

ОБ ОДНОМ МЕТОДЕ МУЛЬТИАГЕНТНОЙ ОРГАНИЗАЦИИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ СЕТИ КОМПЬЮТЕРОВ ЧАСТНЫХ ПОЛЬЗОВАТЕЛЕЙ¹

В работе предлагается новый метод организации распределенных вычислений в облачных сервисах. Данный метод позволяет использовать вычислительные ресурсы частных компьютеров, объединенных с помощью сети, для решения связных задач. Основным преимуществом предлагаемого метода является возможность адаптивной корректировки вычислительного процесса в случае изменения параметров вычислительных узлов системы, что достигается за счет децентрализации системы. В работе так же предлагаются алгоритмы реализации децентрализованных облачных вычислений, базирующиеся на разработанном методе.

ONE METHOD FOR MULTIAGENT ORGANIZATION OF CLOUD COMPUTING BASED ON THE NETWORK OF PRIVATE USERS /
A.I. Kalyaev (MCS Institute of SFU, Taganrog, E-mail: anatoly.kaliaev@gmail.com. The new method of the organization of the distributed computations in a cloud computing systems is offered in the presented paper. The method allows using resources of private computers united by a network for solving coherent tasks. The main feature of the given method is adaptive correcting of process of computation if parameters of computing nodes are varying. This ability is achieved because of system decentralization. Also paper presents algorithms based on the developed method.

1. Введение

С самого начала компьютерной эры человечество использует вычислительную технику для решения сложных вычислительных задач. Именно необходимость решения подобных задач подтолкнуло человечество к автоматизации вычислений и созданию первых электронных вычислительных машин. Однако сложность задач росла и продолжает расти опережающими темпами. Изначально с такими вычислительными задачами могли справиться только суперкомпьютеры, громоздкие и дорогостоящие. Одним из первопроходцев в данной области стал Сеймур Крей со своим векторным суперкомпьютером «Cray-1», который стал одним из самых востребованных суперкомпьютеров в истории. [1] Затем ему на смену пришли параллельные векторные суперкомпьютеры, такие как «Cray X-MP», «Cray-2», а в 90-х появились и начали активно развиваться суперкомпьютеры с массовым параллелизмом, к которым можно отнести как Numerical Wind Tunnel от Fujitsu со 144 процессорами [2], так и суперкомпьютер ASCI Red, состоящий

¹ Работа выполнена при финансовой поддержке РФФИ (грант № 11-07-00463-а).

из нескольких тысяч процессоров, который первым преодолел рубеж пиковой производительности в 1 терафлоп в 1996 году. [3]. При этом суперкомпьютер ASCI Red занимал площадь примерно в 150 квадратных метров и потреблял около 850 киловатт электроэнергии. Удельная стоимость вычислений в подобных машинах была предельно высока, и позволить себе использовать их могли только крупные государственные структуры. Однако в 70-х годах 20-го века появились, и стали широко применяться относительно дешевые персональные компьютеры (ПК). Со временем ПК становились все доступнее и производительнее, и в определенный момент пришло осознание того, что некоторые сложные задачи, которые могут быть разделены на более мелкие части, можно решать используя множество персональных компьютеров за время, сравнимое со временем решения подобных задач на суперкомпьютере. Так появились первые кластерные вычислительные системы, представляющие собой локальные вычислительные сети, собранные на базе общедоступного оборудования. И хотя перечень задач, решаемых в подобных системах, был несколько уже, чем доступный суперкомпьютерам, стоимость построения и эксплуатации таких систем была намного ниже.

