

Сервис-ориентированная архитектура доступа к ресурсам высокопроизводительных вычислительных систем

И.В. Бойченко, С.В. Корытников, А.В. Лежанкин

Томский университет систем управления и радиоэлектроники

В силу архитектурных особенностей применение ВПВС требует особых моделей программирования, что увеличивает время и стоимость разработки ПО. Организация доступа к ресурсам ВПВС в виде набора информационных сервисов повышает эффективность разработки прикладного ПО и информационных систем за счет повторного использования уже существующих решений и ослабления низкоуровневых связей с программно-аппаратными платформами вычислительных систем.

Высокопроизводительные вычислительные системы (ВПВС) в силу высокой стоимости являются разделяемым ресурсом на крупном предприятии или в научном центре. При этом в силу архитектурных особенностей применение ВПВС требует особой модели программирования. Общеизвестно, что программное обеспечение (ПО) для суперкомпьютеров и вычислительных кластеров является гораздо более трудоемким в разработке, чем обычное ПО, рассчитанное на применение одного процессора.

Следует отметить, что не всегда вычислительные задачи требуют применения ВПВС. Необходимость возникает, когда размер (размерность) входных данных превышает некоторый порог, за которым время выполнения задачи на одном процессоре (вычислительном блоке) становится неприемлемо большим.

Мы задались следующими вопросами:

1. Следует ли при необходимости применения ВПВС переносить все созданное ранее программное обеспечение (со всей своей функциональностью) с других платформ, например персональных ЭВМ, на высокопроизводительные?

2. Можно ли отдельные функции (работа с большими массивами данных, решение вычислительно трудоемких задач и др.) реализовать в виде информационных сервисов, доступных по требованию?

Полный перенос всех возможностей с одной платформы на другую – очень трудоемкое решение, и уж тем более не по силам одному разработчику. Но в случае положительного ответа на второй вопрос

можно получить гибкую инфраструктуру, в которой ВПВС предоставляют набор сервисов, доступных по требованию.

Набор сервисов может включать в себя:

- 1) решение численными методами задач линейной алгебры;
- 2) обработку цифровых сигналов, в том числе мультимедийных;
- 3) решение задач дискретной математики и теории графов;
- 4) индексацию и поиск по сверхбольшим массивам информации;
- 5) распознавание образов;
- 6) виртуализацию приложений и др.

Такая организация доступа, получившая название сервис-ориентированной, позволяет снизить стоимость разработки ПО для решения задач с применением ВПВС и, следовательно, повысить коэффициент полезного действия таких дорогостоящих систем, а также обеспечить возможность включения ВПВС в состав регулярно действующих информационных систем, таких, например, как поисковые, предметно-ориентированные системы и т.п.

В настоящее время имеется два отчасти пересекающихся направления развития технологий распределенных вычислений, имеющих некоторые необходимые, но недостаточные, на данный момент, свойства для организации сервис-ориентированной архитектуры доступа к ресурсам ВПВС: «Облачные вычисления» (Cloud computing) [1, 2] и GRID [3, 4]. Нами начаты исследования и эксперименты в этих направлениях. В качестве ВПВС, ресурсы которой предполагается предоставить в виде набора информационных сервисов, используется вычислительный кластер нашего университета (<http://cluster.tusur.ru>).

Один из экспериментов, предпринятый нами, заключается в том, чтобы взять определенную предметно-ориентированную систему, обладающую возможностями расширения, и создать программную инфраструктуру, позволяющую осуществлять часть расчетов на вычислительном кластере (рис. 1). В качестве прикладной системы был выбран пакет MATLAB. Для этого пакета нами разработано специальное расширение на языке Java, позволяющее обращаться к веб-сервису для решения простой задачи — перемножение двух матриц. Основная задача расширения, работающего на уровне клиента, — передача запроса на решение задачи и ее параметров на сервер приложений (использовался Apache Tomcat), а затем после выполнения расчетов получение результата и возврат в программу. При этом предусмотрено хранение массивов в виде удаленных объектов, так как локальная система может не умещать все параметры в локальной ОЗУ.

