

## Разработка схемы адаптивного конфигурирования загрузки виртуальных машин на основе аппликативного исчисления информационных процессов

С. В. Зыков<sup>\*,\*\*</sup>, Л. Д. Шумский<sup>\*\*\*</sup>, И. Е. Тарасов<sup>\*\*\*\*</sup>

<sup>\*</sup>НИУ «Высшая школа экономики»  
101000, Москва, ул. Мясницкая, 20

<sup>\*\*</sup>Национальный исследовательский ядерный университет МИФИ  
115409, Москва, Каширское шоссе, 31

<sup>\*\*\*</sup>IBS Group; 127018, Москва, ул. Складочная, 3, стр. 1

<sup>\*\*\*\*</sup>МИРЭА – Российский технологический университет  
119454, Москва, просп. Вернадского, 78

*e-mail: szykov@hse.ru, ilya\_e\_tarasov@mail.ru*

*Аннотация.* Рассматриваются типичные сценарии, при которых возникает потребность совместного использования ограниченных общесистемных ресурсов несколькими виртуальными машинами. Разработаны правила балансировки нагрузки в зависимости от информационных процессов, выполняемых виртуальными машинами. Вычисление производительности рассмотренных операций основано на типизированной модели, что дает возможность оценки общей производительности.

*Ключевые слова:* виртуальная машина, модель процесса, теория типов, аппликативное исчисление.

### 1. Введение

Целью работы является исследование ситуаций, в которых возникает потребность использовать общий ограниченный ресурс несколькими виртуальными машинами, и разработка правил балансировки нагрузки на эти ресурсы в зависимости от информационных процессов, выполняемых каждой из виртуальных машин (ВМ).

Одной из основных проблем при использовании виртуальных машин является вынужденное снижение производительности, по сравнению с использованием в основе системотехнической инфраструктуры (СТИ) классической архитектуры. Эта проблема вызвана вынужденным разделением ресурсов сервера между несколькими ВМ. Проблема снижения производительности при использовании ВМ, а также существенного увеличения сложности администрирования СТИ относится к числу

наиболее актуальных. Кроме технических ресурсов сервера, снижение производительности затрагивает также прочие ограниченные ИТ-ресурсы организации.

Необходимость балансирования ресурсов между несколькими виртуальными машинами, работающими на одних физических мощностях, отмечалась во множестве исследований [1–3]. Чаще всего для балансировки ресурсов предлагается использование методов, основанных на взаимодействии между основной машиной (host) и рабочими виртуальными машинами. К таким методам относятся: компрессирование памяти (memory compression), разделение контента (content-based sharing), «надувание шара» (ballooning), и замена страниц (page replacement) [4, 5]. Каждый из этих методов позволяет оперировать с ограниченным количеством параметров по каждому из ресурсов: могут анализироваться максимальная ожидаемая нагрузка (пиковая), средняя нагрузка для ВМ и текущее потребление ресурса. Как указывалось ранее [6], для лучшей работы механизма перераспределения ресурсов для кластера виртуальных машин необходимо предоставить информацию об ожидаемой нагрузке для некоторого горизонта планирования распределения ресурсов. Вторым существенным моментом для планирования является специфика выполняемых операций. Особенно это актуально для grid-решений [7], где вычислительная среда является существенно гетерогенной. В таком окружении производительность всего комплекса можно значительно увеличить, если выделить виртуальным машинам ресурсы, наиболее приспособленные для решения их задач.

## **2. Применение аппликативных моделей для конфигурирования виртуальных машин**

При решении задачи выбора конфигурации виртуальной машины необходимо ориентироваться на следующие ключевые показатели: 1) достаточно выделенных ресурсов для выполнения основных функций процесса; 2) достаточность выделенных ресурсов для обеспечения устойчивости инфраструктуры; 3) однородное распределение ресурсов по географическому и организационному делению; 4) минимализация затрат на используемые лицензии при их зависимости от аппаратных характеристик.