Первой системой, работающей подобным образом, стал кластер «Беофульф» (Beowulf), созданный в 1994 году Томасом Стерлингом (Sterling) и Доном Беккером (Becker). Вычислительная система представляла собой 16-узловой кластер из процессоров Intel DX4, соединенных сетью 10 Мбит/с Ethernet с дублированием каналов. Кластер использовался в центре NASA Goddard Space Flight Center для поддержки необходимыми вычислительными ресурсами проекта Earth and Space Sciences. В последующие годы появилось множество программных комплексов, позволяющих создать вычислительный кластер на базе принципов, схожих с используемыми при построении системы «Beowulf». В настоящее время под кластерной вычислительной системой, понимается вычислительный комплекс, состоящий из одного (реже более чем одного) серверного узла (который обычно называется головным), а также одного или нескольких подчиненных узлов, соединенных посредством выделенной компьютерной сети. В наши дни кластерные вычислительные системы обрели очень широкое распространение: среди суперкомпьютеров из списка Top500 (top500.org) по информации на ноябрь 2011 года более 82% суперкомпьютеров являются кластерными вычислительными системами. [4]

Начавшееся в 90-х годах повсеместное распространение компьютерных сетей позволило ускорить процесс доставки задач на географически удаленные ПК, что сделало возможным создание вычислительных систем, распространенных в пространстве. Это позволило еще более сократить удельную стоимость распределённых вычислений за счет объединения в вычислительные сети уже существующего оборудования. Этим воспользовались многие крупные организации для построения своих собственных распределенных вычислительных сетей, однако в конце 90-х стало очевидно, что каждой организации нецелесообразно проектировать собственную вычислительную систему для решения своих задач, и тогда начали создаваться первые облачные сервисы и GRID-системы.

Облачные сервисы являются моделью обеспечения повсеместного и удобного сетевого доступа по требованию к общему пулу конфигурируемых вычислительных ресурсов (например, сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам — как вместе, так и по отдельности), которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами и/или обращениями к провайдеру.

Потребители облачных вычислений могут значительно уменьшить расходы на инфраструктуру информационных технологий (в краткосрочном и среднесрочном планах)

и гибко реагировать на изменения вычислительных потребностей, используя необходимое количество удаленных ресурсов облачных сервисов.

Основную сложность при разработке подобных систем представляет создание диспетчера. Под диспетчером здесь понимается набор программно-аппаратных средств, отвечающих за распределение удаленных вычислительных ресурсов при решении потока поступающих задач. [6][7] Именно диспетчер занимается выбором ИВУ для решения поступающих задач, поэтому от него зависит эффективность работы системы. В современных распределенных системах диспетчер строится, как правило, на базе специально выделенных серверов, что накладывает ряд ограничений, а так же увеличивает удельную стоимость распределенных вычислений. Во-первых, такая централизованная организация диспетчера заметно усложняет процесс масштабирования: каждый из СВУ системы имеет ограничения по количеству одновременно обслуживаемых ИВУ, поэтому в случае расширения вычислительной системы в её состав необходимо будет добавлять не только новые ИВУ, но и СВУ. Во-вторых, такая организация диспетчера приводит к резкому снижению отказоустойчивости, так как множество СВУ является слабым местом системы: в случае отказа одного из них система становится частично или даже полностью неработоспособна. В-третьих, наличие специально выделенных СВУ, не участвующих в процессе вычисления приводит к повышению удельной стоимости вычислений, так как содержание инфраструктуры для этих серверов требует значительных материальных ресурсов.

Таким образом, можно сделать вывод, что организация диспетчера системы распределенных вычислений с помощью выделенных СВУ не позволяет создавать вычислительные системы для решения широкого класса задач на базе сети ПК частных владельцев.

Данная проблема может быть решена путем создания децентрализованного диспетчера, основанного на принципах мультиагентного взаимодействия. Основной идеей данного подхода является перемещение функций диспетчирования с выделенных серверов на множество исполнительных вычислительных узлов и диспетчирование путем взаимодействия независимых агентов, каждый из которых осуществляет управление своим ИВУ-«фрилансером». Для решения каждой из поступающих задач агенты должны объединяться в сообщества, ресурсов которого хватает для решения задачи за требуемое время. Процессы распределения и решения задач внутри сообщества распараллеливаются между всеми членами сообщества. При этом каждый из агентов должен осуществлять постоянный мониторинг параметров своего ИВУ, что позволит реагировать на любые изменения его параметров и оперативно учитывать эти изменения в процессе диспетчирования.

