

ПЛАНИРОВАНИЕ ЗАДАЧ ДЛЯ ОБЛАЧНЫХ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ

П.Н. Полежаев, Ю.А. Ушаков, А.Е. Шухман

Оренбургский госуниверситет

На концептуальном уровне описывается подход к планированию вычислений в облачных центрах обработки данных с использованием программно-конфигурируемых сетей. Предлагается вариант развития облачной управляющей системы OpenStack за счет эффективного планирования задач (групп виртуальных машин), путем их распределения по топологически-близким вычислительным узлам, а также за счет управления потоками данных между ними с целью их топологической локализации и снижения сетевой конкуренции.

Введение

Облачные вычислительные системы стали стандартом де-факто для многих прогрессивных областей сферы информационных технологий. Российские и иностранные компании используют их с целью размещения своих научных и бизнес-приложений, что позволяет им избежать расходов на создание и поддержание функционирования собственных центров обработки данных (ЦОД). С другой стороны, владельцы облачных вычислительных ЦОД путем консолидации вычислительных ресурсов и систем хранения данных (СХД) способны снизить совокупную стоимость владения ИТ-инфраструктурой за счет обслуживания значительного количества клиентов, а также использования эффективных технических средств планирования и балансировки нагрузки, управления пересылкой данных в сети, интеграции нескольких территориально разрозненных сегментов ЦОД.

Существующие сетевые протоколы облачных вычислительных ЦОД Fiber Channel, Infiniband, а также традиционный Ethernet, имеют ограниченные возможности по управлению трафиком и QoS. Усовершенствованные варианты протокола Ethernet – Converged Enhanced Ethernet и Cisco Data Center Ethernet включают расширения по управлению потоками на основе приоритетов, разделению пропускной способности, управлению перегрузками и логическим состоянием полос передачи данных, по обеспечению передачи без потерь, а также по одновременному использованию нескольких параллельных путей передачи данных между узлами. Основные недостатки данных решений – сложная децентрализованная схема управления потоками данных, основанная на множестве закрытых протоколов, значительная стоимость сетевого оборудования, сложность его расширения. Отдельно следует заметить, что данные решения используют реактивную схему управления потоками, когда решения по коммутации принимаются во время передачи потока.

Весьма перспективным направлением технологий вычислительных ЦОД является использование программно-конфигурируемых сетей (ПКС – Software Defined Network). Принципы ПКС впервые возникли в исследовательских лабораториях Стэнфорда и Беркли [1], и в настоящее время развиваются в рамках консорциума Open Network Foundation и европейского проекта OFELIA [2]. Известен положительный опыт компаний Google и Amazon по внедрению ПКС в облачных ЦОД.

В основе подхода ПКС лежит возможность динамически управлять пересылкой данных в сети с помощью открытого протокола OpenFlow. Все активные сетевые устройства объединяются под управлением сетевой операционной системы (СОС), которая обеспечивает приложениям доступ к управлению сетью. Сетевые контроллеры могут быть централизованными, использовать общие абстракции для пересылки пакетов.

За счет управления пересылкой данных в сети использование ПКС в облачных ЦОД позволяет реализовать схемы одновременной многопутевой передачи данных, управление потоками на основе приоритетов, виртуализацию сети, обеспечить QoS, эффективно распределить нагрузку на сеть. Открытость стандарта OpenFlow и вынесение логики управления в отдельный контроллер упрощает программное и аппаратное обеспечение активного сетевого оборудования, что позволит в будущем производителям снизить его стоимость, а следовательно уменьшатся затраты на создание ЦОД. Централизация и открытость средств управления ПКС позволяет гибко и эффективно адаптировать работу ЦОД под возникающие потребности бизнеса, что ускоряет внедрение инноваций и обеспечивает конкурентоспособность компаний.

В настоящее время отсутствуют решения на основе ПКС для облачных вычислительных ЦОД, включающих СХД, направленных на управление потоками данных задач с целью их топологической локализации и снижения сетевой конкуренции между ними, что позволит повысить эффективность работы облачного вычислительного ЦОД. Это определяет новизну настоящей работы.

Наличие потребности компаний в создании эффективных облачных вычислительных ЦОД, включающих СХД, на основе ПКС, перспективный характер технологии ПКС, а также отсутствие готовых решений в этой области являются существенными факторами, определяющими актуальность проведения данных исследований.

1. Недостатки управляющей системы OpenStack

Наиболее распространенным открытым программным обеспечением для управления облачными вычислительными ЦОД является OpenStack [3]. Его популярность, прежде всего, обусловлена открытостью, документированностью, наличием исходных кодов, а, следовательно, возможностью адаптации его компонент под произвольные архитектуры вычислительных систем, в отличие, например, от узко ориентированных закрытых решений VMware.

