

03, март 2016

УДК 004.45

Моделирование подходов к распределению ресурсов и оценка времени распределения задач в облачных грид системах

Кошкарева Ю.И., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

Третьякова А.А., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

Научный руководитель: Черненко М.В., доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

chernen@bmstu.ru

Во время работы с распределенными вычислительными системами и предоставлением ресурсов в облачных пространствах возникает вопрос оптимального распределения поступающих задач между имеющимися вычислительными ресурсами. В связи с увеличением объемов задач так же становится актуальным вопрос расширения таких систем. В настоящее время решение задач на удаленных ресурсах становится выгоднее, чем обслуживание вычислительных машин в непосредственной близости с клиентом. Для исследования проблем, возникающих при создании и работе с облачными грид системами, смоделируем типовые варианты систем.

Для однозначности понимания определим основные термины, используемые в статье:

Облачное хранилище данных (англ. *cloud storage*) – онлайн-хранилище данных, предназначенное для хранения информации на удаленных виртуальных машинах.

Грид система (от англ. *grid* – сетка) – система, предоставляющая множество различных распределенных вычислительных ресурсов, которые необходимы для параллельного выполнения задач, и управляющая распределением задач по имеющимся ресурсам. В [4] выделяют несколько типов грид систем: вычислительные грид системы (*Computational Grid*), грид системы данных (*Data Grid*), семантические грид системы (*Semantic Grid*).

Облачные вычисления - набор сетевых сервисов, предоставляемых клиенту для выполнения его задач. Под клиентом будем понимать независимую рабочую станцию.

Выделим основные свойства, характерные для облачных вычислений:

- клиент имеет доступ к ресурсам по мере необходимости
- система самостоятельно выделяет ресурсы под задачи
- объемы предоставляемых ресурсов могут быть изменены в автоматическом режиме в соответствии с запросами клиентов

На вход грид системы поступают прикладные задачи обработки, хранения данных, которые могут иметь существенно различающиеся ресурсные требования. Возникает проблема с распределением ресурсов между задачами. Схема представлена на рис. 1.

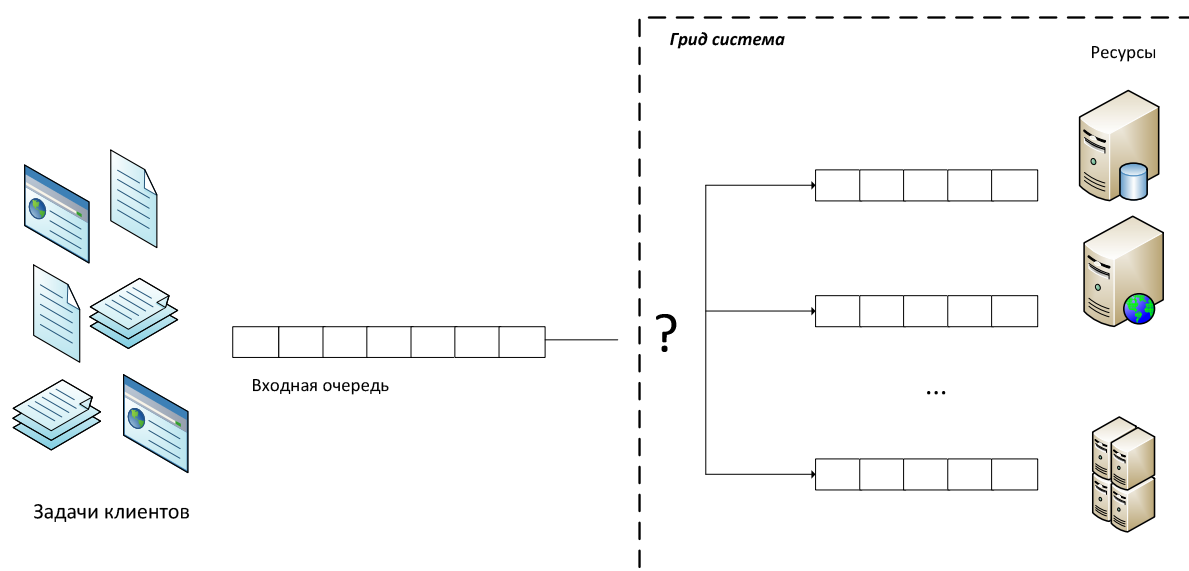


Рис. 1. Проблема распределения ресурсов между задачами

Планирование ресурсов является важным аспектом управления ресурсами облачных вычислительных систем. Моделирование задачи распределения ресурсов для различных алгоритмов является эффективным средством выбора алгоритма распределения.

