



Гвоздева С.Н., Панищев В.С., Вагутин Э.И // Интеллектуальные и информационные системы. – Тула: изд-во ТулГУ, 2019. – с. 59–63.

Цзе Дон, С.В. Востокин

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ EVEREST И MYBINDER

(Самарский университет)

На этапах исследования динамических систем применяются различные программные средства компьютерного моделирования. Этап анализа и разработки модели в форме, пригодной для интегрирования, реализуется в системах компьютерной алгебры (Maple, Mathematica, SymPy). Этап построения траекторий в фазовом пространстве выполняется с помощью пакетов численного интегрирования (Boost.Numeric.Odeint). Этап статистического анализа наблюдаемого поведения систем проводится в фреймворках анализа данных (Apache Spark). Этап представления результатов моделирования в удобном для анализа виде осуществляется при помощи пакетов визуализации (D3.js) [1].

Очевидно, что для снижения трудоемкости рабочего процесса моделирования динамических систем необходима комплексная автоматизация всех этапов исследования. Традиционно для этого использовались монолитные коммерческие системы и сервисы (Mathcad, SMath Studio). Однако в связи с интенсивным развитием web-технологий и технологий облачных вычислений стали активно развиваться новые подходы автоматизации рабочего процесса (workflow) численного моделирования (Project Jupyter, Everest Platform).

Ключевыми парадигмами проекта Jupyter являются клиент-серверная модель и графическая интерактивная среда в стиле REPL (read-eval-print loop), реализованные в форме web-интерфейса. Многооконный интерфейс JupyterLab позволяет работать сразу с несколькими языками программирования в дополнении к языку Python [2]. Публичные облачные сервисы (mybinder.org) автоматизируют развертывание экземпляров JupyterLab на бесплатных виртуальных машинах [3].

Облачная платформа в виде сервиса (PaaS) Everest [4] предназначена для композиции приложений, работающих на различных вычислительных ресурсах, с целью реализации единого рабочего процесса численного моделирования. В качестве аппаратных ресурсов приложений могут выступать физические рабочие станции, виртуальные машины, кластерные системы. При этом для ресурса не требуется выделенного IP-адреса и постоянного подключения. Ресурс может работать в конфигурации «за брэндмауэром». Платформа Everest является публично доступной.

Изложенное выше показывает, что существует техническая возможность совместного использования облачной платформы Everest и платформ проекта



Jupyter, частности MyBinder, для комплексной автоматизации вычислительного эксперимента. Нашей целью являлась разработка технологии проведения вычислительных экспериментов для исследования динамических систем на базе современных программно-аппаратных решений Everest и MyBinder.

В проведенном исследовании решались следующие задачи:

1. анализ принципов моделирования динамических систем с целью выявления существенных для автоматизации особенностей;
2. обзор технологий автоматизации вычислительного эксперимента на высокопроизводительной технике;
3. разработка алгоритмов управления распределенными вычислениями на базе парадигмы «портфель задач»;
4. проектирование архитектуры распределенного приложения;
5. реализация тестового распределенного приложения, автоматизирующего анализ динамических систем, для экспериментального подтверждения его работоспособности.

Обзор предметной области показал, что численное моделирование динамических систем основано на получении множества частных решений систем дифференциальных уравнений при разных начальных условиях. Данный процесс отличается значительной вычислительной сложностью, но в тоже время может быть легко распараллелен. В качестве технологий параллельного программирования моделей динамических систем могут использоваться традиционные технологии OpenMP и MPI. Однако модели обычно реализуются не только на языках C/C++ и Fortran, для которых разработаны стандарты OpenMP и MPI. Запрограммированные модели могут исполняться как на отдельных высокопроизводительных системах, так и на их комбинации. Поэтому для наших целей применение только OpenMP и MPI не достаточно. Решением является применение технологий, разработанных для программирования грид-систем из настольных компьютеров [5], поддерживаемых в том числе платформой Everest.

При проектировании приложений для исследования динамических систем в качестве основы интеграции его компонентов предлагается использовать платформу Everest по следующим причинам. Не требуется развертывание управляющего компонента, так как он реализован в форме публичного PaaS сервиса (everest.distcomp.org). Взаимодействие между компонентами приложения реализуется по REST API, что гарантирует интероперабельность. Настройка сервиса под решаемую задачу осуществляется через удобный графический интерфейс веб-обозревателя. Сервис поддерживает запуск унаследованного программного обеспечения (legacy software) OpenMP и MPI на различные ресурсы, включая кластерные системы. Последнее свойство является важным, так как позволяет использовать готовые пакеты численных методов, например, для численного интегрирования моделей.

