Храпов Николай Павлович

Институт Проблем Передачи Информации РАН, инженер, nkhrapov@gmail.com

Интеграция облачных технологий и грид-систем из персональных компьютеров в контексте обучения и повышения эффективности

КЛЮЧЕВЫЕ СЛОВА:

Облачные технологии, грид-системы, программирование, распределенные вычисления, обучение, персональные компьютеры.

АННОТАЦИЯ:

В статье рассматриваются вопросы, связанные с интеграцией систем добровольных вычислений и облачных технологий. Исследуется актуальность данной интеграции в контексте образования и повышения эффективности использования ресурсов. Проводится сравнительный анализ различных подходов к данной интеграции.
Также в статье приводится описание прототипа интегрированной инфраструктуры и описание различных методов решения технических проблем.

Технологии облачных вычислений в настоящее время приобретают все большую популярность. Внедрение данных технологий в различных областях позволяет автоматизировать процесс развития инфраструктуры и повысить эффективность использования трудовых и вычислительных ресурсов.

В тоже время для решения глобальных вычислительных задач все чаще привлекаются ресурсы добровольных вычислений. Основная идея добровольных вычислений состоит в том, что владелец персонального компьютера предоставляет свою машину в свободное от работы время для решения научных задач. Обмен данными для вычислений осуществляется автоматически посредством сети Интернет. Современные системы организации добровольных вычислений позволяют использовать ресурсы персонального компьютера для научных вычислений таким образом, чтобы это не препятствовало использованию машины по прямому назначению в качестве домашнего или офисного компьютера. Основная сфера применения добровольных вычислений – это научные проекты, требующие вычислительный ресурс и имеющие практическую пользу для человечества.

Добровольные вычисления обладают наилучшим соотношением цена/мощность вычислительного ресурса, что делает данную технологию привлекательной для проведения научных расчётов. Технологии добровольных вычислений (далее грид-систем из персональных компьютеров – ГСПК) позволяют создавать масштабные вычислительные инфраструктуры. В частности, наиболее популярный в настоящее время в мире проект добровольных вычислений SETI@HOME насчитывает более 2 млн. компьютеров добровольных участников. Наиболее известные российские проекты добровольных вычислений: SAT@HOME[1], GERASIMA@HOME[2], OPTIMA@HOME[3].

Растущая популярность добровольных вычислений делает актуальным вопрос о подготовке специалистов в данной области. Данная подготовка предполагает проведение практических занятий. Проведение практических занятий в свою очередь требует создания учебной инфраструктуры. Внедрение технологий облачных вычислений для организации учебных полигонов позволяет автоматизировать процесс

создания, развертывания и настройки макетов учебных серверов. Принципы обучения грид-системам из персональных компьютеров изложены в [4], общие идеи применения облачных технологий в образовании – в [5].

Еще одним важным преимуществом интеграции облачных технологий и ГСПК является возможность существенно повысить эффективность использования имеющихся вычислительных ресурсов в качестве серверных платформ и вычислительных узлов.

Темой данной работы является исследование методов интеграции грид-систем из персональных компьютеров и преимуществ, которые данная интеграция дает в сфере образования и повышения эффективности использования ресурсов. В процессе работы был проведён сравнительный анализ технологий облачных вычислений и ГСПК на предмет взаимной интеграции. После проведения сравнительного анализа был разработан прототип гибридной инфраструктуры.

Технологии облачных вычислений позволяют использовать ресурсы удаленной вычислительной машины как сервисы. По типу используемых ресурсов технологии облачных вычислений подразделяются на три основных группы:

Software As A Service (SaaS) – технология облачных вычислений, позволяющая использовать программное обеспечение, установленное на удаленном сервере. Наиболее известные примеры SaaS – это браузерные текстовые и графические редакторы, а также сервисы, позволяющие автоматизировать процесс создания типовых сайтов (например, интернет-магазинов). Интеграция ГСПК и SaaS предполагает создание системы, позволяющей по запросу клиента запускать типовой проект добровольных вычислений на основании некоторого шаблона. Технология SaaS предполагает, что несколько клиентов способны использовать одинаковый экземпляр программного обеспечения, установленный на одной физической или виртуальной машине. Сильной стороной данного подхода является сокрытие от клиента (т.е. администратора проекта добровольных вычислений) инфраструктурных особенностей ГСПК, т.е. предоставление ему проекта добровольных вычислений «под ключ». Отсюда же следует, что применение данной технологии для создания именно учебных инфраструктур является неактуальным, так как скрывает от студента те инфраструктурные особенности, работе с которыми он должен обучаться. К другой слабой стороне данного подхода можно отнести отсутствие гибкости настройки проекта и слабая изоляция пользовательского пространства для нескольких проектов.