Для решения проблемы распределения ресурсов используются планировщики задач. Под планировщиками понимают дополнительные узлы системы с программными средствами распределения ресурсов.

Основной задачей планировщика облачной вычислительной системы является увеличение пропускной способности системы, то есть увеличение количества задач, завершенных за единицу времени, и сокращение времени выполнения каждой задачи.

В [2] автор выделяет следующие подходы к планированию:

- централизованное планирование;
- иерархическое планирование;
- распределенное планирование.

Централизованное планирование ресурсов

В централизованном подходе к распределению ресурсов, распределение осуществляется одним планировщиком. Он обладает информацией обо всех виртуальных машинах данной облачной грид системы.

Задачи, поступившие от клиентов, поступают в общую очередь планировщика. Он реализует подбор необходимых ресурсов для решения каждой задачи. Схема работы грид системы, основанная на данном подходе, представлена на рис. 2.

Достоинствами централизованного подхода можно считать: относительную дешевизну подсистемы планирования, так как используется всего один дополнительный узел (планировщик), высокую эффективность планирования, простоту в реализации.

Недостатками данного подхода являются: плохая масштабируемость, наличие планировщика в качестве «узкого места» системы.

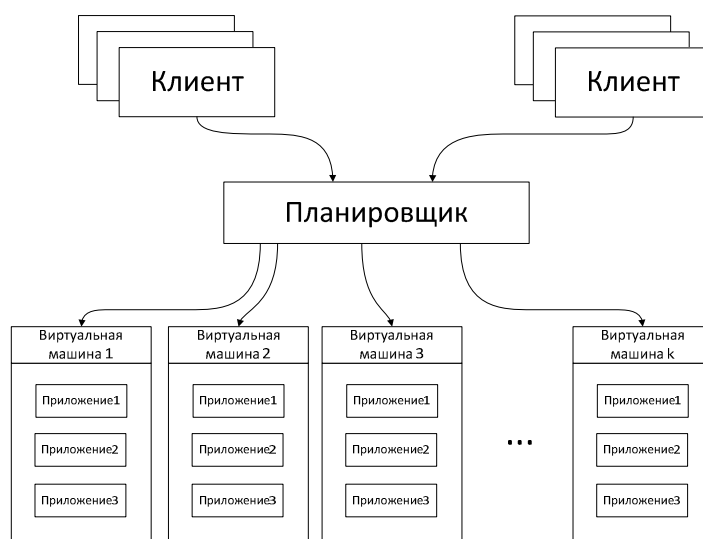


Рис. 2. Схема работы централизованного планировщика ресурсов облачной вычислительной системы

Время T , которое необходимо планировщику для выполнения функции распределения поступающих на него задач между ресурсами облачной вычислительной системы, можно рассчитать по формуле:

$$T = t_{\text{обп}}nk, \quad (1)$$

где $t_{\text{обр}}$ – время обработки одной задачи

n – количество задач, которые поступают от клиентов

k – количество виртуальных машин, предоставляющие ресурсы в облачной вычислительной системе.

Предполагается худший вариант, согласно которому при распределении задач планировщику необходимо просмотреть все доступные ресурсы в количестве k штук.

Иерархическое планирование ресурсов

В иерархическом подходе к распределению ресурсов вместо одного планировщика используется система из нескольких уровней планирования (рис. 3). Глобальный планировщик отвечает за распределение задач между локальными планировщиками. В свою очередь локальные планировщики распределяют задания лишь в доступной им группе виртуальных машин.

