

Моделирование облачной инфраструктуры ЛИТ ОИЯИ и проверка модели

В.С. Айриян^{1,2,4,a}, В.В. Кореньков^{2,3,b}, А.В. Нечаевский^{2,c}

¹ Государственный Университет «Дубна», Дубна, Российская Федерация

² Объединенный институт ядерных исследований, Дубна, Российская Федерация

³ РЭУ им. Г. В. Плеханова, Москва, РЭУ им. Г. В. Плеханова

⁴ Мюнхенский Технический Университет, Мюнхен, Германия

E-mail: ^a wagram@jinr.ru, ^b korenkov@jinr.ru, ^c nechav@jinr.ru

Целью проекта является моделирование облачной инфраструктуры Лаборатории информационных технологий в Объединенном институте ядерных исследований (ОИЯИ) и решение проблемы реализации эффективных алгоритмов для миграции виртуальных машин с точки зрения развертывания и использования виртуальных машин на серверах облачной инфраструктуры. Основная идея заключается в том, чтобы в каждый интервал времени распределить с как можно большей плотностью относительно задействованных ресурсов имеющиеся виртуальные машины между серверами; чтобы удовлетворить начальным требованиям по успешному выполнению данной задачи, было проведено сравнение существующих пакетов моделирования с целью выбора из них наиболее оптимально отвечающего нуждам проекта. Более того, начальная модель для дальнейшей оптимизации в соответствии с последующими требованиями задачи была реализована и протестирована. На текущей стадии выполнения проект ориентирован на исследование применения задачи об упаковке в контейнеры относительно вопроса миграции виртуальных машин и разработку схемы миграции, расширяющей текущую модель системы, для данной облачной инфраструктуры.

Ключевые слова: моделирование, SimGrid, миграция виртуальных машин, алгоритмы миграции, задача об упаковке в контейнеры

Работа выполнена при финансовой поддержке гранта РФФИ №15-29-07027

© 2016 Ваграм Сержикович Айриян, Владимир Васильевич Кореньков, Андрей Васильевич Нечаевский

Введение

В настоящее время, использование моделирования в отношении вычислительных ресурсов становится все более популярным. Моделирование позволяет обнаружить узкие места в существующей архитектуре центров обработки данных, проводить эксперименты с изменением топологии и заменой ресурсов до непосредственного вмешательства, моделировать функционирование пока что не существующих вычислительных инфраструктур. Это хороший способ для проверки предполагаемого решения проблемы без вмешательства в функционирование работающих объектов, которое может привести к выходу инфраструктуры из строя, потере и повреждению данных. Кроме того, пакеты моделирования позволяют намного проще и быстрее отлаживать и оценивать алгоритмы для управления ресурсами.

Рассматриваемая в данной статье проблема относится к области эффективного использования ресурсов. Основная идея заключается в размещении как можно большего количества виртуальных машин на выделенные ресурсы с целью использования освободившихся серверов для решения иных задач.

Основной целью проекта является разработка, испытание и внедрение эффективных эвристических алгоритмов (то есть, миграционной схема для виртуальных машин — планировщика), для высвобождения максимального количества вычислительных ресурсов. Эта статья посвящена моделированию облачной инфраструктуры, тестированию модели и определения дальнейших шагов исследования.

Пакеты моделирования

Для моделирования облачной инфраструктуры необходимо либо разработать свою собственную систему моделирования, что является крупной задачей само по себе, либо рассмотреть и выбрать один из существующих пакетов моделирования. Написано большое количество работ, посвященных сравнению и оценке различных пакетов моделирования, нами были использованы некоторые из них для сбора информации о четырех пакетах с открытым исходным кодом, которые являются одними из самых популярных для моделирования облачных вычислений: CloudSim, GreenCloud, iCanCloud и SimGrid (таблица 1).

Таблица 1: Сравнительные параметры пакетов моделирования CloudSim, GreenCloud, iCanCloud и SimGrid

	CloudSim	GreenCloud	SimGrid	iCanCloud
Язык программирования	Java	C++/OTcl	C, Lua, Java, Ruby	C++
Модель распространения	Исходный код	Исходный код / VM	Исходный код / binary	Исходный код
Стабильный релиз	02.05.2013	13.02.2016	13.10.2015	16.02.2015
Последний релиз	02.05.2013	13.02.2016	17.05.2016	16.02.2015
VM миграция	да	да	да	нет

Несмотря на большое количество возможностей, предлагаемых пакетами, их функциональность во многом одинакова, поэтому для сравнения использовалось только небольшое число параметров, среди них основная характеристика — VM миграция, реализованная в базовых функциях пакета [Wenhong, et al., 2015; Mihailov, Radchenko, 2014; Casanova, et al., 2014].

