

**МОДЕЛИРОВАНИЕ ДЕЦЕНТРАЛИЗОВАННОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ****Р. Ю. Лопаткин, В. А. Иващенко, В. В. Куприенко (Сумы, Украина)**

На сегодняшний день эффективное использование накопленных человечеством вычислительных ресурсов является достаточно актуальной задачей. Достаточно напомнить, что, например, для накопления и обработки данных, полученных с большого адронного коллайдера, создается специальная грид-инфраструктура [1]: строятся серверные помещения с дорогими системами кондиционирования, устанавливаются кластеры, затрачивается огромное количество денежных и человеческих ресурсов. И это на фоне того, что персональные компьютеры пользователей, суммарная вычислительная мощность которых вполне достаточна для выполнения подобных задач, в среднем работают по 3 ч в сутки, а остальное время простаивают.

Попытки вовлечь в научные расчеты персональные компьютеры сторонних пользователей предпринимались неоднократно [2, 3] и, по всей видимости, имели некоторый успех. Однако стоит заметить, что такие проекты очень узкоспециализированны и требуют написания под решаемую задачу специализированного программного обеспечения. Еще один недостаток такого рода систем заключается в полной централизации, которая заложена в их архитектуре – управляющие компьютеры раздают задачи вычислителям, а затем собирают и компонуют результаты. Одним словом, такие системы не могут претендовать на роль универсальных средств для эффективного использования вычислительных ресурсов сторонних пользователей, а строящиеся грид-инфраструктуры на это просто не нацелены.

Таким образом, с нашей точки зрения, задача менеджмента распределенных вычислительных ресурсов является очень актуальной и требует системного подхода. Такая система должна быть сервисориентированной, нацеленной на эффективное использование динамически меняющихся распределенных вычислительных мощностей, обладать достаточной пропускной способностью пользовательских задач, иметь высокую степень устойчивости, быть максимально простой в установке и управлении или вообще не требовать вмешательства администратора.

Основная идея, положенная нами в разработку такой системы, состоит в её полной децентрализации. По нашему мнению, изначально децентрализованная система сможет обеспечить лучшую устойчивость по сравнению с полностью централизованным аналогом или системой, где центры управления сервисами указываются при установке. Однако полностью децентрализованные системы требуют для своего функционирования более сложных алгоритмов, которые связаны с коллективным принятием решений. Перед разработкой децентрализованной вычислительной сети (ВС) для проверки алгоритмов её функционирования мы проводили моделирование системы, что и является предметом изложения данной публикации.

Для решения поставленной задачи нами применялось агентное моделирование системы, основные принципы которой можно сформулировать так:

- каждый элемент системы представлен соответствующим агентом, под которыми понимаются: компьютер с установленным на него специальным программным обеспечением, канал связи, пользователь и т.п.;
- каждый агент может играть в системе одну или несколько ролей, например, заниматься мониторингом, вычислениями, хранением или авторизацией;
- агенты всегда проявляют инициативу, а не ждут управляющих команд;
- в основу положены алгоритмы коллективного принятия решений;
- система сама принимает решение о перераспределении имеющихся ресурсов и ролей, что существенно упрощает процесс её администрирования.

Для исследования рассматриваемой системы нами была разработана ее мультиагентная модель в среде моделирования Repast Symphony [4]. Модель создавалась с учетом того, что каждый агент имеет одинаковую логику поведения и в каждый момент времени (в зависимости от предыдущего влияния окружающего мира и других агентов, а также деятельности самого агента) агент находится в состоянии, определяемом его ролью. Такая унификация существенно упрощает разворачивание системы – необходимо будет устанавливать одну версию программы на каждый компьютер ВС.

На первом этапе моделировалась однородная локальная сеть, в которой агенты однозначны по производительности и имеют равные каналы связи типа «каждый с каждым». Несмотря на такие, казалось бы, сильные упрощения, модели с такой топологией и характеристиками часто можно наблюдать в реальности, например, локальная односегментная сеть офиса или отдела института из нескольких десятков практически одинаковых компьютеров, подключенных к сети через сетевые коммутаторы. Как показали результаты моделирования, такую сеть можно эффективно использовать для построения ВС на основе коллективных алгоритмов.