OpenStack позволяет создавать многоарендные архитектуры IaaS облачных вычислительных ЦОД, когда вычислительные и сетевые ресурсы разделяются между несколькими пользователями-арендаторами. Каждому арендатору выделяется виртуализованные ресурсы в виде группы виртуальных машин, связанных виртуальной сетью. Пользователь вправе сам настраивать конфигурацию виртуальных машин, включая параметры процессора, памяти, загружаемый образ с операционной системой, программами и данными. Хранение образов виртуальных машин обеспечивается компонентой Glance, а данных – облачным хранилищем Swift.

OpenStack был выбран в качестве основы для построения собственной управляющей системы облачного вычислительного ЦОД с использованием ПКС. Его наиболее существенным недостатком, требующим доработки, является алгоритм планирования задач (групп виртуальных машин).

В существующем реализованном варианте данный алгоритм подбирает подходящие по конфигурации физические узлы для размещения виртуальных машин, руководствуясь их интегральными характеристиками. Интегральная характеристика узла представляет собой линейную свертку его динамических характеристик (host capabilities), взятых с некоторыми весовыми коэффициентами. Алгоритм планирования ранжирует

узлы в порядке убывания значения интегральной характеристики и выбирает запрашиваемое число первых узлов. Данный процесс в терминологии OpenStack носит название “Host weighting”. Очевидным недостатком данного подхода является недо-груженность вычислительных ресурсов вследствие фрагментации расписания из-за нерационального распределения виртуальных машин в соответствии с их запросами к физическим ресурсам.

Еще одним недостатком OpenStack является отсутствие планирования сетевого трафика между виртуальными машинами. Действительно, когда известно расположение виртуальных машин пользователя, необходима локализация сетевого трафика между ними, направленная на снижения сетевой конкуренции. Конкуренция возникает из-за очередей пакетов в коммутаторах, отправляемых виртуальными машинами различных групп.

Другой недостаток OpenStack – отсутствие гарантий по пропускной способности виртуальной сети, соединяющей виртуальные машины. Нет возможности задания минимально-гарантированной пропускной способности для каждой сетевой связи, которая бы обеспечивалась сетевой частью облачного вычислительного ЦОД.

Авторы планируют устранить данные недостатки за счет использования адаптированных вариантов ранее разработанных алгоритмов планирования задач для кластерных вычислительных систем [4], ПКС на основе протокола OpenFlow, а также сетевых средств обеспечения QoS.

2. Суть предлагаемого решения по развитию OpenStack

На рис. 1 приведена концептуальная схема планирования задач, предлагаемая в рамках настоящей работы. Серым цветом обозначены элементы, разрабатываемые авторами.

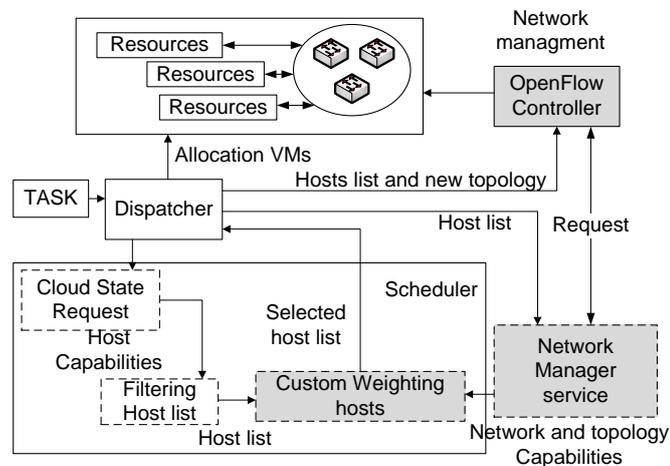


Рис. 1. Концептуальная схема планирования в улучшенном варианте системы OpenStack

Опишем основные элементы схемы.

1. Вычислительные узлы, связанные с помощью коммутаторов с поддержкой OpenFlow. Использование данных коммутаторов позволяет управлять потоками данных групп виртуальных машин с целью их топологической локализации и снижения сетевой конкуренции между ними.
2. Dispatcher представляет собой диспетчер облачной вычислительной системы, решающий задачу управления и планирования виртуальными машинами.
3. OpenFlow Controller – контроллер протокола OpenFlow, представляющий собой центральную компоненту ПКС. Сосредотачивает в себе всю программную логику управления маршрутизацией пакетов. Используется для формирования таблиц

коммутации в OpenFlow-коммутаторах на основе рассчитанных потоков данных между назначенными на физические узлы виртуальными машинами планируемых групп. Также используется для сбора сетевой статистики с коммутаторов OpenFlow и передачи ее элементу Network Manager Service.