Хотя iCanCloud является мощным инструментом для моделирования вычислительных инфраструктур предлагающим: POSIX, большое количество коммуникационных протоколов

(UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, 802.11, MPLS, OSPF), поддержку систем хранения данных, графический интерфейс и учет затрат на поддержку инфраструктуры, у него отсутствует реализованная возможность миграции виртуальных машин в облаке, что является одним из основных его недостатков [Korenkov, Muravyev, Nechaevsky, 2014].

Кроме того, мы также отказались от использования CloudSim и GreenCloud по нескольким причинам. Так CloudSim имел неприемлемо давний срок релиза в последней стабильной версии и обновился лишь в 2016 г., поэтому мы решили отказаться от этого пакета [University of Melbourne, 2016]. Также GreenCloud был хорошим кандидатом с удобным *dashboard*, предоставляющим информацию о модели и симуляции в структурированном виде, было решено отказаться от идеи его использования из-за недостаточной документации и языка Ocaml, несомненно усложняющего поддержку модели в будущем [University of Luxembourg, 2013].

Таким образом, единственным пакетом моделирования, соответствующим нашим требованиям, является SimGrid. SimGrid представляет собой универсальный фреймворк моделирования, который позволяет симулировать поведение различных распределенных систем, таких как: грид, облака, HPC и P2P.

Верхний слой абстракции в SimGrid состоит из трех API, с помощью которых может быть реализован симулятор: MSG, SMPI и SIMDAG. В данном проекте используется API MSG. MSG позволяет пользователям описывать симуляцию как набор параллельных процессов, реализованных при помощи языков C, C++, Java, Lua или Ruby, и использовать вызовы MSG для имитации вычислений и коммуникационной активности модели. Механизмы для симуляции параллельных процессов в MSG реализуются как часть слоя под названием SIMIX, который представляет собой ядро, обеспечивающее управление процессами и синхронизацией. Каждая переменная состояния соответствует симулируемому действию, вычислению или передаче данных, и используется для того, чтобы симулировать последовательность действий с учетом времени выполнения. Ядро моделирования, то есть компонент, который симулирует выполнение действий в привязке к моделируемым ресурсам, называется SURF. Каждое действие определяется общим объемом выполненной и оставшейся работы — когда объем оставшейся работы достигает нуля, действие завершается, сообщая об этом соответствующей переменной состояния SIMIX. Такой подход к симуляции несет исключительно аналитический характер [Casanova, et al., 2014].

Модель облачной инфраструктуры

Облачная инфраструктура ИТ основана на OpenNebula [Baranov, Balashov, Kutovskiy, Semenov, 2016]. Облачные узлы (сервера) поддерживают виртуализацию при помощи OpenVZ и KVM. Каждый узел соединен с двумя 1 Гбит подсетями: частной, для обмена данными между облачных услуг и публичной, которая используется для взаимодействия с пользователями и внешними службами. Такая структура обеспечивает необходимый уровень сервиса. Все виртуальные машины и контейнеры инициализируются на облачных узлах [Baranov, Balashov, Kutovskiy, Semenov, 2015].

Физическое оборудование представлено в следующей конфигурации:

- 30 серверов с 4 ядрами CPU, 8 Гб оперативной памяти и 250 Гб HDD каждый;
- Серверы с 2 ядрами CPU, 4 Гб оперативной памяти и 800 Гб HDD каждая;
- 1 сервер с 4 ядрами CPU, 8 Гб оперативной памяти и 2330 Гб HDD;
- серверы с 12 ядрами процессора, 36 Гб оперативной памяти и 1000 Гб HDD каждая;
- 1 сервер с 24 процессорными ядрами, 24 Гб оперативной памяти и 1000 Гб HDD.

В SimGrid вычислительные ресурсы задаются с точки зрения характеристик вычислительных мощностей. Они соединены между собой посредством виртуальной сети, которая включает в себя связи и элементы маршрутизации, задаваемые характеристиками пропускной способности и задержек. Всем ресурсам могут быть также присвоены временные метки доступных зна-

чений вычислительных мощностей, включая, например, возможное время простоя [Casanova, et al., 2014].

Описание платформы реализовано в XML, и в качестве моделируемой инфраструктуры содержит единственную автономную систему (AS), состоящую из 39 узлов и пяти типов серверов, четыре типа HDD и четыре типа RAM; один маршрутизатор для подключения всех серверов внутри частной подсети, с одним типом связи 1 Гбит, образующей 780 вспомогательных сетевых маршрутов.

