Модель распределения ресурсов в процессе функционирования облачной вычислительной среды

В статье предложен новый подход к распределению нагрузки в облачной вычислительной среде, который основывается на анализе равномерности использования ресурсов и прогнозе нагрузки с использованием интеллектуальных алгоритмов.

Ключевые слова: облачные вычисления, инфраструктура как сервис (IaaS), распределение ресурсов, искусственная нейронная сеть (ИНС).

RESOURCE DISTRIBUTION MODEL IN CLOUD ENVIRONMENTS

A new approach to distribution of load in computer clouds is proposed, based on the analysis of the equability of resource usage and resource usage forecast using intelligent algorithms.

Keywords: cloud computing, infrastructure-as-a-service (IaaS), resource distribution, artificial neural network

1. Введение

В статье [1] была доказана актуальность проблемы распределения ресурсов в облачных вычислительных средах и предложена методика первоначального выделения ресурсов на основе метода анализа иерархий для повышения эффек-

тивности использования ресурсов облачной вычислительной среды (рис. 1).

Суть указанной методики заключается в том, чтобы запустить очередной экземпляр (виртуальную машину, предоставляемую как сервис внутри облачной вычислительной среды) на хосте с наимень-

шей вероятностью нерационального использования ресурсов, что обеспечивает более эффективное использование ресурсов облачной вычислительной среды в целом.

В данном случае под эффективностью облачной вычислительной среды $E_{обл}$ понимается доля экземпляров из общего их числа,

Рис. 1. Первоначальное выделение ресурсов в облачной вычислительной среде с использованием метода анализа иерархий

Рамиль И. Хантимиров,
ООО «СТОРМ СИСТЕМС»
Эл. почта: ramil@stormwall.pro

Ramil I. Khantimirov,
LLC STORM SYSTEMS
E-mail: ramil@stormwall.pro

Андрей Александрович Микрюков,
к.т.н., доцент
Российский экономический
университет им. Г.В. Плеханова
Тел.: 8 (495) 442-61-11
Эл. почта: AMikrukov@mesi.ru

Andrey A. Mikrukov,
PhD in Technical Sciences,
Associate Professor
Plekhanov Russian University
of Economics
Тел.: 8 (495) 442-61-11,
E-mail: AMikrukov@mesi.ru

функционирующих без потери производительности из-за перегрузки каких-либо ресурсов.

Показатель эффективности $E_{\text{обл}}$ является комплексным показателем, зависящим от совокупности частных показателей:

$E_{\text{ЦП}}$ – процент экземпляров с удовлетворенной потребностью в ресурсах центрального процессора;

$E_{\text{ОЗУ}}$ – процент экземпляров с удовлетворенной потребностью в ресурсах памяти;

$E_{\text{Д}}$ – процент экземпляров с удовлетворенной потребностью в дисковых ресурсах;

$E_{\text{С}}$ – процент экземпляров с удовлетворенной потребностью в сетевых ресурсах;

T – время функционирования облачной вычислительной среды.

$$E_{\text{обл}} = F(E_{\text{ЦП}}, E_{\text{ОЗУ}}, E_{\text{Д}}, E_{\text{С}}, T) \rightarrow \max \quad (1)$$

При этом необходимо учесть следующее ограничение:

$$R_{\text{исп}} < R_{\text{имеющ}} \quad (2)$$

где:

$R_{\text{исп}}$ – общий объем использованных ресурсов облачной вычислительной среды;

$R_{\text{имеющ}}$ – общий объем имеющихся ресурсов облачной вычислительной среды.

Максимальное значение показателя $E_{\text{обл}}$ соответствует максимальному значению производительности существующих экземпляров при максимальной емкости облачной вычислительной среды, т.е. способности без потери производительности или с минимальной ее потерей обеспечивать функционирование дополнительных экземпляров с учетом существующих ресурсных ограничений.