Классические средства масштабирования ресурсов виртуальных машин предполагают использование выделенной отдельной независимой виртуальной машины под каждую систему или класс систем. Для оптимизации выделения ресурсов (в первую очередь оперативной памяти) между несколькими виртуальными машинами на одном host (физическом сервере) может использоваться механизм, известный под названием ballooning. Этот механизм по работе гипервизора с оперативной памятью позволяет запустить на физическом сервере виртуальные машины, совокупная выделенная память которых превышает суммарный объем памяти данного сер-

вера. Достигается это за счет использования специального драйвера. Он позволяет «надуть шар» (balloon) внутри «гостевой» операционной системы (ОС), который захватывает физическую память, выделенную этой ОС, и отдает ее другим «гостевым» операционным системам, которые в ней нуждаются. Такой «шар» не позволяет «гостевой» ОС производить работу приложений с данной областью памяти, поэтому если им потребуется дополнительная память, будет использоваться «гостевая» область памяти для обмена.

В классическом механизме «надувания шара» драйвер, управляющий выделенной памятью, растет и сдувается постепенно, с учетом динамики ситуации. При этом информация о планируемом поведении операционной системы драйверу недоступна. Такая схема работы системы приводит к излишне консервативной работе адаптирующего алгоритма: если прогноз потребности в ресурсах виртуальной машины не определен однозначно и не реализован как скрипт драйвера, он не может однозначно планировать перераспределение ресурсов и выполняет балансировку только на основании актуальной информации. Поскольку приоритетным является поддержание работоспособности уже запущенных виртуальных машин, драйвер будет преимущественно распределять память в пользу уже неактивных процессов, ввиду чего будут перераспределены ресурсы, которые недавно использовались.

Таким образом, применение адаптивных механизмов к задаче конфигурации виртуальных машин оправдано, если позволяет избежать излишне консервативного поведения классических методов оптимизации. В случае использования таких адаптивных методов конфигурации общая архитектура системы принимает следующий вид (рис. 1).



Рисунок 1. Общий вид архитектуры системы

В данной архитектуре для каждой виртуальной машины, выполняющей те или иные бизнес-операции, выполняется сбор необходимой информации в режиме реального времени. Данная информация по определенным правилам передается драйверу адаптивной настройки, который консолидирует записи от всех узлов системы и с использованием прогностических моделей подготавливает варианты конфигурации ресурсов, которые передаются хост-виртуальной машине. Данные записи могут быть использованы хост машиной как непосредственно, так и в виде настроек для классических средств автоконфигурации.

### **3. Варианты применения механизма адаптивной конфигурации виртуальных машин**

С учетом предложенной схемы архитектуры могут быть рассмотрены следующие варианты применения механизма адаптивной конфигурации.

**Сценарий № 1** (рис. 1). Виртуальные машины, выполняющие процессы в разных часовых поясах. Стандартной ситуацией для крупных компаний является выделение серверных ресурсов для обеспечения работы информационных систем в различных географических точках. Недостатком такого подхода является то, что он не учитывает снижения и пики активности пользователей в различных зонах обслуживания. Обычно каждая ВМ рассчитывается на пиковые значения для своего региона и ее ресурсы, в моменты спада активности, никак не используются. Классические средства распределения ресурсов не применимы для данного класса задач, т. к. виртуальные машины различных регионов значительно отделены друг от друга и не управляются единым драйвером памяти. В таком случае применение адаптивного подхода будет оправдано для настройки перераспределения ресурсов от виртуальных машин, пользователи которых вышли из пиковой зоны к виртуальным машинам, испытывающим пиковые нагрузки. Этот подход применим для таких ресурсов, как вычислительные мощности центральной аналитической системы, доступ для выполнения запросов в центральной базе данных компании или ресурсы для обработки запросов от клиентских приложений пользователей.

**Сценарий № 2** (рис. 3). Выполнение критически важных процессов. В компании присутствуют системы различных классов критичности. Основной задачей инфраструктуры является поддержание работоспособности систем высококритичных, т. е. таких систем, без которых в краткосрочном промежутке времени невозможно функционирование предприятия. Выход из строя этих систем может повлечь невосполнимые потери для управления. Для таких систем должны использоваться специализированные серверные платформы и инфраструктурные уровни с полным многократным резервированием всех компонентов, в том числе с использованием резервных удаленных ЦОД. Однако высокая стоимость таких серверных решений

является ограничивающим фактором их использования. В таком случае применение адаптивного подхода к конфигурации виртуальных машин может обеспечить дополнительную надежность решения в части используемых серверных ресурсов.