Преимуществами такого подхода к диспетчированию являются: повышение отказоустойчивости системы (отказ любого из узлов системы не приведет к ее неработоспособности), простота масштабирования (нет необходимости добавлять в систему новые СВУ в случае увеличения количества ИВУ системы), низкие требования к инфраструктуре (не нужно содержать парк СВУ для поддержания работоспособности системы) и как следствие заметное понижение удельной стоимости вычислений. Но его реализация требует разработки новых методов и алгоритмов мультиагентного диспетчирования.

2. Метод децентрализованной организации облачных вычислений

На сегодняшний день существует несколько различных подходов к децентрализации функций между независимыми узлами, однако одним из наиболее распространенных является мультиагентный подход.

Мультиагентный подход подразумевает под собой схему, основанную на одновременной работе множества независимых агентов, которые в процессе взаимодействия приходят к определённым договоренностям, определяющим в итоге эмергентное поведение системы. В концепции мультиагентных систем под агентом подразумевается некоторая проактивная сущность в рамках системы, которая обладает собственными интересами. Одновременно в мультиагентной системе существует множество независимых друг от друга агентов, они взаимодействуют друг с другом, и каждый из них стремится максимально удовлетворить свой интерес. При этом, коллективное взаимодействие независимых агентов в рамках системы позволяет достичь глобальную цель системы в целом как совокупность локальных целей каждого из агентов. Именно это делает мультиагентную систему децентрализованной: каждый из агентов не контролируется извне, однако их группа решает общую задачу.

Мультиагентный подход может быть использован при организации и децентрализованного облачного сервиса (далее для краткости ОС). Основными взаимодействующими элементами системы целесообразно сделать агентов, которые выполняют функции как клиента ОС, так и диспетчера. Агенты физически располагаются на каждом ИВУ, и представляют их интересы в процессе работы мультиагентного ОС. При этом целесообразно положить, что основным интересом, который будут преследовать ИВУ при участии в работе ОС, будет некоторая виртуальная или материальная выгода: за решение каждой задачи ее автор (пользователь) предлагает некоторое количество виртуальных очков. Эти очки могут в реальности быть чем угодно: деньгами, различными привилегиями, просто показателем проделанной определенным ИВУ работы. Таким образом, агент будет заинтересован в решении как можно большего количества задач за как можно меньшее время, одновременно с тем для него выгодно решать задачи с наибольшей ценой и наименьшим количеством работы, так как это приносит наибольшую прибыль его владельцу. Основным отличием такой схемы построения системы от существующих является то, что подход не предполагает централизованного управления агентами.

При этом для решения каждой полученной системой задачи должна быть сформирована виртуальная организация множества агентов - сообщество. Под сообществом в дальнейшем будем понимать некоторый сформированный коллектив агентов, собравшихся вместе для решения одной определенной задачи. Основной целью создания сообщества является решение своей задачи в указанный пользователем срок, при этом состав сообщества может изменяться в процессе решения задачи: вследствие отказов оборудования ИВУ, изменения их производительности, снижения скорости передачи данных и другие изменения вычислительной сети, приводящие к увеличению времени решения пользовательской задачи, могут вызвать изменение состава сообщества.

В процессе формирования и работы сообщества части задачи, для решения которой сформировано данное сообщество, должны распределяться между агентами, включенными в состав сообщества. При этом в связи с неоднородностью оборудования в зависимости от полученного распределения частей задачи время решения задачи может варьироваться в широких пределах. Поэтому целесообразно производить оптимизацию размещения задач в сообществе с целью сокращения времени решения задачи. При этом для того, чтобы равномерно распределить нагрузку, процесс оптимизации должен быть распараллелен между всеми агентами сообщества.

Подобная мультиагентная организация ОС позволяет эффективно использовать вычислительные ресурсы ИВУ-«фрилансеров»: каждый из агентов может самостоятельно осуществлять мониторинг параметров своего ИВУ и, в случае их существенного изменения, инициировать перераспределение задач в сообществе или даже изменение состава сообщества.

