РАЗДЕЛ II. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Ермаков А.В., Сучкова Л.И.

Исследование производительности алгоритмов обработки измерительной информации в системе принятия решений автономного робота

ФГБОУ ВО «Алтайский государственный технический университет им. И.И. Ползунова»
(Россия, Барнаул)

doi:10.18411/spc-22-06-2020-04

idsp: sciencepublic-22-06-2020-04

Аннотация

В системе принятия решений автономного робота важно найти быстро подходящее решение. Поскольку время реакции ограничено, эти алгоритмы должны быстро выполняться, выдавать лучшее решение и масштабироваться при увеличении вычислительных ресурсов. В этой статье показано, как алгоритмы обработки измерительной информации могут быть заданы сценарным интерпретируемым языком. Создано несколько возможных реализаций системы принятия решений и предоставлены экспериментальные результаты, демонстрирующие, что интерпретируемые языки могут сочетать удобство и производительность.

1. Введение

Потребность в сборе и анализе огромного количества информации возросла в настоящее время. Может потребоваться выполнить множество вычислительных задач, и поиск оптимального решения зачастую занимает много времени, что неприемлемо для условий реального времени. В частности, система принятия решений автономным роботом должна обеспечивать обработку значительного объёма входной измерительной информации при существующем лимите времени.

В этой работе мы обсудим аспекты использования сценарных интерпретируемых языков в системе принятия решений автономным роботом, сфокусировавшись на производительности. Мы создаём реализацию системы и показываем, что, несмотря на явный параллелизм – когда каждый алгоритм выполняется в потоке параллельно – производительность увеличилась незначительно. Мы показываем, что более эффективным может оказаться простая, но оптимизированная реализация. И, наконец, мы покажем, что такая реализация системы может легко масштабироваться при добавлении вычислительных ресурсов.

В начале статьи мы обсудим применимость интерпретируемых языков для описания алгоритмов в системе принятия решения и рассмотрим общие подходы параллелизма (раздел 2). В разделе 3 мы опишем, как реализовать транслятор из языка описаний алгоритмов в три выбранных реализации интерпретируемого языка: python2, lua и luaJIT. И, наконец, мы сравниваем производительность и масштабируемость всех трёх реализаций (раздел 4).

2. Анализ существующих решений

В этом разделе рассматриваются наиболее распространенные модели выполнения и взаимодействия существующих параллельных систем обработки информации.

Авторами [1] рассмотрены два основных подхода к параллелизму: синхронный и асинхронный. Синхронный подход предполагает, что вычисляемые задачи выполняются за примерно одинаковое время, что упрощает синхронизацию и позволяет обойтись более простым планировщиком задач. Асинхронный подход предполагает, что компоненты системы, работающие параллельно, выполняются за разное время, и планировщик задач сможет уменьшить время простоя вычислительных

ядер процессора и увеличить среднее время обработки информации в параллельной системе [2].

Существующие конкурентные языки различаются во многих аспектах, однако все они предоставляют параллелизм на уровне потоков. Чтобы код, написанный на таком языке, мог выполняться в более чем одном потоке, языки предоставляют общие примитивы и зачастую глобальный механизм для синхронизации процессов.

Основной проблемой при разработке конкурентных программ является управление конкурентностью: обеспечение правильной последовательности взаимодействий и коммуникаций между различными потоками и координации доступа к общим ресурсам [3]. Существуют два основных подхода к таким взаимодействиям: взаимодействие через разделяемую память и взаимодействие с помощью передачи сообщений. Подходы имеют разную производительность, но оба вносят задержки.

3. Структура модели принятия решений

Низкая производительность системы принятия решений автономного робота, описанная в работах [4-5], требует разработки новых алгоритмических решений, обладающих низкой временной сложностью. Для тестирования алгоритмов обработки информации целесообразно разработать инструментальную систему, позволяющую описывать алгоритмы на модельном языке программирования и исследовать скорость их работы при учёте возможности параллельной обработки данных.