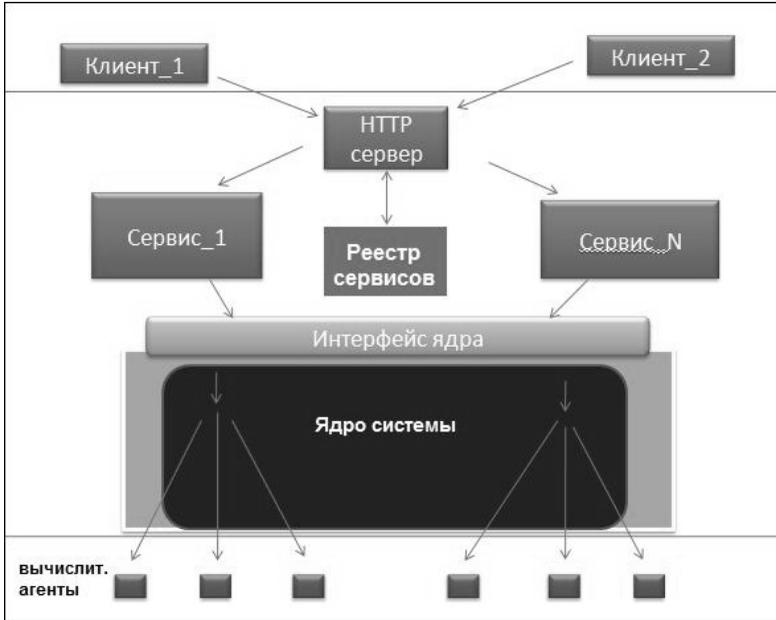


Рис. 1. Сервис-ориентированная архитектура доступа к ресурсам ВППС

Реестр сервисов обеспечивает хранение и выдачу технических параметров для решения конкретного типа задачи. Решение задачи происходит в ядре системы, которое определяет метод решения задачи, разделяет данные в зависимости от выбранного метода, запускает процесс счета на вычислительных агентах, контролирует выполнение и выдает результат на уровень сервиса, чтобы затем результат был возвращен клиенту.

Предполагается, что задача должна быть типовая и метод решения должен быть полностью описан и зарегистрирован в системе. Мы проводили эксперимент с использованием двух методов решения: с использованием прикладной библиотеки Intel Math Kernel Library (MKL), установленной на вычислительном кластере, и с использованием собственной разработки, основанной на технологии MPI. Это позволяет смоделировать две ситуации: первая, когда в систему интегрируется уже готовое программное обеспечение, и вторая, когда система может быть расширена за счет включения пользовательского прикладного ПО. На данный момент разработанное программное обеспечение находится в стадии испытаний.

Кроме чисто вычислительных задач, существуют и другие прикладные задачи, для которых уже разработаны соответствующие сервисы. Например:

- сервис обработки распределенных баз данных с использованием технологии MapReduce. Реализация этого сервиса осуществлена в проекте Hadoop (<http://hadoop.apache.org/>);
- сервис доступа к виртуальным средам. Здесь пользователь может собрать как из конструктора свою прикладную систему, используя блоки: операционная система, библиотеки выполнения программ, прикладные библиотеки, серверное ПО и т.д. Собранный таким образом система загружается на один из имеющихся серверов и предоставляется пользователю как реально действующая. Примером реализации является проект Tashi (<http://incubator.apache.org/tashi/>);
- сервис доступа к физическим средам. В отличие от предыдущего сервиса здесь предполагается, что пользователь может также конфигурировать аппаратное обеспечение. Примером реализации является проект Tusoop.

В будущем мы предполагаем встраивать в нашу сервисориентированную архитектуру и такие виды сервисов.

Литература

1. Cloud computing [Электронный ресурс] / Википедия. Свободная энциклопедия. 2009 Режим доступа: http://en.wikipedia.org/wiki/Cloud_computing, свободный
2. Grid computing [Электронный ресурс] / Википедия. Свободная энциклопедия. 2009 Режим доступа: http://en.wikipedia.org/wiki/Grid_computing, свободный
3. Cloud Computing: высокая облачность [Электронный ресурс] / Авт. Андрей Крупин. Электрон. журнал. М. : Компьютерра, 2009. Режим доступа: <http://www.computerra.ru/interactive/461761/>, свободный
4. Распределённые вычисления, GRID-технологии или кластеры? [Электронный ресурс] / Авт. Леонид Черняк. Электрон. журнал М. : Открытые системы, 2004. Режим доступа: <http://www.osp.ru/cw/2004/04/72923>, свободный