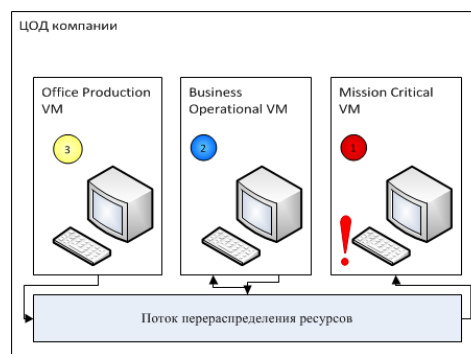
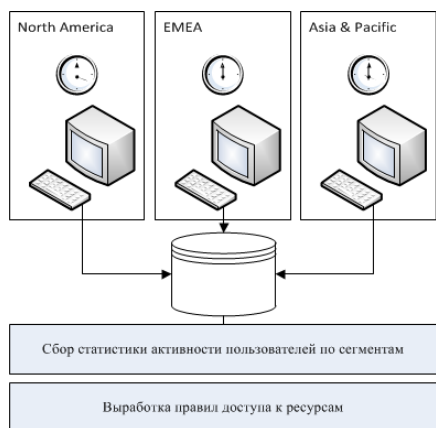


Рисунок 2. Схема применения сценария № 1      Рисунок 3. Схема применения сценария № 2

Адаптация распределения ресурсов обеспечивается как экспертными оценками критичности каждой системы для корректной работы предприятия, так и вычисляемыми метриками, основанными на обрабатываемых данных. Механизм адаптации позволит в зависимости от быстродействия выделенных критичных виртуальных машин оперативно перераспределять серверные ресурсы в их пользу. Подход применим для всех ресурсов сервера компании, таких как оперативная память, кэш процессора и дисковая память.

**Сценарий № 3** (рис. 4). Виртуальные машины, каскадно выполняющие общий бизнес-процесс. Правила лицензирования для целого ряда поставщиков ограничивают возможность установки своего программного обеспечения. Примером такого ограничения может служить количество физических ядер сервера, на котором развернута соответствующая виртуальная машина. Сумма таких ограничений приводит к ситуации, когда несколько виртуальных машин выполняют последовательные операции по одному бизнес-процессу. Ожидаемое поведение сервера в такой конфигурации заключается в перераспределении ресурсов от виртуальных машин, выполнивших определенный шаг процесса, к виртуальным машинам, которым передается основное выполнение операций. Однако, если каждая операция занимает незначительное время, но потребляет при этом значительный объем серверных ресурсов, классические средства перераспределения ресурсов показывают достаточно невысокую эффективность решения этой задачи. В этом сценарии использования применяется наиболее комплексный подход к адаптации перераспределения вы-

числительных ресурсов. Средство адаптивной конфигурации на основе модели выполняемого процесса составляет расписание передачи ресурсов определенной виртуальной машине и контролирует ход выполнения задач каждой ВМ, отслеживая этапы выполнения бизнес-процесса. Модель бизнес-процесса в данном случае не определяется методологически, а определяется в автоматическом режиме на основе журналов выполнения операций. Данные модели проверяются методологами организации только на предмет технических ошибок.

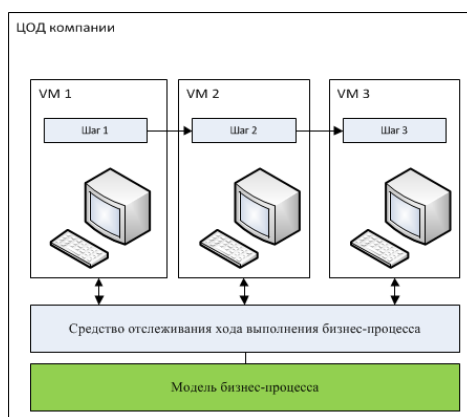


Рисунок 4. Схема применения сценария № 3

Данный подход применим для тех ресурсов, которые затрагиваются лицензионными ограничениями установленного на виртуальных машинах ПО. В отличие от ранее рассмотренных примеров, данный вариант использования предназначен не для увеличения надежности инфраструктурного решения, а для снижения общей стоимости владения ПО в масштабе предприятия и для увеличения средней доступности каждого компонента.

