УДК 519.687

Востокин С.В., Дорошин А.В., Артамонов Ю.С.

ПРИМЕНЕНИЕ СИСТЕМЫ TEMPLET WEB ДЛЯ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ СИСТЕМ

Введение

В современных вычислительных задачах с использованием высокопроизводительных систем используют множество технологий обработки и хранения данных, организации параллельных вычислений, представления и анализа результатов.

При разработке приложений возникает ряд характерных проблем:

– растущая сложность алгоритмов и систем;

– сильная зависимость приложений от окружения: нестандартный набор программного обеспечения (ПО), периферийных устройств, сложная конфигурация оборудования;

– необходимость обеспечения заданной производительности даже для тестовых окружений;

– сложный (длительный) процесс развёртывания ПО в целевом окружении;

– необходимость соблюдать регламент работы с целевым окружением, которое часто является общим для многих групп разработчиков и проектов, а также требует оформления доступа.

Под окружением будем понимать некоторый программно-аппаратный комплекс: совокупность физических ЭВМ, каналов связи, периферийных устройств и ПО, требуемого для работы системы. Одним из примеров является кластер серверов: узлы кластера, высокоскоростные сети передачи данных, система пакетной обработки.

Организация совместной работы

Командная работа становится одним из важнейших аспектов разработки, как промышленного ПО, так и в научного [1]. Во многом это обусловлено тем, что команда может разрабатывать многие модули и подсистемы параллельно, а одному разработчику сложно держать под контролем все аспекты работы приложения. Кроме того, команды могут включать как специалистов по предметной области, так и разработчиков инструментария параллельной обработки.

Выделим наиболее важные аспекты эффективной коллективной работы над приложениями.

1. Использование систем контроля версий. В процессе разработки код приложения претерпевает значительные изменения, которые требуется отслеживать и контролировать.

2. Наличие эффективного и простого механизма сборки приложения без использования графических утилит, поддерживающего управление зависимостями и различные целевые архитектуры.

3. Написание модульных, интеграционных и приёмочных тестов. Тестирование сложных вычислений вручную крайне неэффективно, и поэтому требуются регулярные автоматические проверки.

4. Использование тестовых окружений без эксплуатации основных доступных мощностей для проведения тестирования и отладки. Запуск приложения на суперкомпьютере для целей тестирования занимает разделяемые ресурсы, что нерационально. Кроме того, результаты тестирования могут быть получены с существенной задержкой в случае загруженности основных ресурсов.

Сервис запуска задач научных вычислений Templet Web

Для автоматизации командной работы над проектами параллельных приложений нами был разработан веб-сервис Templet (http://templet.ssau.ru/templet), работающий на базе Медиацентра СГАУ.

Сервис позволяет:

- ускорить разработку приложения при помощи шаблонов вычислительных методов пакета Templet SDK [2]: портфель задач, конвейер и другие;
- организовать работу команды над проектом с применением VCS;
- организовывать частное облако окружений для запуска проектов [3];
- получить доступ к общим окружениям развёртывания и запущенным задачам;
- разворачивать приложение в тестовом и целевом окружениях;
- отслеживать работу приложения во время продолжительных вычислений;
- получать уведомления о статусе задач в целевых окружениях;
- управлять бинарными зависимостями шаблонов и проектов для различных платформ и архитектур.

Шаблоны проектов позволяют предоставлять готовые структуры проектов и стандартизировать пакеты поставки приложений в целевые окружения.

Этапы развёртывания проекта включают: получение исходного кода или бинарных артефактов из VCS; получение зависимостей из репозитория для конкретного окружения;

загрузка пакета развёртывания в целевое окружение; сборка проекта в целевом окружении; запуск или добавление в стороннюю пакетную систему.

В системе поддерживаются виртуальные окружения Linux и суперкомпьютер «Сергей Королев».

Сервис ведёт учёт использования ресурсов суперкомпьютера и предоставляет статистику по запущенным задачам и загрузке пользователям. Это помогает оценить максимальный объём ресурсов, который сможет использовать приложения, с учётом допустимого времени ожидания вычислительной задачи в очереди пакетной системы [4].

Решение прикладных задач при помощи программного комплекса Templet

Сервис Templet был опробован при разработке приложения для анализа параметров многомерных динамических систем. Использовался как веб-сервис, так и набор библиотек Templet SDK (шаблон «Портфель задач») для распараллеливания исходной задачи. Шаблон Templet SDK «Портфель задач» является одним из типовых решений для распараллеливания алгоритмов, он может использоваться как для вычислений в общей, так и в распределённой памяти [5]. Он подходит для оптимизационных и переборных задач. Если исходный последовательный алгоритм может быть разбит на независимые по данным задачам, выбор следует остановить на данном решении.

