вычислительных задач практически в каждом исследовательском институте строятся вычислительные кластеры, которые в перспективе объединяются в грид-ассоциации. Для построения кластера нужно дорогостоящее серверное оборудование, квалифицированные кадры, серверное помещение со специальными системами кондиционирования, электропитания, оповещения о недопустимых событиях (например, резкое повышение температуры из-за выхода из строя системы кондиционирования). В то время, как на постройку и обслуживание кластера тратятся достаточно большие средства и кадровые ресурсы, уже имеющиеся мощности ПК загружены в среднем примерно на 1 %. Причем, что очень важно, персональные компьютеры сотрудников не нуждаются в специализированном обслуживании, они не сконцентрированы в пределах одного помещения и не требуют отбора тепла, замена и ремонт стоят значительно дешевле и т.п. Например, Институт прикладной физики НАН Украины имеет кластер из 10 двухпроцессорных четырехядерных узлов,

что позволяет одновременно решать 80 независимых задач. Но на балансе самого института находится около 150 персональных компьютеров общей производительностью превышающей возможности вычислительного кластера.

На основе данных мониторинга можно с уверенностью утверждать, что на базе персональных компьютеров возможно построение целой инфраструктуры для решения достаточно больших вычислительных задач как научного, так и производственного плана. Но к такой системе с другой стороны выдвигаются дополнительные требования. К ним можно отнести адаптацию и сохранение работоспособности в условиях динамического изменения доступных ресурсов: во-первых, система должна минимально влиять на удобство работы владельца машины, во-вторых, любой персональный компьютер в сети может быть выключен в любой момент и это не должно привести к краху системы.

Эти требования мы учитываем при разработке нами системы для распределенных вычислений на базе ПК.

Результаты

С помощью сервиса мониторинга распределенной МАС, разрабатываемой на основе платформы JADE, получены данные об использовании ресурсов парка ПК типичного сегмента локальной сети академического института. Результаты позволяют с уверенностью утверждать, что на имеющихся ресурсах может быть построена вычислительная распределенная система, для решения достаточно больших вычислительных задач. Такая система должна быть сервисно-ориентированной, устойчивой к сильному изменению ресурсов, эффективной и при этом не затруднять работу владельцев компьютеров.

1. *LHC_Homepage* [Электронный ресурс]. Режим доступа: <http://lhc.web.cern.ch/lhc/> – Заголовок с экрана.
2. *Rosetta@home* [Электронный ресурс]. Режим доступа: <http://boinc.bakerlab.org/rosetta/> – Заголовок с экрана.

3. *Open Grid Forum* [Електронний ресурс]. Режим доступу: <http://www.ogf.org/> – Заголовок с екрана.
4. *SETI@home - Search for ExtraTerrestrial Intelligence at home* [Електронний ресурс]. Режим доступу: <http://setiathome.berkeley.edu/>. – Заголовок с екрана.
5. *Rosetta@home - Protein Folding* [Електронний ресурс]. Режим доступу: <http://boinc.bakerlab.org/rosetta/>. – Заголовок с екрана.
6. *Jade - Java Agent DEvelopment Framework* [Електронний ресурс]. Режим доступу: <http://jade.tilab.com/> – Заголовок с екрана.
7. *Bellifemine*, Fabio. Developing multi-agent systems with JADE / Fabio Bellifemine, Giovanni Caire, Dominic Greenwood – John Wiley & Sons, Ltd – 2007.

Получено 09.04.2015

Об авторах:

Лопаткин Роман Юрьевич,
кандидат физико-математических наук,
доцент кафедры общей
и экспериментальной физики
Сумского государственного университета,

заведующий Научно-исследовательским
центром учебно-научных приборов
Института прикладной физики НАН Украины,

Иващенко Виталий Анатольевич,
младший научный сотрудник
Научно-исследовательского центра
учебно-научных приборов Института
прикладной физики НАН Украины,

Игнатенко Сергей Николаевич,
младший научный сотрудник,

Куприенко Виктор Викторович,
младший научный сотрудник.

Место работы авторов:

Институт прикладной физики
НАН Украины,
Украина, г. Сумы,
ул. Петропавловская, 58.
Тел.: (0542) 604538.

E-mail: lopatkin@iap.sumy.org,
va.ivashchenko@gmail.com,
mynameissergey@gmail.com,
k_vic@mail.ru