

Востокин С.В., Казакова И.В., Попов С.Н.

МЕТОД ПОСТРОЕНИЯ КОМПОЗИТНЫХ ПРИЛОЖЕНИЙ ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

В настоящее время на всех этапах исследования динамических систем предполагается применение компьютерных средств моделирования. Этап анализа и разработки модели в форме, пригодной для интегрирования, выполняется с использованием систем компьютерной алгебры (Maple, Mathematica, SymPy). Построение траекторий в фазовом пространстве может выполняться с помощью пакетов численного интегрирования (Boost.Numeric.Odeint). Статистический анализ наблюдаемого поведения систем целесообразно проводить при помощи специальных фреймворков анализа данных (Apache Spark). Кроме этого, для наглядного представления результатов моделирования применяются пакеты визуализации (D3.js).

Существуют интегрированные приложения и сервисы (Mathcad, SMath Studio), которые могут использоваться для комплексной автоматизации всех этапов исследования. Однако при их использовании приходится принимать компромиссное решение, выбирая между функциональностью, аппаратными требованиями и стоимостью. Также существенным фактором при выборе компьютерных средств моделирования динамических систем является возможность организации высокопроизводительных вычислений.

Перспективным методом организации вычислительных экспериментов на основе комбинации необходимых пакетов и программ, который также позволяет задействовать необходимые вычислительные ресурсы, являются композитные приложения. Основное преимущество композитных приложений перед интегрированными приложениями состоит в открытости и гибкости их структуры. Это позволяет эффективно управлять процессом исполнения компонентов приложения на различных ресурсах распределенной среды.

Целью данного исследования была разработка приложения, проверяющего на тестовой задаче возможности технологии композитных приложений для моделирования динамических систем. Специфические требования, которые учитывались при разработке тестового приложения, включают: 1) использование простаивающих рабочих станций для моделирования и статистической обработки; 2) исполнение на разных операционных системах; 3) простоту и оперативность развертывания компонентов композитного

приложения; 4) организацию длительных вычислений, устойчивых к отказу; 5) организацию вычислений с большим количеством задач и сложными зависимостями между ними.

Функционирование разработанного в ходе исследования композитного приложения основано на взаимодействии двух облачных сервисов. Веб-сервис Templet Web [1] применялся для разработки и развертывания кода программы-оркестратора композитного приложения. Данная программа осуществляет централизованное управление процессом вычислений, формируя поток независимых задач в исполняющие их подсистемы. Код оркестратора реализован на языке C++ на базе фреймворка Templet [2], интегрированного в веб-сервис. Для программирования оркестратора используется акторо-подобная модель вычислений [3], которая позволяет оркестратору поддерживать модель всех компонентов приложения и отслеживать их взаимодействие.

Запуск композитного приложения осуществляется через веб-интерфейс сервиса Templet Web, при этом компонент оркестратора развертывается на выделенной виртуальной машине на кластере Суперкомпьютерного центра Самарского университета. В процессе работы оркестратор обращается к сервисам системы Everest [4]. Everest – облачный сервис, в который через веб-интерфейс предварительно устанавливаются компоненты композитного приложения. Сервис Everest переадресует запросы на выполнение задач программным агентам, развернутым на простаивающих компьютерах внутри корпоративной сети Самарского университета. Агенты, в свою очередь, запускают компоненты композитного приложения для обработки переданных оркестратором задач.

Было разработано композитное приложение блочной сортировки большого массива экспериментальных данных, в котором были учтены заявленные требования. Протестировано использование ноутбуков, рабочих станций, виртуальных машин, имитирующих варианты доступных простаивающих вычислительных ресурсов в корпоративной среде Самарского университета. Разработанные компоненты композитного приложения исполнялись под управлением операционной системы Linux (оркестратор), а также под управлением ОС Windows (сортировщики и мерджеры). Простота развертывания обеспечивалась за счет использования веб-интерфейса, исполняемого в стандартных браузерах. Программы-агенты также используют несложную процедуру установки и зависят только от среды Python (её требуется установить в случае отсутствия на простаивающем ресурсе). Отказоустойчивость вычислений задач обеспечивается внутренними механизмами сервиса Everest. Оркестратор выполняется в виртуальной среде, отказоустойчивость которой поддерживается системой VMware.

Сложные зависимости между запускаемыми задачами организуются с использованием акторо-подобной модели фреймворка Templet.

Композитные приложения – достаточно общий принцип разработки, используемый как в исследовательских проектах, так и в продуктах крупных компьютерных корпораций IBM, SAP и др. Наиболее близкой по функциональности системой-аналогом для математического моделирования, применяющей данную технологию, является система CLAVIRE [5]. Потенциальным преимуществом принятого метода разработки композитных приложений является возможность реализации более сложных зависимостей между задачами по сравнению с зависимостями, описываемыми направленными ориентированными графами (DAG) системы CLAVIRE. Также проще выполняется процедура развертывания компонентов приложения, что позволяет использовать простаивающие вычислительные ресурсы корпоративной среды для организации моделирования.

Проведенное исследование подтвердило перспективность метода композитных приложений для решения задач моделирования динамических систем, а также состоятельность технического подхода, выбранного для его реализации. В дальнейшем планируется расширить код фреймворка Templet для прозрачного взаимодействия между оркестратором и сервисом Everest, что позволит реализовывать код оркестратора непосредственно исследователем, без участия системного программиста.

Работа выполнена в рамках Государственного задания №9.1616.2017/ПЧ по теме «Разработка фундаментальных основ аналитического синтеза регулярных и хаотических процессов в динамике космических аппаратов».

Библиографический список

1. Vostokin S.V. Templet Web: the use of volunteer computing approach in PaaS-style cloud [Text] / S.V. Vostokin, Y.S. Artamonov, D.A. Tsaryov // Open Engineering. – 2018. – Vol. 8(1). – P. 50-56.
2. Vostokin S. V. The templet parallel computing system: Specification, implementation, applications [Text] / S. V. Vostokin // Procedia Engineering. – 2017. – Vol. 201. – P. 684-689.
3. Vostokin S. V. Templet: A markup language for concurrent actor-oriented programming [Text] / S. V. Vostokin // CEUR Workshop Proceedings. – 2016. – Vol. 1638. – P. 460-468.
4. Sukhoroslov O. A Web-Based Platform for Publication and Distributed Execution of Computing Applications [Text] / Sukhoroslov O., Volkov S., Afanasiev A. // IEEE Xplore. – 2015. – Vol. 14. – P. 175-184.
5. Knyazkov K. CLAVIRE: e-Science infrastructure for data-driven Computing [Text] / Kovalchuk S., Tchurov T., Maryin S., Boukhanovsky A., // Journal of Computational Science. – 2012. – Vol. 3, N 6. – P. 504–510.