Задача анализа многомерных динамических систем с помощью сечений Пуанкаре содержит независимые по данным этапы вычисления координат пересечения траекторий с плоскостью сечения [6]. Распараллеливание вычисления одной траектории является крайне сложной задачей, так как не зависимыми по данным в каждый момент времени являются не более нескольких десятков машинных инструкций. В связи с высокими затратами на синхронизацию в случае распараллеливания вычислений одной траектории производилось распараллеливание по блокам вычисления траекторий.

Распараллеливание приложения для шаблона «Портфель задач» состоит в разработке функций извлечения задачи из портфеля, её обработки и помещения результатов обработки в портфель. Причём последовательность выполнения при параллельном выполнении сохраняется только для функции извлечения задач. Эксклюзивным доступом к портфелю обладают функции извлечения задачи и помещения результатов в портфель.

Типичный вариант расчётов включает в себя вычисления сечения 100-200 траекторий, что накладывает ограничения на масштабируемость алгоритма. Некоторого ускорения вычислений можно добиться, увеличивая число потоков до числа траекторий, но для эффективного использования вычислительных ресурсов необходимо, чтобы число

траекторий было в 2-10 раз больше числа потоков вычислений. Достигнутая длительность вычисления 160 траекторий на 20 узлах кластера в 120 раз меньше, чем у последовательном варианта, при этом использовались 160 потоков вычислений.

Таким образом, исходная задача была декомпозирована и представлена в виде системного и прикладного уровней. Шаблон вычислительного метода реализуется системными программистами и может быть повторно использован во многих проектах. Прикладные программисты разрабатывают приложение для вычислений в терминах и идиомах шаблона вычислительного метода, что избавляет их от необходимости ручного распараллеливания кода.

Templet SDK, включающий набор шаблонов вычислительных методов, упростил создание параллельного приложения. Инструментарий позволяет запускать приложение в нескольких режимах: отладка, эмуляция, приложение Win32, POSIX приложение, распределённое MPI приложение. В Templet SDK также включён эмулятор, при помощи которого можно оценить максимальное ускорение (оптимистичную оценку) для большого числа процессоров без реального запуска на суперкомпьютере.

Заключение

Программный комплекс Templet позволяет использовать различные особенности целевых окружений, предоставляя абстракцию для запуска приложений и упрощая повторный запуск вычислительных задач. Он поддерживает собственные библиотеки шаблонов параллельных приложений, средства для разработки и отладки прикладных решений, адаптированные для использования особенностей поддерживаемых окружений. Сервис реализует принцип разделения системной и прикладной работы над приложениями. Такой подход позволяет прикладным программистам сосредоточить усилия на разработке вычислительных алгоритмов, снизить трудоёмкость разработки и добиться экономии высокопроизводительных ресурсов на стадии тестирования и отладки.

Библиографический список

1. Dongarra J.J. Multiphysics simulations: Challenges and opportunities [Текст] / J. Dongarra // International Journal of High Performance Computing Applications, 2013. №2. – University of Tennessee, Knoxville, USA. С. 50 - 54.

2. Востокин С.В. Препроцессор языка Templet: инструмент программирования в терминах модели «процесс-сообщение» / С.В. Востокин // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки. – 2014. – № 3(36) – С. 169-182.

3. Vostokin S.V. TEMPLET – a Cloud Service for Rapid Development of High Performance Applications [Текст] / S.V. Vostokin, Y.S. Artamonov, Y.P. Nazarov,

A.E. Zagumennikov // Distributed Computing and Grid-Technologies in Science and Education: Book of Abstr. of the 5th Intern. Conf. (Dubna, July 16-21, 2012). – 2012. C.166-167.

4. Артамонов Ю.С. Постановка задачи прогнозирования доступных вычислительных ресурсов в кластерных системах [Текст] / Ю.С. Артамонов // Международная научно-техническая конференция «Перспективные информационные технологии (ПИТ 2013)». Самара. Издательство Самарского научного центра РАН. – 2013. С. 178-180.

5. Литвинов В.Г. Разработка и применение вычислительной модели типовых решений. Пример использования «портфеля задач» для обучения нейронной сети HRBF [Текст] / В.Г. Литвинов // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки. – 2014. – №3(36). – С. 183-195.

6. Doroshin A.V. Heteroclinic dynamics and attitude motion chaotization of coaxial bodies and dual-spin spacecraft [Текст] / A.V. Doroshin // Communications in nonlinear science and numerical simulation. – 2012. – №3. – С. 1460-1474.