#### 4. Аппликативные подходы к моделированию информационных процессов

Следующим шагом совершенствования распределения ресурсов является использование информации об информационных процессах, которые выполняются в рамках данных виртуальных машин.

Такая информация может быть предоставлена драйверу «надувного шара» для модели информационного процесса, выполняемого виртуальной машиной. При этом модель процесса должна описывать: действия пользователей, ручные шаги в системе; автоматические действия, сложные вычисления или передачу данных; периоды бездействия, предусмотренные процессом.

С каждым типом действий настройки драйвера соотносится определенный класс доступности ресурсов операционной системы и на основании данных настроек переносятся общие ресурсы. Применение данного подхода позволяет расширить действие драйвера общих процессов не только на оперативную и постоянную память, но и на более специфичные ресурсы, такие как пропускная способность каналов связи, внешние устройства или операции прикладного ПО, количество которых регулируется лицензией.

Для управления моделями процессов предлагается использовать комбинированный подход: построение моделей процесса осуществляется автоматически, с использованием технологии анализа процессов (process mining). Система анализирует журналы деятельности пользователей в операционных системах и формирует модель процесса «как есть» (as-is). При этом в случае, если расхождения данной модели с заранее подготовленной ожидаемой схемой находятся в допустимом уровне, модель должна передаваться как непосредственная инструкция на выполнение драйверу хост-машины. Для непосредственного выполнения модели предлагается использовать технологию аппликативных вычислительных систем: модель процесса строится как терм  $\pi$ -исчисления, который может непосредственно выполняться реализацией абстрактной машины в драйвере.

## 5. Особенности применения моделей процессов в форме $\pi$ -исчисления

Для того, чтобы расширить функциональность предлагаемого подхода, возможно использование его для распределения нескольких видов ресурсов. При этом фиксация доступного перечня ресурсов приводит к новому ограничению и снижает универсальность метода балансировки ресурсов. Для устранения этого препятствия предлагается использовать типизированное исчисление. Статическая типизация позволяет моделировать поведение информационной системы, в связи с чем наличие проработанной системы типов существенно увеличивает выразительные способности модели.

Каждый тип исчисления соответствуют набору параметров настройки ВМ, который значим для некоторых выполняемых функций и влияет на оцениваемые характеристики. Типы исчисления могут быть получены как явно, так и при помощи конструкторов. При этом конструкторы типов позволяют явно указывать механизмы потребления определенных ресурсов в ходе выполнения связанных с ними процессов. Возможные виды подобных ресурсов включают:

- степень дискретности ресурса: насколько может быть разделен общий ресурс;
- латентность ресурса: время реакции ресурса на определенное событие;

- эластичность ресурса: время на восстановление ресурса после его высвобождения процессом;
- возможность совместного использования ресурса.

В рамках процессной модели типы исчисления присваиваются каждому действию процесса: т. е. каждой операции переноса данных.

Присвоение типа каналу приема или передачи интерпретируется как выполнение операции при некоторых настройках ВМ. Корректной является такая модель, что настройки, полученные из каждого типа, применимы и непротиворечивы. Благодаря тому, что вычисление производительности операции основано на типах модели, корректное присвоение типов всем объектам модели позволяет получить оценку общей производительности.

## 6. Заключение

Рассмотрен ряд типичных ситуаций, при которых возникает потребность использовать общий ограниченный ресурс несколькими виртуальными машинами. На основе сценарно-ориентированного подхода разработаны правила балансировки нагрузки на эти ресурсы в зависимости от информационных процессов, выполняемых виртуальными машинами. Благодаря тому, что вычисление производительности операции основано на типизированной модели, корректное присвоение типов позволяет получить оценку общей производительности.