Platform As A Service (PaaS) – технология облачных вычислений, позволяющая использовать удаленную виртуальную машину как некоторый сервис. Основное отличие PaaS от SaaS состоит в том, что в PaaS пользователь получает в свое распоряжение платформу (в большинстве случаев, виртуальную машину с установленной на ней операционной системой и набором базового ПО) для запуска в рамках данной платформы своего прикладного ПО. Наиболее известными примерами данной технологии являются сервисы Amazon, Google App Engine и Microsoft Azure, предоставляющие пользователю серверную платформу для загрузки своего программного кода. Также к PaaS можно отнести сервисы, предоставляющие услугу Remote Desktop. Дистрибутив наиболее популярной в настоящее время ГСПК BOINC по умолчанию включает в себя возможность функционирования нескольких проектов на одной машине с одинаковым IP-адресом, но различными URL. Относительно SaaS данный подход предполагает для нескольких клиентов функционирование одинакового инфраструктурного ПО ГСПК, но различного управляющего программного обеспечения.

Infrastructure As A Service (IaaS) – технология облачных вычислений, позволяющая на базе одной физической вычислительной инфраструктуры создавать и запускать другие вычислительные инфраструктуры. Многие современные реализации SaaS и PaaS являются надстройками над технологией IaaS. Интеграция IaaS и ГСПК предоставляет пользователю наибольшую гибкость в настройке собственного проекта добровольных вычислений, а также предоставляет ему возможность использовать для данной цели несколько вычислительных машин. К слабым сторонам данного подхода можно отнести необходимость клиенту-администратору проекта работать с инфраструктурными особенностями ГСПК.

В качестве создания прототипа инфраструктуры был выбран способ интеграции IaaS и ГСПК по следующим причинам:

- максимальная гибкость в настройке ПО;
- данный подход является наиболее актуальным при создании именно учебных инфраструктур, так как способен предоставить студенту полный монопольный доступ к изучаемым компонентам;
- доступ ко всем системным компонентам открывает широкие возможности для настройки и повышения эффективности использования ресурсов;
- данный подход является универсальным, и методы, отработанные при создании прототипа гибридной системы могут быть использованы для интеграции других систем распределённых вычислений с облачной инфраструктурой.

Наиболее распространенными на данное время системами IaaS являются:

- eucalyptus;
- openStack;
- openNebula;
- Xen Cloud Platform.

Все перечисленные системы имеют примерно одинаковую базовую функциональность. При проведении сравнительного анализа выбор пал на систему XCP, т.к. данная платформа является наиболее отказоустойчивой, имеет наибольшее распространение, имеет наиболее обширную документацию и является бесплатной. При выборе реализации ГСПК для создания прототипа гибридной инфраструктуры была отобрана технология BOINC, т.к. данная технология является наиболее универсальной и распространенной. Подготовка специалистов работе с данной платформой является наиболее актуальной.

Методы и подходы, используемые при интеграции ГСПК и облачных технологий имеют универсальный характер и могут быть использованы для создания тренинговых инфраструктур для обучения системам распределенных вычислений другого типа. При разработке прототипа интегральной инфраструктуры возникли следующие сложности.

Методы управления облачной инфраструктурой Xen Cloud Platform.