Таким образом, мы получаем двухуровневую систему планирования. При иерархическом способе распределения задач появляется возможность использовать разные алгоритмы планирования на глобальном и локальном уровнях. Однако такой способ ведет к увеличению стоимости системы, так как необходима покупка и установка множества дополнительных узлов.

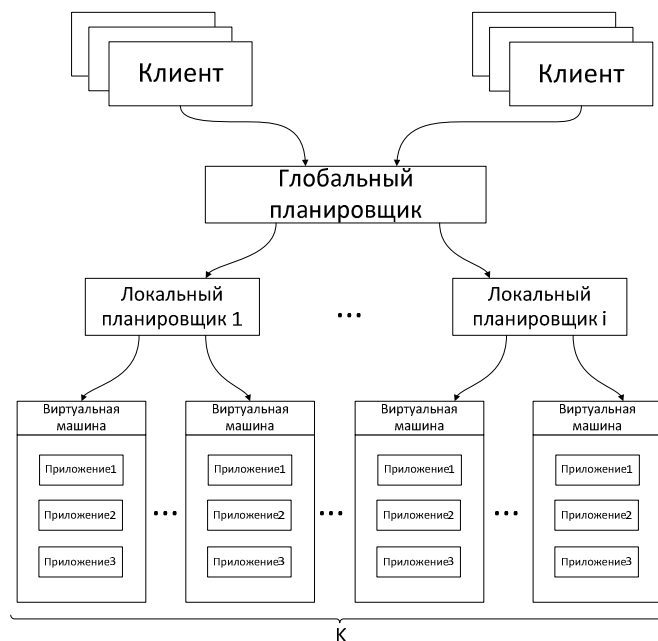


Рис. 3. Схема работы иерархического планировщика ресурсов облачной вычислительной системы

Интервалы времени, необходимые глобальному планировщику и локальным планировщикам для распределения n задач по k виртуальным станциям можно представить в виде формул:

$$T_{\text{гл}} = t_{\text{обр}} n i \quad (2.1)$$

$$T_i = t_{\text{обр}} n_i k_i, \quad (2.2)$$

где $T_{\text{гл}}$ - время распределения задач глобальным планировщиком

i - количество локальных планировщиков

n_i - количество задач, которые получил для распределения i -ый локальный планировщик от глобального планировщика

k_i - количество ресурсов (виртуальных станций), за которые отвечает i -ый локальный планировщик

При этом очевидны равенства: $\sum_i n_i = n$, $\sum_i k_i = k$.

Распределенное планирование ресурсов

В случае, когда используется распределенное планирование ресурсов возможно два варианта реализации:

- распределение ресурсов облачной вычислительной системы с прямым взаимодействием локальных планировщиков
- распределение ресурсов облачной вычислительной системы с взаимодействием локальных планировщиков через общий пул

Рассмотрим оба способа подробнее. На рис. 5 представлен вариант системы с прямым взаимодействием локальных планировщиков. В данной реализации каждый локальный планировщик получает новые задачи напрямую от клиентов, по возможности распределяя задания на доступных ему ресурсах. Каждый локальный планировщик может видеть всю карту сети, поэтому в случае отсутствия возможности выполнить задание самостоятельно, он находит доступные ресурсы у других локальных планировщиков и передает свои задания туда.