Основываясь на приведенных выше соображениях можно предложить следующие основные принципы функционирования децентрализованного ОС с мультиагентным диспетчером:

- Основными активными элементами системы являются агенты, физически реализуемые на ИВУ ОС.
- Диспетчирование работы ОС осуществляется путем взаимодействия агентов. При этом агенты постоянно осуществляют мониторинг параметров своих ИВУ.
- Пользователь системы формирует и отправляет задачу на решение в ОС. При этом ИВУ-«фрилансеры» системы должны иметь мотив для решения задачи пользователя. Поэтому пользователь должен предлагать некоторую оплату за решение своей задачи за требуемое время;
- Для решения каждой поступающей задачи агенты организуют сообщество. Агенты объединяются в сообщество до тех пор, пока время решения задачи сообществом не станет меньше времени, указанного пользователем;
- В созданном сообществе агенты совместно производят распределение частей задачи, при этом размещение должно осуществляться таким образом, чтобы минимизировать время времени решения пользовательской задачи;
- Как только части задачи размещены, агенты запускают процесс решения задачи на ИВУ. В случае, если параметры какого-либо ИВУ сообщества существенно изменяются, то его агент оповещает остальных членов сообщества об этом и сообщество может изменить свой состав в случае необходимости. При этом происходит процесс перераспределения частей пользовательской задачи в новом сообществе.

В качестве способа взаимодействия частей децентрализованного ОС предлагается использовать взаимодействие с помощью пассивных служебных серверов, реализующих минимум функций. За счет пассивности наличие таких узлов в системе не приведет к централизации функций, а если функциональная нагрузка этих узлов будет сведена к минимуму, то для их реализации не потребуются производительных компьютеров, и как следствие не потребуются дополнительные материальные затраты на этапах организации системы её эксплуатации. Тогда эти серверы фактически будут выполнять роль «досок объявлений», в дальнейшем так и будем их называть. Схема работы подобной системы представлена на рисунке 1.