Проверка модели

Для проверки модели необходимо провести симуляционный эксперимент. Однако, так как текущая конфигурация облачной инфраструктуры не имеет каких-либо миграционных схем и процесс миграции VM выполняется вручную, в настоящее время не существует способа оценить модель статистически, симулируя ту же схему миграции и сравнивая полученные результаты в модели и в реальной конфигурации.

С учетом данной ситуации была предложена идея для запуска простой миграционной схемы, которую мы также могли легко оценить, чтобы проверить согласованность модели и отсутствие неопределенного поведения при симуляции процесса миграции.

С точки зрения процесса миграции, на базовом уровне обмена VM между серверами использовался стандартный алгоритм предварительного копирования, который является таковым по умолчанию для SimGrid (предполагается оптимизация алгоритма миграции после выполнения основных этапов данного проекта).

Алгоритм предварительного копирования, состоит из шести этапов [Clark, et al., 2005]:

- Pre-Migration. Определяется VM на физическом сервере А.
- Резервирование. Запрашивается перенос VM с сервера А на сервер Б.
- Итерационный предварительное копирование. Во время первой итерации, все страницы памяти передаются с А на Б. В последующих итерациях копируются только те страницы, в которые произошла запись данных во время выполнения алгоритма.
- Остановка и копирование. VM полностью приостанавливается на сервере А и сетевой трафик к данной машине перенаправляется на Б.
- Передача. Узел Б сообщает А, что образ VM успешно получен.
- Активация. Мигрировавшая VM активируется на Б.

В данном случае, миграционная схема представляет собой простой алгоритм, который начинает работу (моделирование) с N виртуальными машинами, каждой из которых присвоена случайная величина вычислительной нагрузки от 0 до 90 процентов виртуального процессора (VCPU), и которые случайным образом распределены между 39 серверами. Перед осуществлением первой миграции виртуальных машин на текущей итерации алгоритма все значения состояния модели сохраняются для дальнейшей оценки. Нагрузка на каждый виртуальный процессор остается неизменной до конца работы алгоритма, поскольку в противном случае трудно провести оценку все множество динамически меняющихся значений. На каждой итерации моделирования миграционная схема обнаруживает сервер с наименее загруженными ресурсами и перемещает на него все виртуальные машины с наименьшей вычислительной нагрузкой с учетом совместного избыточного использования ресурсов (overcommitment). Если какой-либо сервер бездействует по окончании текущей итерации, то данный сервер переводится в режим сна.

В таблице 2 показаны результаты моделирования, где t внутреннее время моделирования, полученное из SimGrid MSG. Каждая из трех симуляций была запущена 60 раз. Средние значения очень близки к индивидуальным полученным на каждом запуске результатам (присутствовали статистически незначимые колебания в силу случайности исходных данных). Результаты (ожидаемые) показывают, что несмотря на большое количество свободных серверов (sleep) число серверов, которые перегружены (starve) и страдают от нехватки ресурсов, очень высоко

по окончании каждой из симуляций, а также начиная с некоторого $500 < t_i < 10000$ множества каждых типов серверов стабилизируется и продолжение симуляции не показывает изменений в их соотношении. Однако, эта миграционная схема не предназначена для применения на практике, поскольку не обеспечивает доступность ресурсов для виртуальных машин (перегруженные сервера не способны удовлетворить требованиям по скорости вычислений).

Таблица 2: Результаты симуляции для 64, 128 и 256 виртуальных машин

	64 VM			128 VM			256 VM		
	OK	Starve	Sleep	OK	Starve	Sleep	OK	Starve	Sleep
t = 0	32	0	7	38	0	1	36	3	0
t = 300	20	1	18	31	2	6	34	5	0
t = 500	9	2	28	24	3	12	31	7	1
t = 10000	1	5	33	1	7	31	1	26	12

Заключение

Как видно из полученных результатов, необходим способ для эффективного размещения виртуальных машин по отношению к задействованным ресурсам при сохранении количества свободных серверов как можно большим. Это означает, что необходимо решить, какое количество используемых серверов и их вычислительной мощности достаточно для имеющегося множества виртуальных машин. Это проблема известна как задача об упаковке в контейнеры. Ее можно сформулировать следующим образом: n объектов должны быть размещены в контейнерах емкости L каждый. Объект I требует L_i единиц емкости; цель: определить минимальное количество контейнеров, необходимых для размещения всех n объектов [Xin, Zhuanzhuan, 2015]. Для виртуальных машин проблема является более сложной, поскольку этот вариант задачи предполагает совместное избыточное использование ресурсов различными VM, что известно как задача об упаковке виртуальных машин, где, в данном случае, необходимо оценить характеристики как серверов, так и запускаемых виртуальных машин, и провести их ранжирование [Sindelar, Sitaraman, Shenoy, 2011; Balashov, et. al, 2015].