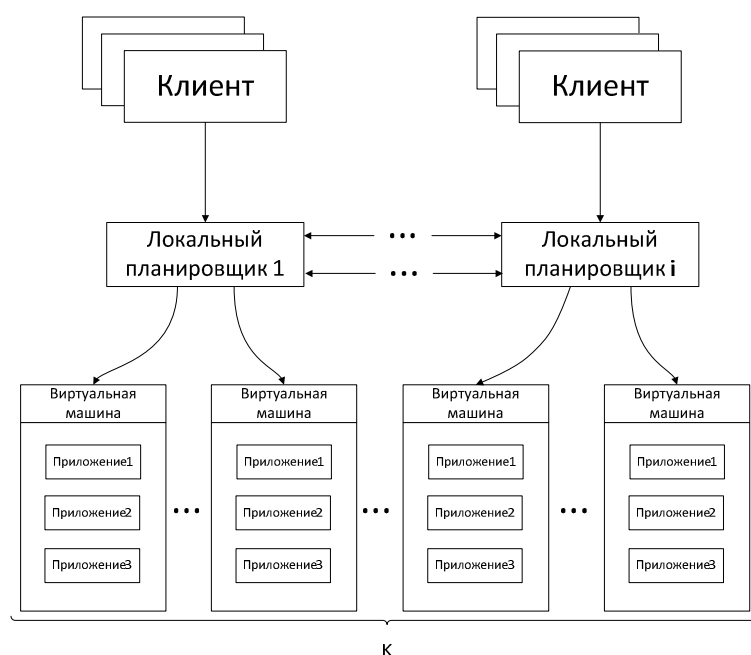


Рис.4. Схема работы децентрализованного планировщика ресурсов облачной вычислительной системы с прямым взаимодействием

Более сложные вычисления происходят в случае схемы децентрализованного планировщика ресурсов с прямым взаимодействием локальных планировщиков:

$$T_i = t_{\text{обр}} n_i k_i + t_{\text{пров}} * (i - 1) * n_j + t_{\text{обр}} k_i n_l, \quad (3)$$

где T_i - время планирования распределения ресурсов для i -го локального планировщика.

В T_i учитывается время на распределение задач, поступивших напрямую от клиентов i -му планировщику, между ресурсами, время на распределение задач, поступивших от других локальных планировщиков, время на проверку других планировщиков в случае, если какую-то задачу невозможно решить с помощью ресурсов, подчиненных i -му планировщику ($t_{\text{пров}}$).

n_j - количество задач, которые нельзя выполнить на имеющихся ресурсах. Эти задачи передаются для распределения на другой планировщик.

n_l - количество задач, которые поступают с других планировщиков

При этом очевидны равенства: $\sum_i n_i = n$, $\sum_i k_i = k$.

Взаимодействие планировщиков через общий пул задач

Вторым способом реализации распределенного планирования ресурсов является система с общим пулом задач, представленная на рис.6. В данном случае все задачи от клиентов поступают сразу в общий пул задач. Каждый локальный планировщик ищет в

пуле задания, которые он способен реализовать на своих ресурсах и забирает часть задач на выполнение. Несмотря на то, что в данном случае локальные планировщики ничего не знают о возможности друг друга, данная схема позволяет существенно снизить скорость обработки пользовательских задач. Недостатком способа реализации распределенного планирования ресурсов с общим для локальных планировщиков пулом задач является стоимость подобных систем.

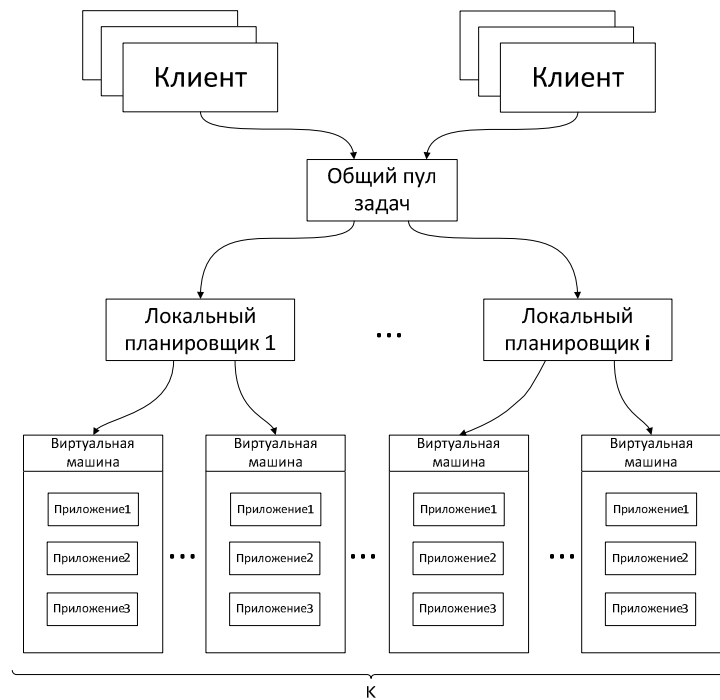


Рис. 5. Схема работы децентрализованного планировщика ресурсов облачной вычислительной системы с взаимодействием через общий пул