4. Network Manager Service – служба, реализующая механизмы сбора статической и динамической информации об облачном вычислительном ЦОД. Включает обнаружение топологии сети на основе протокола LLDP, получение статистики от контроллера OpenFlow, сбор информации о загрузке вычислительных ядер и памяти узлов с помощью SNMP.
5. Scheduler – планировщик задач, который на основе запроса пользователя на выделение ресурсов для группы виртуальных машин, а также сведений о состоянии облачной вычислительной системы, подбирает оптимальные вычислительные узлы (hosts) для запуска виртуальных машин.

Более подробно схема работы планировщика изображена на рис. 2. Предполагается выполнение следующих шагов в процессе планирования:

1. Получение сведений о состоянии вычислительных узлов. В этом шаге отличие от стандартного планировщика OpenStack в том, что используется полный список всех узлов облачной вычислительной системы.
2. Получение сведений о состоянии сети, включая ее статические и динамические характеристики.
3. Объединение всей полученной информации в виде взвешенного неориентированного графа системы. Его вершинами являются узлы и коммутаторы, ребрами – сетевые связи между ними. Вся полученная информация описывается в виде весов вершин и ребер.
4. Выбор оптимального подграфа. На данном шаге предлагается использовать модифицированные варианты ранее разработанных авторами алгоритмов планирования Backfill MDM и Backfill SDM, которые учитывают топологию вычислительной системы, а также сетевую конкуренцию. Более подробно о них можно прочитать в работе [4].
5. Для совместимости с существующим кодом диспетчера OpenStack выбранные узлы записываются в список, в котором каждому элементу приписывается фиктивный интегральный вес.

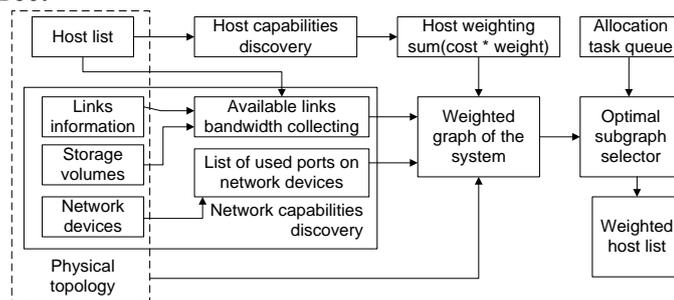


Рис. 2. Концептуальная схема работы планировщика

Также будет частично дописана компонента Dispatcher системы OpenStack, в ней планируется реализовать передачу сведений контроллеру OpenFlow о назначаемой группе виртуальных машин, чтобы тот предварительно (до запуска виртуальных машин на узлах) установил правила коммутации в таблицы соответствующих коммутаторов. Проактивный (предварительный) характер задания правил позволит исключить задержку на установление записей по каждому потоку, которая имела бы место при реактивной схеме (когда коммутатор при получении первого пакета потока в случае отсутствия правил передает его на анализ контроллеру).

Если на базе группы виртуальных машин разворачивается вычислительный кластер (HPC on cloud), то топологически-близкое размещение виртуальных машин, а также использование эффективной маршрутизации, позволит снизить время выполнения коммуникационно-интенсивных параллельных задач, решаемых на данном кластере. Тот же результат будет иметь место (снижение времени вычислений, снижение времени отклика) для прочих вычислительных приложений, выполняющихся на виртуальных машинах.

Заключение

На концептуальном уровне описан подход к планированию вычислений в облачных центрах обработки данных с использованием программно-конфигурируемых сетей. Предложен вариант развития облачной управляющей системы OpenStack за счет эффективного планирования задач (групп виртуальных машин), путем их распределения по топологически-близким вычислительным узлам, а также за счет управления потоками данных между ними с целью их топологической локализации и снижения сетевой конкуренции. В ближайшем будущем планируется реализация данного подхода в рамках облачного вычислительного ЦОД, создаваемого в Оренбургском государственном университете.

Исследования поддержаны федеральной целевой программой «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» (госконтракт №07.514.11.4153) и РФФИ (проект №12-07-31089).

Литература

1. Смелянский Р.Л. Проблемы современных компьютерных сетей // Труды XIX Всероссийской научно-методической конференции «Телематика 2012». – СПб: СПбГУ ИТМО, 2012. – Т. 1. – С. 371–375.
2. OFELIA: OpenFlow in Europe – [<http://www.fp7-ofelia.eu/>].
3. OpenStack Open Source Cloud Computing Software – [<http://www.openstack.org/>].
4. Полежаев П.Н. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора // Параллельные вычислительные технологии (ПАВТ'2010): Труды международной конференции. – Челябинск: ЮУрГУ, 2010. – С. 287–298.