

МОДЕЛИРОВАНИЕ МНОГОПОТОЧНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С ЦЕЛЬЮ АНАЛИЗА И ОПТИМИЗАЦИИ ИХ ПРОИЗВОДИТЕЛЬНОСТИ

Ю.Ю. Белых

Моделирование и предсказывание производительности компьютерных программ способствует их эффективному дизайну, установке, современному анализу и устранению проблем. Однако, моделирование производительности многопоточных систем сильно затруднено в виду использования в таких программах методов блокировок и конкурентного доступа к вычислительным ресурсам. Существующие модели производительности требуют либо большого объема начальных данных для их эффективного использования, либо накладывают определенные ограничения на типы моделируемых систем. Данная статья описывает новый подход в моделировании систем, позволяющий преодолеть данные ограничения.

Введение

Производительность – одна из важнейших характеристик программных систем, которая зависит от множества факторов, таких как архитектура программы, аппаратные свойства платформы запуска и характер системных нагрузок. Производительность так же, сильно зависит от выбора конфигурационных опций программы, примерами таких опций могут быть размеры встроенного кэша операции ввода/вывода либо количество потоков выполнения.

Правильное понимание процесса выбора системной конфигурации является определяющим для производительности большинства приложений. Как правило, этот процесс подразумевает построенные модели производительности рассматриваемой системы. Данная модель, должна позволять предсказывать характеристик производительности системы в различных конфигурациях, таких как нагрузки, опции настройки и аппаратные параметры.

Модели производительности могут применяться в различных ситуациях. Во время непосредственной разработки, такие модели могут предсказывать характеристики производительности платформы и выявлять потенциальные проблемы архитектуры на ранних стадиях. После поставки системы, такие модели позволяют оптимизировать производительность путем подбора подходящей к данной аппаратной платформе конфигурации разработанного программного продукта.

Модели анализа производительности являются основным компонентом в разработке адаптивных, самоконфигурируемых и автономных систем. В таких случаях данные полученные с помощью моделей таких используются для динамической конфигурации с целью повышения производительности систем [1].

Наконец, рассматриваемые модели являются крайне полезным инструментом для анализа и устранения проблем, связанных с производительных программных продуктов. Значительных расхождений в прогнозируемых и реальных данных производительности системы являются явным признаком проблемы.

Традиционные методы моделирование показателей производительности информационных систем

Построение моделей многопоточных программ крайне трудоёмкий процесс. Как правило связанный с моделированием сложны систем блокировок, имитации использования аппаратных ресурсов, таких как процессор и системы ввода/вывода, и других атрибутов реальных программных продуктов. Как результат существующие модели либо накладывают определенные ограничения на типы систем с которыми они могут работать, либо требуют предварительного сбора значительных объёмов реальных данных о производительности системы в различных конфигурациях. Такие ограничения часто приводят к неприменимости данных моделей в реальных проектах. Преодоление данных ограничений является одной из основных целей данной работы.

Разработанная техника позволяет строить модели производительности с меньшим количеством исходных данных и для широкого спектра систем. Полученные модели могут имитировать работу многопоточных приложений, запущенных в различных конфигурациях и с различными входным нагрузками. Также разработанные модели можно применять для анализа отдельных компонент системы, получения индивидуальных характеристик производительности и потребления ресурсов, что в свою очередь позволяет выявлять узкие места в производительности системы в целом.

Моделирование информационной системы как процесса обработки заявок

В данной работе были задействованы комбинации методов статического и динамического анализа данных для сбора информации о моделируемых системах. Сначала программа анализируется, моделируются ее ключевые части, затем выполняется симуляция работы в выбранной конфигурации.

Для непосредственного моделирования поведения программы используется дискретно-вероятностный аппарат математического моделирования. Программа представляется как поток заявок, где очереди соответствуют очередям и буферам реальной системы, а обработчики заявок соответствуют программным потокам. Каждый поток представлен в виде вероятностного графа вызовов, а каждый переход – вызовов процедуры программного кода.