В случае децентрализованного планирования ресурсов облачной вычислительной системы с взаимодействием через общий пул для расчета T_i необходимо учесть время выборки задачи из пула i -ым локальным планировщиком.

$$T_i = t_{\text{обр}} n_i k_i + t_{\text{выб}} n_j, \quad (4)$$

где $t_{\text{выб}}$ - время проверки локальным планировщиком одной задачи на возможность ее выполнения с помощью имеющихся ресурсов

n_j - количество задач, находящихся в пуле в момент выборки задачи. В начальный момент времени $n_j = n$

В качестве примера рассмотрим конфигурацию датацентра со следующими параметрами: количество задач, которые поступили, $n=1000$. $t_{обp}=1$ мс. Количество виртуальных машин (ресурсов) $k=4$.

Централизованное планирование

Согласно формуле (1): $T = 1 \text{ мс} * 1000 * 4 = 4000 \text{ мс} = 4 \text{ с}$.

Иерархическое планирование

Предполагается, что сначала отрабатывает глобальный планировщик, а после параллельно друг другу начинают работать локальные планировщики. Пусть имеется два локальных планировщика ($i=2$). Каждый локальный планировщик может отдать задачу на одну из двух виртуальных машин.

Используя формулу (2.1), получаем: $T_{гл}=1 \text{ мс} * 1000 * 2 = 2 \text{ с}$.

Рассмотрим первый случай, в котором задачи между локальными планировщиками были разделены поровну: $n_{1,2} = n_1 = n_2 = 500$.

Согласно формуле (2.2): $T_1 = T_2 = T_{1,2} = 1 \text{ мс} * 500 * 2 = 1 \text{ с}$.

Общее время распределения задач: $T = T_{гл} + \max(T_1; T_2) = 2 \text{ с} + 1 \text{ с} = 3 \text{ с}$.

Рассмотрим худший из возможных случаев, когда задачи между локальными планировщиками были разделены так, что все n задач достались одному локальному планировщику для дальнейшего распределения: $n_1 = 1000, n_2 = 0$.

Согласно формуле (2.2): $T_1 = 1 \text{ мс} * 1000 * 2 = 2 \text{ с}, T_2 = 1 \text{ мс} * 0 * 2 = 0 \text{ с}$.

Общее время распределения задач: $T = T_{гл} + \max(T_1; T_2) = 2 \text{ с} + 2 \text{ с} = 4 \text{ с}$.

Децентрализованное планирование

Как и в предыдущем варианте в распоряжении каждого локального планировщика по две виртуальные машины. На каждый планировщик от клиентов поступает $n_{1,2} = n_1 = n_2 = 500$ задач. При таком типе распределения ресурсов планировщики могут обмениваться задачами. Пусть $t_{пров} = 3 \text{ мс}$.

Рассмотрим первый случай, когда задачи между локальными планировщиками были разделены так, что пересылать задачи другому планировщику не надо, поскольку их возможно выполнить с помощью имеющихся ресурсов. Тогда $n_j = 0$ и $n_l = 0$.

Используя формулу (3), получаем: $T_i = T_1 = T_2 = 1 \text{ мс} * 500 * 2 + 3 \text{ мс} * (2 - 1) * 0 + 1 \text{ мс} * 2 * 0 = 1 \text{ с}$.

Общее время распределения задач: $T = \max(T_1; T_2) = 1 \text{ с}$.

Рассмотрим второй случай, когда задачи между локальными планировщиками были разделены так, что все свои задачи каждый планировщик должен отдать другому планировщику за неимением ресурсов для выполнения поставленных задач. Тогда $n_j = 500$ и $n_l = 500$.

Используя формулу (3), получаем: $T_1 = T_2 = 1 \text{ мс} * 500 * 2 + 3 \text{ мс} * (2 - 1) * 500 + 1 \text{ мс} * 2 * 500 = 3,5 \text{ с}$.