Разработчиками облачного ПО Xen Cloud Platform предусмотрено три основных метода управления облачной инфраструктурой, данные подходы возможно использовать совместно:

- **XEN API** является наиболее низкоуровневым средством управления инфраструктуры XCP. Представляет из себя библиотеку функций, вызываемых из управляющей программы, выполняющей различные административные действия как на локальной, так и на удаленных машинах.
- **XE Command Line Interface (CLI) –** написанная на XEN API утилита командной строки, позволяющая выполнять административные действия как на своей, так и на удаленной машине, имеет версию для ОС Windows. Утилита XE предоставляет примерно такие же возможности, как и XEN API. XEN API удобнее использовать при разработке собственного управляющего инфраструктурой ПО. XE удобен при разработке работающих на машинах облака управляющих скриптов.
- **XEN Center** является наиболее высокоуровневым средством управления облачной инфраструктурой. Представляет собой графическую программу, функционирующую под управлением ОС Windows. Использование XEN-Center является наиболее простым в использовании и наименее функциональным способом управления инфраструктурой. Данный способ предполагает запуск виртуальных машин пользователем непосредственно через GUI и исключает автоматизацию управления и запуска виртуальных машин. Программа XenCenter способна функционировать только в среде Microsoft Windows, для осуществления управления посредством Unix-подобной среды имеется open source-аналог XenManager.

Использование облачной инфраструктуры предполагает как автоматизированное управление, реализованное на стороне инфраструктуры, так и прямое управление администратором облачной инфраструктуры со своего персонального компьютера. При создании прототипа для автоматизированного управления на стороне облачной инфраструктуры использовался XE CLI. Для прямого управления и отображения состояния – XEN Center.

Настройка виртуальных сетевых интерфейсов.

Использование технологии IaaS предполает создание новых виртуальных машин на основе некоторого, заранее приготовленного шаблона. Допустим, необходимо создать 30 BOINC-серверов. Для этого создается новая серверная виртуальная машина, на которую устанавливается необходимый набор системного программного обеспечения. Далее из данной виртуальной машины создается шаблон, затем на основе шаблона происходит клонирование нескольких виртуальных машин. При данном подходе возникает проблема предоставления статических адресов клонированным виртуальным машинам. Существует несколько подходов к решению данной проблемы:

- использование предустановленного в шаблоне Xen-агента, который способен взаимодействовать с инфраструктурой, сделать необходимые локальные настройки или передать в управляющий центр полученные посредством dhcp настройки. Используемый Xen-агент взаимодействует с компонентами инфраструктуры посредством технологии Xen-API.
- настройка необходимым образом dhcp-сервера для выдачи нужного IP-адреса на основе MAC-адреса. При создании виртуальной машины указывается необходимый MAC-адрес для дальнейшего предоставления нужного IP-адреса. Свободный IP-адрес определяется управляющим

центром посредством осуществления пинга. Внутри шаблона виртуальной машины используется инфраструктурно-независимый агент, который выполняет все необходимые настройки ПО на основе полученного посредстов DHCP IP-адреса.

Для реализации прототипа интегральной инфраструктуры был выбран второй способ как не использующий инфраструктурно-зависимых компонентов внутри виртуальной ОС, а по данной причине более универсальный. Если речь идет о создании именно учебной инфраструктуры, то в качестве шаблона использовался чистый образ, содержащий операционную систему Linux Debian 6.0 (серверное программное обеспечение BOINC изначально спроектировано для работы под управлением данной операционной системы). Для операционной системы Debian настройки DHCP расположены в файле /etc/dhcp/dhclient.conf. В данном файле запрашиваемые сетевые параметры приведены после ключевого слова request. После получения сетевых параметров по DHCP автоматически запускается скрипт /etc/dhcp/dhclient-exit-hooks.d/hostname, который производит настройку полученного по DHCP доменного имени:

```
#!/bin/bash
if [ "$interface" != "eth0" ]
then
return
fi
echo $new_host_name > /etc/hostname
hostname $new_host_name
echo "Hostname changed : $new_host_name"
```

Реализация инфраструктурных элементов

Для автоматизации управления разрабатываемой инфраструктуры был разработан набор управляющих скриптов. Данные скрипты предназначены для запуска на одной из машин облачного кластера.