Для удобства моделирования, программы разделяются на составные части, где каждая часть представляет собой элементарную операцию, такую как задействование процессорных мощностей, операции ввода/вывода,

синхронизации, буферизации и т.д. Каждой такой части в соответствие ставится компонент модели. Такой подход сильно отличается, от более распространённого, но менее общего (неприменимого для широкого набора системы) подхода, где программа рассматривается как комбинация высокоуровневых объектов специфичных для рассматриваемой системы, например, компонента обращения к системе передачи сообщений (MPI). В результате, полученные модели позволяют имитировать поведение практически любой многопоточно программы. Для дальнейшего увлечения гибкости модели и наращивания спектра применения, компоненты, отвечающие за симуляцию операционной системы и аппаратных блоков реализованы совершенно независимыми от непосредственной модели программы.

В процессе моделирования вычислительные задачи, выполняемые рассматриваемыми программными системами, представляются как процесс обработки заявок. Заявка определяется, как что-то, на что программа реагирует определённым образом. Программа обрабатывает заявку выполняя некоторый набор операций. Производительность процесса обработки заявки может быть описана различными метриками, такими как: время ответа (общее время задержки между прибытием заявки и ее полной обработкой), пропускная способность (количество обрабатываемых заявок в единицу времени) и количество отказов в обработке. Данный подход позволяет естественным образом моделировать событийные информационные системы, представляющие большинство современных программных продуктов.

Например, в работе веб-сервера, заявка представляет входящий HTTP запрос. Обработка заявки соответствует процессам загрузки веб-страницы с диска и отправки ее пользователю. В модулирование систем с графическим интерфейсом, таких как, например, текстовый редактор, заявка соответствует вводу символа, либо перемещению мыши. Обработкой заявки, в данном случае является обновление графического интерфейса в ответ на действие пользователя и работа с данными программы.

Разработанная модель позволяет предсказывать производительность системы в различных конфигурациях. Конфигурации описывают различия в нагрузке, программных параметрах, таких как количество потоков, и аппаратных характеристик изучаемых систем.

Полученная модель рассматривается как иерархичная дискретно-вероятностная сложная система:

- первый уровень отвечает за имитацию потоков данных используя вероятностные цепи;
- второй уровень имитирует программные потоки с помощью случайных графов;
- третий уровень моделирует аппаратное поведение.

Модели строятся на основе данных полученных с помощью методов статического и динамического анализ изучаемых систем. Сюда входит информация о структуре программы, семантика взаимодействия программных потоков, требования к ресурсам индивидуальных компонент системы и т.д. Полученная информация переводится в дискретно-вероятностную модель для дальнейшего изучения.

Для проведения описанного вида моделирования разработан специальный программный пакет, позволяющий анализировать и предсказывать поведение многопоточных программ, написанных на различных языках и с использованием различных библиотек для многопоточной разработки.

Для проверки описанного подхода в моделировании многопоточных системы были построены модели простых вычислительных приложения и веб-сервера, работающего на операционной системе Linux.

Выводы

Данная работа в отличии от существующих решений в области моделирования производительности программных систем имеет ряд особенностей:

- данный подход не ограничен моделированием конкретных типов информационных систем и может быть использован для имитации широко спектра программных продуктов;
- в подходе применяется достаточно простой, однако выразительный метод имитации работы с системами ввода/вывода, процессором, памятью и другими внешними ресурсами;
- особое внимание уделяется симуляции такого аспекта как многопоточность.

Полученные модели позволяют успешно имитировать поведение больших многопоточных программных продуктов предсказывая их поведение и характеристики в различных конфигурациях с достаточной степенью точности.

Список литературы

1. Циллюрик, О. QNX/UNIX: анатомия параллелизма / О. Циллюрик, Е. Горошко. – Санкт-Петербург: Символ-Плюс, 2005. – 288 с.

Белых Юрий Юрьевич, аспирант кафедры фундаментальной и прикладной математики, Гродненского государственного университета имени Янки Купалы, notanone@yandex.ru