Список литературы

- Aleksankov, S.* Models of live migration with iterative approach and move of virtual machines // Scientific and Technical Journal of Information Technologies, Mechanics and Optics, — 2015 — Vol. 15, No. 6. — P. 1098-1104.
- Balashov, N. et al.* JINR Cloud Service: Status and Perspectives // Trudy Instituta sistemnogo programirovaniya RAN. — 2015. — Vol. 27, No. 6. — P. 345-354.
- Baranov, A., Balashov, N., Kutovskiy, N., Semenov, R.* JINR cloud infrastructure evolution // Physics of Particles and Nuclei Letters. — 2016. — Vol. 13, No. 5 — P. 672–675.
- Baranov, A., Balashov, N., Kutovskiy, N., Semenov, R.* Cloud Infrastructure at JINR // Компьютерные исследования и моделирование. — 2015. — Vol. 7, No. 3. — P. 463-467.
- Casanova, H. et al.* Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms // Journal of Parallel and Distributed Computing. — 2014. — Vol. 74, No. 10. — P. 2899-2917.
- Clark, C. et al.* Live migration of virtual machines // Berkeley, CA, USENIX Association. — 2005. — P. 273-286.
- Korenkov, V., Muravyev, A., Nechaevsky, A.* Cloud Computing Simulation Packages // Системный анализ в науке и образовании. — 2014. — Issue 2.

- Lewis, R.* A General-Purpose Hill-Climbing Method for Order Independent Minimum Grouping Problems: A Case Study in Graph Colouring and Bin Packing // *Computers and Operations Research*. — 2009. — Vol. 36, No. 7. — P. 2295-2310.
- Mihailov, P., Radchenko, G.* Modeling and Performance Evaluation of Cloud Systems // *Bulletin of the South Ural State University*. — 2014. — Vol. 3, No. 3. — P. 109-123.
- Sindelar, M., Sitaraman, R., Shenoy, P.* Sharing-Aware Algorithms for Virtual Machine Colocation // San Jose, CA, s.n. — 2011. — P. 367–378.
- University of Luxembourg.* GreenCloud User Manual // [Online] Available at: <https://greencloud.gforge.uni.lu/ftp/greencloud-user-manual.pdf> — 2013.
- University of Melbourne.* Cloudslab/cloudsim. // [Online] Available at: <https://github.com/Cloudslab/cloudsim/releases> — 2016.
- Wenhong, T. et al.* Open-Source Simulators for Cloud Computing: Comparative Study and Challenging Issues // [Online] Available at: <https://arxiv.org/abs/1506.01106> — 2015.
- Xin, L., Zhuanzhuan, Z.* A Virtual Machine Dynamic Migration Scheduling Model Based on MBFD Algorithm // *International Journal of Computer Theory and Engineering*. — 2015. — Vol. 7, No. 4. — P. 278-282.

Modelling LIT Cloud Infrastructure at JINR and Evaluating the Model

V.S. Airiian^{1,2,4,a}, V.V. Korenkov^{2,3,b}, A.V. Nechaevskiy^{2,c}

¹Dubna State University, Dubna, Russian Federation

²Joint Institute for Nuclear Research, Dubna, Russian Federation

³Plekhanov Russian University of Economics, Moscow, Russian Federation

⁴Department of Informatics, Technical University of Munich, Munich, Germany

E-mail: ^a wagram@jinr.ru, ^b korenkov@jinr.ru, ^c nechav@jinr.ru

The goal of the project is modelling of LIT Cloud Infrastructure at the Joint Institute for Nuclear Research (JINR) and solving a problem of implementing effective algorithms for virtual machines migration regarding their creation, allocation and usage on the servers composing the cloud infrastructure. The main intention is to distribute virtual machines between the servers considering involved resources with the highest density possible in each period of time; to fulfil the initial requirements in respect of the successful implementation of this task a comparison between the existing modelling frameworks has been performed, in order to choose one of them optimally meeting the requirements of the project. Moreover, an initial model for further optimization regarding future requirements has been created and evaluated. On the current stage of implementation, the project is concentrated on studying applicability of the Bin Packing Problem to the virtual machines migration and developing a migration scheme extending the existing model of the cloud infrastructure.

Keywords: modelling, SimGrid, simulation, virtual machines, live migration, Bin Packing Problem

The work was supported by RFBR grant № 15-29-07027

© 2016 Vagram S. Airiian, Vladimir V. Korenkov, Andrey V. Nechaevskiy