Основной скрипт, производящий сканирование состояния инфраструктуры и осуществляющий запуск виртуальных машин приведен в листинге 1.

```
#!/bin/bash
network='188.93.105'
first_ip='140'
last_ip='170'
echo "First IP: $network.$first_ip"
echo "Last IP : $network.$last_ip"
echo $#
null=0
count=1
if [ "$#" -gt "$null" ]
then
  if [ "$1" -gt "$count" ]
  then
```

```
    count=$1
  fi
fi
echo count=$count
i=0
while [ "$count" -gt "$i" ]
do
  let "i=i + 1"
  echo $i;
done
i=0
for current_ip in `seq $first_ip $last_ip`
do
  if ping -c 1 -s 1 -W 1 "$network.$current_ip" > buf
  then
    echo "$network.$current_ip is used"
  else
    if [ "$i" -lt "$count" ]
    then
      mac=`./boinc_ip_to_mac $current_ip`
      dn=`./boinc_ip_to_dn $current_ip`
      dn=_$dn
      ./make-boinc-machine $mac  _$network.$current_ip$dn
      let "i = i + 1"
      echo "new VM debian 6.0 ip: $network.$current_ip"
    else
      echo "$network.$current_ip is free"
    fi
  fi
done
rm buf
```

Листинг 1. Базовый скрипт осуществляющий запуск необходимого количества виртуальных машин

Для создания непосредственно виртуальной машины используется скрипт make-boinc-machine, который в качестве аргументов использует MAC и IP-адреса создаваемой машины:

```
#!/bin/bash
vm_uuid_buf=`xe vm-install template=debian_6.0_clean new-name-label=BOINC_$2`
echo $vm_uuid_buf
xe vif-create device=0 network-uuid=afba88f3-9ee4-2172-715a-7756434ae4ab vm-uuid=$vm_uuid_buf mac=$1 device=0
xe vm-start vm=BOINC_$2
```

Листинг 2. Скрипт, осуществляющий запуск отдельной виртуальной машины

Здесь используются три основных команды утилиты xe:

Vm- install – команда создающая виртуальную машину на основе имеющегося шаблона.

Vif-create – команда, создающая виртуальный сетевой интерфейс для указанной виртуальной машины.

Vm-start – команда, осуществляющая запуск виртуальной машины.

Заключение

Дальнейшее развитие учебной облачной инфраструктуры предполагает использование для создания учебных инфраструктур различного типа. Во-первых, возможно использование для создания учебных полигонов для прочих реализаций грид-систем. Во-вторых, это использование облачной инфраструктуры для создания виртуальных кластеров различного типа. В третьих, возможно использование облачных технологий для обучения технологиям сервисных грид-систем. Дальнейшее развитие учебной облачной инфраструктуры предполагает. создание аналогичных виртуальных макетов для обучения информационным технологиям другого типа. В частности посредством облачных технологий возможно создавать различные учебные виртуальные кластеры. Кроме того, облачные технологии возможно внедрять при обучении грид-системам сервисного типа.

Литература

1. О.С. Заикин, М.А. Посыпкин, А.А. Семѐнов, Н.П. Храпов. Опыт организации добровольных вычислений на примере проектов OPTIMA@HOME и SAT@HOME.Вестник Нижегородского университета им. Н.И. Лобачевского, 2012, No5(2), с.340-347.
2. О.С. Заикин, М.А. Посыпкин, А.А. Семёнов, Н.П. Храпов. Организация добровольных вычислений на платформе BOINC на примере проектов OPTIMA@home и SAT@home // CAD/CAM/CAE Observer # 3 (71) / 2012. С. 87-92.
3. Vatutin E.I., Titov V.S. Voluntary distributed computing for solving discrete combinatorial optimization problems using Gerasim@home project // Distributed computing and grid-technologies in science and education: book of abstracts of the 6th international conference. Dubna: JINR, 2014. PP. 60–61. ISBN 978-5-9530-0387-2.
4. Храпов Н.П. Рабочий инструментарий и методы обучения для грид-систем из персональных компьютеров. // Сборник избранных трудов VI Международной научно-практической конференции "Современные информационные технологии и ИТ-образование" 2011г. с 966 - 972.
5. Афзалова А.Н., Голицина И.Н. Использование облачных технологий в образовательном процессе. Образовательные технологии и общество. Выпуск № 2 / том 17 / 2014г. с 450-460.