Общее время распределения задач: $T = \max(T_1; T_2) = 3,5 \text{ с}$.

Децентрализованное планирование с взаимодействием через общий пул

Примем $t_{\text{выб}} = 1 \text{ мс}$. Пусть все задачи были забраны первым локальным планировщиком. Тогда согласно формуле (4): $T_1 = 1 \text{ мс} * 1000 * 2 + 1 \text{ мс} * 1000 = 3 \text{ мс}$, $T_2 = 0$.

Общее время распределения задач: $T = \max(T_1; T_2) = 3 \text{ с}$.

В другом возможном случае каждый планировщик выбрал по 500 задач из пула

Согласно формуле (4): $T_1 = 1 \text{ мс} * 500 * 2 + 1 \text{ мс} * 500 = 1,5 \text{ с}$, $T_2 = 1 \text{ мс} * 500 * 2 + 1 \text{ мс} * 500 = 1,5 \text{ с}$.

Общее время распределения задач: $T = \max(T_1; T_2) = 1,5 \text{ с}$.

Сводная таблица результатов общего времени распределения задач для различных способов планирования, рассмотренных в примере:

Централизованный способ планирования	Иерархический способ планирования		Децентрализованный способ планирования		Децентрализованный способ планирования с взаимодействием через общий пул	
T, с	T _{min} , с	T _{max} , с	T _{min} , с	T _{max} , с	T _{min} , с	T _{max} , с
4	3	4	1	3,5	1,5	3

В итоге нами был выполнен обзор технологий моделирования распределенных вычислительных систем. Были рассмотрены основные подходы к планированию облачных ресурсов. Также была проведена оценка времени распределения задач в облачных грид системах. Сравнение результатов анализа показало, что для различных типов конфигурации сети необходим индивидуальный подход. Централизованный способ планирования ресурсов подходит для небольших сетей с малой загруженностью и требует небольшого аппаратного обеспечения. Иерархический и распределенный способы

планирования подойдут для сетей большого масштаба и с большей загруженностью. Выбирая из этих двух подходов, следует отталкиваться от показателей нагрузки для каждой отдельной конфигурации сети. Иерархический и распределенный способы требуют для реализации покупки и установки определённого количества дополнительных узлов, что сказывается на стоимости системы в целом.

Список литературы

- [1]. Walter Lee, Rajeev Barua, Matthew Frank, Devabhaktuni Srikrishna, Jonathan Babb, Vivek Sarkar, Saman Amarasinghe Space-Time Scheduling of Instruction-Level Parallelism on a Raw Machine. Режим доступа:
<http://groups.csail.mit.edu/cag/raw/documents/Lee-ASPLOS-1998.pdf> (дата обращения 15.09.15).
- [2]. Михайлов П.А., Радченко Г.И. Методы моделирования и оценки производительности облачных систем. Режим доступа:
<http://cyberleninka.ru/article/n/metody-modelirovaniya-i-otsenki-proizvoditelnosti-oblachnyh-sistem> (дата обращения 19.09.15).
- [3]. Голубев И. А. Планирование задач в распределённых вычислительных системах на основе метаданных. Режим доступа:
http://www.eltech.ru/assets/files/nauka/dissertacii/2014/Dissertaciya_golubev_26apr.pdf (дата обращения 15.09.15).
- [4]. Величкевич С., Демченко Ю. Использование Грид-технологий для построения распределенных САПР Режим доступа:
<http://www.uazone.org/demch/papers/relam2005-cad-grid.pdf> (дата обращения 20.09.15).
- [5]. Черненький В.М., Черненький М.В. Основы процессного описания функционирования систем. Режим доступа:
<http://simulation.su/files/immod2013/material/immod-2013-1-290-296.pdf> (дата обращения 17.09.15).
- [6]. Печинкин А.В., Тескин О.И., Цветкова Г.М. Теория вероятностей 3-е издание, испр. М.: Издательство МГТУ им. Н.Э. Баумана, 2004. 456 с.