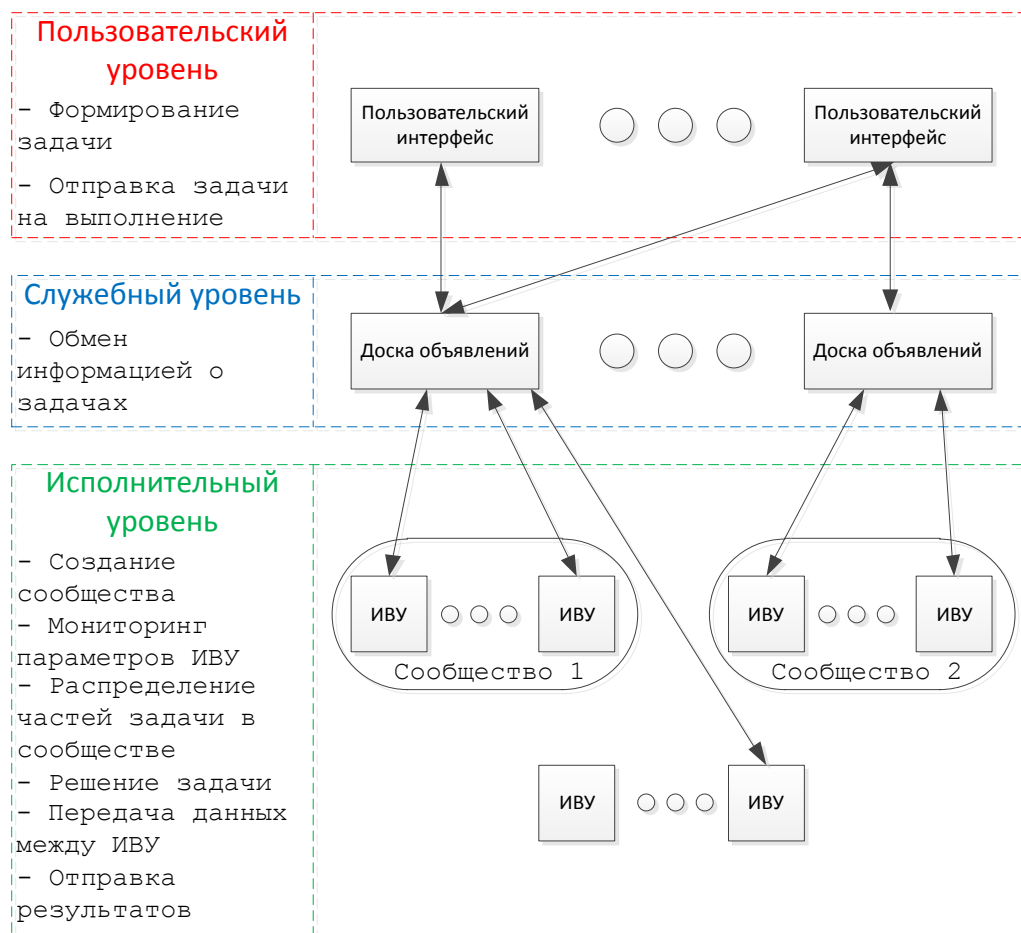


Рисунок 1 – Распределение функций в децентрализованном ОС

Из рисунка видно, что только функции приема и хранения задачи и выдачи служебной информации отданы доскам объявлений, остальные функции диспетчирования реализуются на уровне множества ИВУ. Количество досок объявлений, используемых в системе, ничем не ограничено, они могут быть выделены как пользователями ОС для своих собственных нужд, так и быть предоставленными третьей стороной, заинтересованной в получении выгоды от ОС. Это фактически делает систему полностью децентрализованной как на исполнительном, так и на служебном уровнях.

Принимая во внимание приведенные соображения можно предложить следующий процесс решения пользовательской задачи в децентрализованном ОС.

Пользователь системы у себя на ПК специальным образом формирует задачу для решения (разбивает задачу на подзадачи, регламентирует передачи данных от одной подзадачи к другой, указывает параметры подзадач, такие как трудоемкость и количества передаваемых данных), задает требования по времени ее решения в ОС, а также определяет мотивацию для решающих задачу ИВУ, а именно количество баллов (материальных средств, привилегий, и т.д.), которые владельцы ИВУ-«фрилансеров» получают за ее решение. После этого задача отправляется на ДО. Агенты децентрализованного ОС с некоторой периодичностью опрашивают ДО на предмет наличия нерешенных пользовательских задач. Если такая задача обнаруживается, то агенты начинают формировать сообщество для ее решения. При этом каждый агент оценивает для себя «выгодность» решению данной задачи с точки зрения соотношения ее трудоемкости (вычисли-

тельной сложности) и количества баллов, получаемых в результате ее решения. Если задача «выгодна» с точки зрения агента, то «его» ИВУ включается в сообщество по решению данной задачи. В определенный момент времени в сообществе набирается количество агентов, достаточное для решения задачи в указанный пользователем срок, после этого расширение сообщества прекращается. После этого агенты сообщества распределяют между собой подзадачи задачи и начинают процесс их решения. При этом в процессе решения задачи агенты производят постоянный мониторинг параметров своих ИВУ (производительность, скорость доступа к сети). В случае если какой-либо из параметров ощутимо изменяется, в сообществе подсчитывается время решения задачи с учетом новых параметров ИВУ, и, если время решения становится неприемлемым, происходит перераспределение задач или расширение сообщества. Схожим образом сообщество ведет себя и в случае отказа какого-либо узла: его задачи перераспределяются между агентами сообщества и в сообщество приглашаются агенты ему на замену. После того, как задача полностью решена, результат отправляется на ДО и обратно пользователю, а сообщество распускается.

В целом решение задачи в децентрализованного ОС происходит по алгоритму, приведенному на рисунке 2.