В качестве системы программирования и реализации интерфейса приложений в нашем проекте выбрана платформа JupyterLab по следующим причинам: обеспечивается простота развертывания на основе публичных



binderhub-сервисов, например, mybinder.org; интерфейс JupyterLab изначально ориентирован на задачи математического моделирования; типовой рабочий процесс моделирования реализуется в REPL интерфейсе jupyter-ноутбуков. Интерфейс ноутбуков JupyterLab совместим с различными языками, включая языки C/C++, используемые для программирования моделей динамических систем.

Для сканирования параметрического пространства модели динамических систем нами выбран паттерн «портфель задач» (bag-of-tasks) [6]. Данный паттерн, наряду с менее общим паттерном «применить ко всем» (map), является традиционным для программирования грид-систем из настольных компьютеров.

Распределенное приложение для исследования динамических систем включает следующие компоненты. Интерфейсная часть реализована в виде git-репозитория в формате binderhub. Репозиторий содержит программу управления вычислениями по паттерну «портфель задач» на языке C++; REST API (на языке C++) для связи с сервисом Everest и запуска задач интегрирования уравнений дискретных систем с заданными начальными условиями; сценарий рабочего процесса моделирования, реализованный в виде jupyter-ноутбука; сценарий развертывания на виртуальную машину под управлением сервиса mybinder.org.

Вычислительная часть реализована в виде Everest-приложения, которое представляет собой программу на языке C++, решающую задачу интегрирования уравнения дискретной системы для заданных начальных условий, и специальный конфигурационный файл формата JSON. Файл JSON формируется при установке приложения на платформу Everest через графический интерфейс (everest.distcomp.org). Для развертывания вычислительной части приложения на ресурсе (кластере, рабочей станции или виртуальной машине) устанавливается агент ресурса – специальная программа на языке Python, взаимодействующая с сервером платформы Everest. Агент ресурса регистрируется на сервере Everest, для этого выполняется аутентификация с использованием токена.

В вычислительном эксперименте исследовались: настройка приложения; возможность устойчивой работы в распределенной конфигурации в гибридном облачном окружении [7]; длительность выполнения отдельной задачи интегрирования уравнения дискретной системы для получения значимого ускорения.

Вычислительный эксперимент выполнялся на эмуляторе распределенной системы, который включает виртуальную машину с интерфейсной частью приложения; виртуальную машину с агентом вычислительной части приложения; сервер Everest. В запускаемых задачах для контроля работоспособности выполнялась фиктивная операция (возведение переданного из интерфейсной части числа в квадрат) и заданная временная задержка. Агент настраивался таким образом, чтобы одновременно выполнялись несколько задач. Тем самым имитировалась параллельная обработка. Для заданного



количества параллельных задач время выполнения отдельной задачи увеличивалось до тех пор, пока наблюдался прирост ускорения. Найденное таким образом время является рекомендованным временем вычисления отдельной задачи в реальных расчетах. Это время в реальных расчетах подбирается путем объединения нескольких задач в одну (операция *parceling-a*). В экспериментах показано, что для достижения значимого ускорения (на эмуляторе достигавшего 51 раз при количестве одновременно выполнявшихся задач 100 единиц) в предложенной архитектуре необходимо объединять вычисления так, чтобы длительность обработки одной задачи не превышала 20 секунд.

Таким образом, нами разработана и апробирована технология проведения вычислительных экспериментов для исследования динамических систем на базе современных программно-аппаратных решений *Everest* и *MyBinder*. Данная технология может быть рекомендована для других предметных областей численного моделирования, связанных с проведением большого числа ресурсоемких и независимых вычислительных экспериментов.

Литература

1. Востокин С.В., Бобылева И.В., Попов С.Н. Метод построения композитных приложений для моделирования динамических систем // XXI Всероссийский семинар по управлению движением и навигации летательных аппаратов. — 2019. — Ч. II. — С. 83-85
2. Granger B., Colbert C., Rose I. JupyterLab: The next generation Jupyter frontend // JupyterCon 2017. — 2017.
3. Ragan-Kelley B., Willing C. Binder 2.0-Reproducible, interactive, sharable environments for science at scale. — 2018.
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. Ivashko E., Chernov I., Nikitina N. A survey of desktop grid scheduling // IEEE Transactions on Parallel and Distributed Systems. — 2018. — Т. 29. — №. 12. — С. 2882-2895.
6. Senger H., da Silva F. A. B. Bounds on the scalability of bag-of-tasks applications running on master-slave platforms // Parallel Processing Letters. — 2012. — Т. 22. — №. 02. — С. 1250004.
7. Goyal S. Public vs private vs hybrid vs community-cloud computing: a critical review // International Journal of Computer Network and Information Security. — 2014. — Т. 6. — №. 3. — С. 20.