Тем не менее, решение задачи первоначального выделения ресурсов не обеспечивает последующего рационального распределения ресурсов в ходе дальнейшего функционирования облачной вычислительной среды. С учетом того, что внутри экземпляров (виртуальных машин – будем считать, что термин «экземпляр» и «виртуальная машина» в контексте облачной вычислительной среды являются

синонимами) реализованы приложения и программные комплексы, такие, как СУБД, веб-сервисы, ERP-системы и др. [4]. Параметры этих экземпляров, связанные с нагрузкой, существенно изменяются во времени. Из этого следует, что показатель эффективности функционирования облачной вычислительной среды также меняется во времени и для обеспечения требуемого значения показателя эффективности $E_{\text{обл}}$ необходим постоянный мониторинг использования ресурсов и своевременное их перераспределение между виртуальными машинами во избежание потери производительности в различных участках облачной вычислительной среды, иными словами, балансировка нагрузки виртуальных машин на имеющихся ресурсах облачной вычислительной среды.

В качестве механизма балансировки нагрузки в облачной вычислительной среде предложено использовать подход на основе «живой миграции», под которой понимается технология миграции виртуальных машин между хостами или ресурсами хранения с нулевым временем простоя приложений.

Очередность процесса живой миграции в случае смены хоста (физического сервера, на котором выполняется виртуальная машина) выглядит следующим образом:

- 1) Остановка виртуальной машины;
- 2) Передача параметров виртуальной машины с сервера исходного расположения на сервер целевого расположения;
- 3) Передача образа оперативной памяти с сервера исходного расположения виртуальной машины на сервер целевого расположения;
- 4) Создание виртуального домена и размещение образа оперативной памяти в оперативной памяти сервера целевого расположения;
- 5) Запуск виртуальной машины на сервере целевого расположения.

При необходимости смены хранилища, образ виртуальной машины переносится с одного ресурса хранения на другой.

Для того, чтобы эффективно балансировать нагрузку в облачной вычислительной среде, необходимо прогнозировать ее изменение во времени, поскольку основываясь только на текущей нагрузке довольно сложно предотвратить нехватку ресурсов – можно только устранить ее по факту. Более того, в некоторых ситуациях перенос виртуальных машин не требуется, а лишь повышает накладные расходы на миграцию [2].

Исходя из вышесказанного, процесс перераспределения ресурсов в облачной вычислительной среде можно разделить на два этапа:

- 1) Прогноз нагрузки;
- 2) Выявление хостов с повышенной нагрузкой и динамическое перераспределение ресурсов.

В настоящей статье рассмотрено решение задачи на втором этапе процесса распределения ресурсов.

2. Постановка задачи распределения ресурсов

Для решения задачи распределения ресурсов предложено использовать концепцию неравномерности N_R использования ресурсов p -го сервера (N_R^p), представленную в [1]. Пусть n – число рассматриваемых ресурсов, а r_i – прогнозируемая загрузка i -го ресурса сервера p . Определим прогнозируемую неравномерность загрузки ресурсов сервера p как:

$$N_R^p = \sqrt{\sum_{i=1}^n \left(\frac{r_i - \bar{r}}{\bar{r}} \right)^2}, \quad (4)$$

где $\bar{r}$ – это средняя прогнозная загрузка всех ресурсов сервера p .

На практике не для всех типов ресурсов снижение производительности сказывается на эффективности функционирования системы в целом, поэтому необходимо учитывать только значимые ресурсы в расчете (такие, как загрузка центрального процессора (ЦП), оперативной памяти, дисковой подсистемы, сети и др.). Путём минимизации показателя неравномерности удастся совместить различные типы нагрузки и добиться улучшения общего использования

Рис. 2. Динамическое перераспределение ресурсов в облачной вычислительной среде

серверных ресурсов облачной вычислительной среды (рис. 2).

Алгоритм выполняется периодически для оценки состояния выделенных ресурсов, которая основана на прогнозе потребностей экземпляров. Сервер называется «горячей точкой», если использование любого из его ресурсов выше т.н. «допустимой отметки», предварительно заданной администратором облачной вычислительной среды для каждого типа ресурсов. Это говорит о том, что хост перегружен, поэтому часть экземпляров должна быть перемещена с него на другие хосты. «Температура» t^* горячей точки определяется как квадратичная сумма использования всех ее ресурсов выше «допустимой отметки» [5]:

$$t^* = \sum_{r \in R} (r - r_i)^2, \quad (5)$$

где R – набор перегруженных ресурсов сервера p ;

r_i – «допустимая отметка» ресурса r (при расчете учитываются только перегруженные ресурсы).