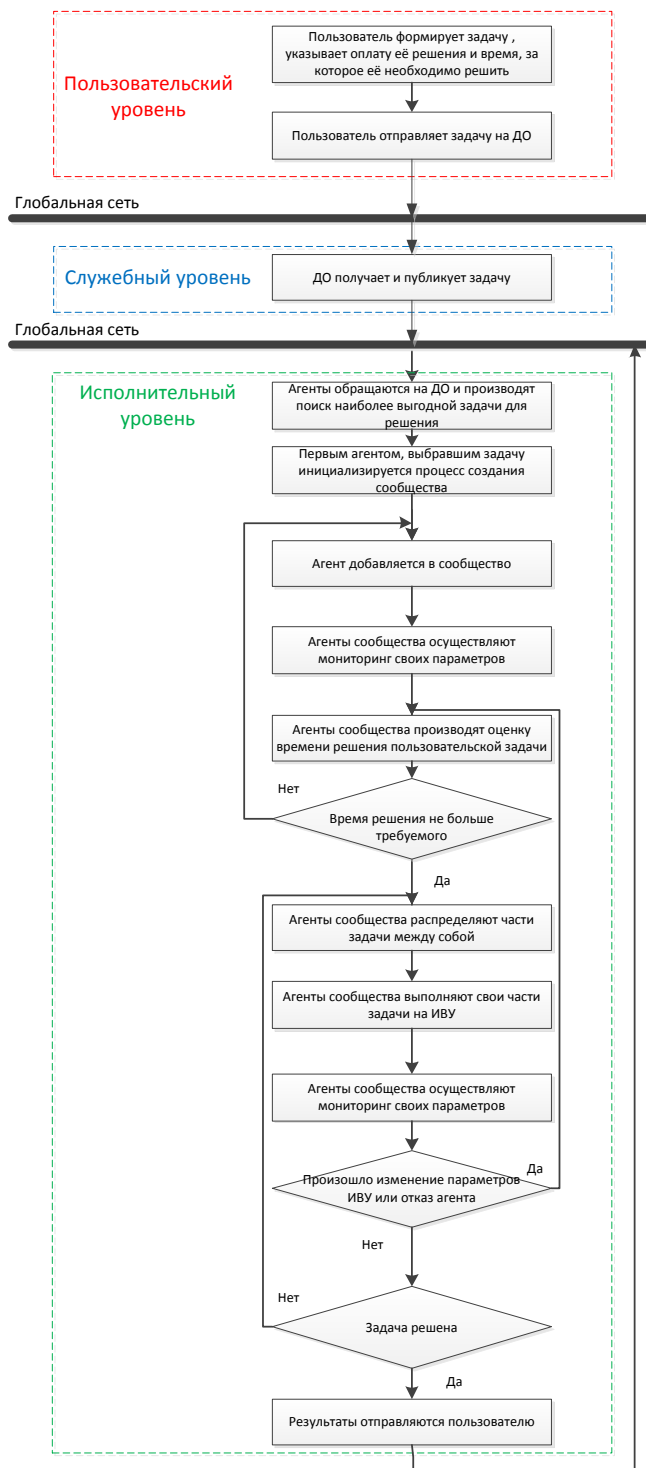


Рисунок 2 – Обобщенный алгоритм работы децентрализованного ОС

Описанный децентрализованный ОС, построенный на базе мультиагентного диспетчера, обладает рядом преимуществ по сравнению с классическими ОС:

Во-первых, децентрализованная организация системы позволяет постоянно отслеживать изменения параметров ИВУ вычислительной сети ОС и адаптивно подстраивать вычислительный процесс под эти изменения. Это достигается за счет того, что каждый из ИВУ имеет возможность отслеживать свои параметры в процессе решения задач, и, зная эти параметры, влиять на принятие решения о назначении на себя тех или иных подзадач с целью оптимизации времени решения задачи. Это позволяет строить ОС на базе вычислительных ресурсов ИВУ-«фрилансеров», параметры которых могут динамически изменяться во времени в зависимости от действий их владельцев.

Во-вторых, децентрализованная форма организации ОС подразумевает более высокую отказоустойчивость в связи с вынесением большинства функций управления с центрального диспетчера на узлы вычислительной сети, которые являются независимыми друг от друга. Это позволяет системе работать при отказе определенного количества узлов или ДО, в то время как в классических ОС отказ любого диспетчера приведет к частичной или полной неработоспособности системы;

В-третьих, существенно снижается удельная стоимость вычислений по сравнению с классическими ОС. Действительно, на служебный уровень системы возлагается ряд простейших пассивных функций доски объявлений, реализация которых не требует наличия дорогостоящего оборудования и больших затрат на его обслуживание. Поэтому можно считать что стоимость затрат на организацию и обслуживание служебного уровня ОС мало по сравнению со стоимостью работы ИВУ и соответственно стоимость вычислений в ОС в основном зависит только от стоимости работы множества ИВУ.