Изначально в исследуемой ВС компьютерам не назначается конкретная роль, а распределение ролей регулируется квотами. На первом этапе вне зависимости от числа агентов в ВС происходит коллективный выбор претендента на роль «Мастера ролей» (MP) и его дублеров. Эта процедура проходит по алгоритму, схожему с алгоритмом выбора лидера [ссылка на алгоритм выбора лидера], и может повторяться при потере агента, выполняющего роль MP, или при возникновении коллизий в случае, если вследствие задержек в обмене информацией между агентами по каналам связи было выбрано несколько MP. Таким образом, в ВС всегда назначен MP, который обладает информацией о распределении агентов по ролям согласно установленным квотам, а каждый агент знает его сетевой адрес.

При подключении к ВС новый агент проявляет инициативу и запрашивает у первого найденного агента адрес MP. Далее новый агент направляет запрос к MP на получение роли и после ответа переходит к исполнению заданий согласно выданной роли.

Описанная выше логика коллективного поведения агентов позволяет системе оставаться работоспособной при случайных сбоях в каналах связи и физическом отключении агентов от ВС. Следует отметить, что подобные алгоритмы не применимы для выбора агентов-«исполнителей», таких как хранилище, вычислитель, сервис автоматизации и т.п. Поэтому децентрализацию ВС следует понимать в контексте динамически меняющихся со временем ресурсов и коллективного выбора управляющих агентов, которые в дальнейшем управляют поведением системы.

Исследование зависимости устойчивости системы от её размеров моделируется случайными равномерно распределенными (возможен и другой закон распределения) потоками. Задается входящий в систему поток вычислительных ресурсов, который означает, что к сети постоянно подключаются новые компьютеры, а также исходящий поток, моделирующий отключение компьютеров от системы. Из моделирования понятно, что при преобладании входящего потока система растет и на первый план выходит вопрос эффективности системы вследствие непропорционального роста накладных расходов на её управление. Если же мощнее исходящий поток, то система сжимается и в этом случае важным становится вопрос о её работоспособности. Логично предположить, что если один агент может выполнять больше одной роли, сжатие системы в конечном итоге приведет к тому, что все сервисы ВС будут поддерживаться одним агентом. Такая ситуация вряд ли позволяет рассматривать систему как вычислительный инструмент, но по крайней мере не ведет к её полному краху. На самом деле результаты моделирования говорят о том, что существует некоторая «критическая масса» ВС, при

которой она не может выполнять свои функции, и эта масса далека от единицы. По всей видимости, это связано с тем, что делегирования роли определенному из оставшихся агентов не происходит мгновенно, особенно если идет речь о роли хранилища, которая связана с перекачкой большого объема данных.

### Выводы

Разработана мультиагентная модель распределенной децентрализованной вычислительной сети, которая позволила отработать некоторые алгоритмы ее работы без создания самой системы. Исследовано поведение системы при её росте и сжатии, что дало возможность определить пределы применимости такой системы. В данный момент исследуются вопрос оптимальных квот на роли, при которых система покажет себя максимально эффективной.

### Литература

1. WLCG [Электронный ресурс]. Режим доступа: <http://lcg.web.cern.ch/lcg/>  
Заголовок с экрана
2. SETI@home [Электронный ресурс]. Режим доступа: <http://setiathome.berkeley.edu/>  
Заголовок с экрана
3. PrimeGrid [Электронный ресурс]. Режим доступа: <http://www.primegrid.com/>  
Заголовок с экрана
4. **Macal C. M.** Repast Symphony Runtime System / C.M. Macal, M.J. North, and D. Sallach (eds.) // Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms, ANL/DIS-06-1, cosponsored by Argonne National Laboratory and The University of Chicago, Oct. 13–15, 2005.