Температура горячей точки отражает степень перегрузки сервера. Если сервер не является горячей

точкой, его температура равняется нулю.

Различные типы ресурсов имеют различные допустимые отметки. К примеру, для загрузки ЦП и занятой оперативной памяти они могут быть определены как 90% и 80% соответственно. Таким образом, сервер становится горячей точкой при достижении этой нагрузки.

3. Алгоритм балансировки нагрузки в облачной вычислительной среде

Алгоритм балансировки нагрузки в облачной вычислительной среде включает совокупность шагов:

- 1) Сортировка списка серверов-«горячих точек» p по убыванию температуры t^* (т.е. наиболее горячая становится первой в списке). Цель – исключить все горячие точки, если это возможно, или, по крайней мере, поддерживать их температуру настолько низкой, насколько это возможно.

- 2) Определение для каждого сервера p списка экземпляров e , для которых должна быть выполнена миграция.

Рис. 3. Алгоритм снижения неравномерности использования ресурсов в облачной вычислительной среде

3) Сортировка списка экземпляров e на основе результирующей температуры хоста $t_{результ}^*$ которая определяется после миграции экземпляра (виртуальной машины). Цель – выполнить миграцию того экземпляра, который понизит тем-

пературу сервера p_n до минимального значения: $t_{результ}^*(p_n) \rightarrow \min$.

4) Выбор виртуальной машины e' , миграция которой позволит максимально снизить неравномерность сервера.

5) Определение для найденной виртуальной машины e' возможности нахождения целевого сервера $p_{целевой}$ для ее размещения. Сервер считается подходящим для миграции, если после переноса на него виртуальной машины e' неравномерность целевого сервера окажется меньше неравномерности текущего.

6) Выбор сервера $p'_{целевой}$ из этого списка, неравномерность которого будет максимально снижена после переноса данной виртуальной машины. Стоит обратить внимание, что неравномерность целевого сервера может быть и увеличена. В этом случае необходимо выбрать сервер, для которого такое увеличение будет минимальным.

7) Если целевой сервер $p'_{целевой}$ найден, инициируется миграция виртуальной машины на этот сервер и обновляется прогноз нагрузки для всех участвующих серверов. В противном случае, из списка выбирается следующая виртуальная машина и происходит поиск целевого сервера для нее.

Блок-схема алгоритма представлена на рис. 3.

До тех пор, пока можно найти целевой сервер для любой виртуальной машины сервера-«горячей

точки», выполнение алгоритма для данного сервера продолжается. Как только это становится невозможным, осуществляется переход к следующей «горячей точке».

Прогнозирование нагрузки с использованием искусственной нейронной сети перед запуском алгоритма позволяет не учитывать несущественных, непродолжительных пиков нагрузки, что ведет к значительному снижению числа ложных срабатываний алгоритма устранения горячих точек и, таким образом, снижению накладных расходов на перемещение виртуальных машин.

4. Вывод

В результате применения механизма прогнозирования изменения нагруженности хостов в облачной вычислительной среде и своевременного перераспределения ресурсов в динамике функционирования приложений внутри экземпляров удалось достичь значительного повышения эффективности использования ресурсов в облачной вычислительной среде, повысить способность облачной вычислительной среды принимать новые экземпляры с минимальным снижением производительности уже функционирующих приложений. Так, использование предложенного алгоритма позволило уменьшить снижение производительности экземпляров в тестовой облачной среде в среднем на 12%.

Литература

1. Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment – Zhen Xiao, Weijia Song and Qu Chen – IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), 2013
2. Хантимиров Р.И., Микрюков А.А. Прогнозирование нагрузки в облачной вычислительной среде с использованием нейросетей Элмана, обучаемых системой искусственного иммунитета // Нейрокомпьютеры, Москва 2015.
3. Васильев В.И. Интеллектуальные системы защиты информации – 2-е изд. – М.: Машиностроение, 2013. – 171 с.
4. Sanjib Mishra. Short Term Load Forecasting Using Computational Intelligence Methods, Department of Electronics and Communication Engineering National Institute Of Technology, Rourkela, 2008, с. 53–62.