Оценим удельную стоимость вычислений в децентрализованного ОС, использующей ресурсы ИВУ-«фрилансеров»:

$$C_{\text{ОС}}^{\text{уд}} = \frac{C_{\text{дисп}} + C_{\text{выч}}}{T_{\text{раб}}}$$
, где $C_{\text{ОС}}^{\text{уд}}$ - удельная стоимость вычислений в ОС, $C_{\text{дисп}}$ – стоимость работы служебного уровня, $C_{\text{выч}}$ – стоимость работы исполнительного уровня, $T_{\text{раб}}$ – время работы ОС при решении задачи.

$C_{\text{дисп}} = \sum_{i=1}^M (C_{\text{ДО}}^i + C_{\text{наклДО}}^i)$, где M -количество СВУ в системе, $C_{\text{ДО}}^i$ – стоимость работы ДО i , $C_{\text{наклДО}}^i$ – стоимость накладных расходов, связанных с поддержкой инфраструктуры СВУ i . За счет возможности использования в качестве ДО географически распределённого низкопроизводительного оборудования $C_{\text{дисп}}$ для данной системы будет намного ниже, чем в классических ОС.

$C_{\text{выч}} = \sum_{j=1}^N C_{\text{иву-ф}}^j$, где N -количество ИВУ-«фрилансеров» в системе, $C_{\text{иву}}^j$ – стоимость работы ИВУ-«фрилансера» j . За счет использования ИВУ-«фрилансеров» дополнительные траты на накладные расходы по содержанию инфраструктуры и поддержке работоспособности в данном случае не нужны.

Таким образом, можно сделать вывод, что удельная стоимость вычислений в ОС $C_{\text{ОС}}^{\text{уд}}$ при использовании мультиагентного диспетчера будет значительно ниже, чем при использовании классических подходов, использующих управляющие СВУ.

3. Заключение

Предлагаемый алгоритм организации облачных вычислений позволяет реализовать все потенциальные преимущества децентрализованного облачного сервиса, такие как:

- Масштабируемость: вне зависимости от количества фрилансеров в системе объем оборудования, требуемого для организации работы системы практически не возрастает;
- Адаптивность: каждый из агентов системы обладает актуальной информацией о параметрах своего компьютера-фрилансера, что позволяет оперативно отслеживать их изменения и осуществлять изменения в составе сообщества в случае увеличения времени решения задачи выше указанного пользователем;
- Сокращение себестоимости облачных вычислений: За счет использования пассивных «досок объявлений» и фрилансеров организатор облачного сервиса может ощутимо сократить материальные затраты на организацию и поддержание работы сервиса.

СПИСОК ЛИТЕРАТУРЫ

1. *M. O. Tokhi, Mohammad Alamgir Hossain*. Parallel computing for real-time signal processing and control // 2003 ISBN 978-1-85233-599-1 pages 201-202
2. *N. Hirose and M. Fukuda* (1997). Numerical Wind Tunnel (NWT) and CFD Research at National Aerospace Laboratory. // Proceedings of HPC-Asia '97. IEEE Computer Society. doi:10.1109/HPC.1997.592130.
3. http://www.ai.mit.edu/projects/aries/course/notes/ascii_red.pdf
4. <http://i.top500.org/stats>
5. *I. Foster, C. Kesselman*. "The Grid: Blueprint for a New Computing Infrastructure", Morgan-Kaufman, 1999. <http://www.globus.org>
6. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications // 15(3), 2001. <http://www.globus.org/alliance/publications/papers/anatomy.pdf>
7. *Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Любимский Э.З., Хухлаев Е.В., Шорин О.Н.* Грид-диспетчер: реализация службы диспетчеризации заданий в грид // Труды международной конференции "Распределенные вычисления и Грид-технологии в науке и образовании" (Дубна, 29 июня-2 июля 2004 г.). - Дубна: 11-2004-205, ОИЯИ, 2004, сс. 133-139.