## Литература

- [1] Melekhova A., Vinnikov V. Cloud and Grid Part I: Difference and Convergence // *Indian J. Sci. Technol.* 2015. Vol. 8. No. 29. DOI: 10.17485/ijst/2015/v8i29/86859
- [2] Melekhova A., Vinnikov V. Cloud and Grid Part II: Virtualized Resource Balancing // *Indian J. Sci. Technol.* 2015. Vol. 8. No. 29. DOI: 10.17485/ijst/2015/v8i29/86860
- [3] Poon W. -C., Mok A. K. Bounding the running time of interrupt and exception forwarding in recursive virtualization for the x86 architecture. Technical Report VMware-TR-2010-003, VMware, Inc., 3401 Hillview Avenue, Palo Alto, CA 94303, USA, 2010.
- [4] Melekhova A., Markeeva L. Estimating Working Set Size by Guest OS Performance Counters Means // *CLOUD Comput.* 2015. Vol. 48. P. 33–38.
- [5] Waldspurger C. A. Memory resource management in VMware ESX server // *ACM SIGOPS Oper. Syst. Rev.* 2002. Vol. 36. No. SI. P. 181–194.
- [6] Zykov S., Shumsky L. Application of Information Processes Applicative Modelling to Virtual Machines Auto Configuration // *Procedia Comput. Sci.* 2016. Vol. 96. P. 1041–1048.
- [7] Economics Paradigm for Service Oriented Peer-to-Peer and Grid Computing [Электронный ресурс] URL: <http://www.cloudbus.org/ecogrid/colostate.html> (accessed 16.04.2017).



**Авторы:**

Сергей Викторович Зыков — доктор технических наук, доцент, профессор Департамента программной инженерии, НИУ «Высшая школа экономики»; доцент Института интеллектуальных кибернетических систем, НИЯУ МИФИ

Леонид Дмитриевич Шумский — кандидат технических наук, начальник отдела интеграционных решений Департамента аналитических решений, IBS Group

Илья Евгеньевич Тарасов — доктор технических наук, профессор, профессор кафедры корпоративных информационных систем Института информационных технологий, МИРЭА — Российский технологический университет

---

## Designing Adaptive Load Configuration for Virtual Machines Based on Applicative Calculus for Information Process

S. V. Zykov<sup>\*\*</sup>, L. D. Shumskiy<sup>\*\*\*</sup>, I. E. Tarasov<sup>\*\*\*\*</sup>

<sup>\*</sup>Higher School of Economics, 20, Myasnitskaya Str., Moscow, 101000

<sup>\*\*</sup>National Research Nuclear University MEPhI, 31, Kashirskoe sh., Moscow, 115409

<sup>\*\*</sup>IBS Group, 3, Block 1, Skladochnaya Str., Moscow, 127018

<sup>\*\*\*</sup>MIREA-Russian Technological University, 78, Vernadskogo prosp., Moscow, 119454

e-mail: szykov@hse.ru, ilya\_e\_tarasov@mail.ru

**Abstract.** This paper examines typical scenarios that require limited system resources shared by several virtual machines. Load balancing rules were developed; these are dependent on the information processes performed by the virtual machines. The productivity of these operations is based on types that allow assessment of the entire system productivity.

**Keywords:** virtual machine, process model, type theory, applicative calculus

## References

- [1] Melekhova A., Vinnikov V. (2015) *J. Sci. Technol.* **8**(29). DOI: 10.17485/ijst/2015/v8i29/86859
- [2] Melekhova A., Vinnikov V. (2015) *J. Sci. Technol.* **8**(29). DOI: 10.17485/ijst/2015/v8i29/86860
- [3] Poon W. -C., Mok A. K. (2010) Bounding the running time of interrupt and exception forwarding in recursive virtualization for the x86 architecture. Technical Report VMware-TR-2010-003, VMware, Inc., 3401 Hillview Avenue, Palo Alto, CA 94303, USA.
- [4] Melekhova A., Markeeva L. (2015) *CLOUD Comput.* **48**:33–38.
- [5] Waldspurger C. A. (2002) *ACM SIGOPS Oper. Syst. Rev.* **36**(SI):181–194.
- [6] Zykov S., Shumsky L. (2016) *Procedia Comput. Sci.* **96**:1041–1048.
- [7] Economics Paradigm for Service Oriented Peer-to-Peer and Grid Computing. URL: <http://www.cloudbus.org/ecogrid/colostate.html> (accessed: 16.04.2017).