Легко предположить, что параллелизм позволит с лёгкостью увеличить пропускную способность системы. Однако, за счёт увеличившихся задержек на синхронизацию и межпроцессорное взаимодействие.

Известен подход к проектированию систем обработки измерительной информации, где алгоритмы обработки данных описываются с применением интерпретируемого сценарного языка при условии формализации структуры входных и выходных данных [5]. Поскольку в исследуемых алгоритмах отсутствует зависимость по входным данным, то они могут выполняться параллельно и независимо друг от друга, что в том числе исключает «гонку данных». Система обработки данных включает в себя: воркеры, одновременно работающие процессы, выполняющие интерпретацию описанных алгоритмов и обработку данных параллельно; входную очередь задач и выходную очередь с результатами работы (см. рисунок 1).

Созданный сценарный язык описания алгоритмов транслируется в сценарный язык программирования общего назначения, после выбранные алгоритмы помещаются во входную очередь, где они параллельно выполняются воркерами и обрабатывают измерительную информацию. Трансляция языков является одним из общепринятых подходов [7].

Транслятор языка описания алгоритмов способен генерировать исходные коды программ на следующих языках: python2, lua.

Рисунок 1 – Потоки данных в параллельной системе

4. Экспериментальные результаты

Теоретически прирост скорости при распараллеливании пропорционален числу ядер процессора. Для первого теста была создана параллельная система обработки измерительной информации со структурой, приведённой в разделе 3. Описанные алгоритмы задавались языком описаний и транслировались в python2, который выполнялся одновременно в нескольких процессах.

Результаты тестирования разработанной системы показали увеличение производительности в 2,4 раза на четырёхъядерном процессоре (см. рисунок 2).

Рисунок 2 – Увеличение производительности обработки в зависимости от числа вычислительных ядер

Для второго теста, число воркеров было уменьшено до одного, чтобы минимизировать разброс измерений. Исходный код описанных алгоритмов транслировался в языки python2 и lua, причём второй язык интерпретировался двумя разными реализациями интерпретаторов. Итого для тестирования было предложено три варианта: трансляция в python2 и стандартный интерпретатор CPython; трансляция в lua и стандартный интерпретатор; трансляция в lua и интерпретатор luaJIT, которая выполняется виртуальной машиной [8].

Проведённые исследования показали относительный прирост скорости обработки информации в 137 раз при использовании luaJIT (рисунок 3) по сравнению с первоначальной реализацией.

Рисунок 3 – Сравнение времени обработки информации различными реализациями системы принятия решений на одном вычислительном ядре

5. Заключение

Была спроектирована высокопроизводительная система обработки измерительной информации, позволяющая удобно описывать алгоритмы обработки на

сценарном языке программирования. Применением такой системы может стать быстрое прототипирование и проверка работы алгоритмов.

1. D. Krzywicki, Ł. Faber, A. Byrski, M. Kisiel-Dorohinicki, Computing agents for decision support systems, *Future Gener. Comput. Syst.* 37 (2014) 390–400.
2. Villela, D. Minimizing the Average Completion Time for Concurrent Grid Applications. *Journal of Grid Computing* March 2010, Volume 8, Issue 1, pp 47–59 <https://doi.org/10.1007/s10723-009-9119-2>
3. Ben-Ari, Mordechai (2006). *Principles of Concurrent and Distributed Programming* (2nd ed.). Addison-Wesley. ISBN 978-0-321-31283-9.
4. Ермаков А.В. Разработка модели принятия решений с применением формализма нечетких паттернов для коллектива автономных роботов / А.В. Ермаков, Л.И. Сучкова // Южно-Сибирский научный вестник. – 2018. – № 4. – С. 191-195.
5. Ермаков А.В. Проектирование операционной среды обработки паттернов автономным интеллектуальным роботом [Текст] / Ермаков А.В., Сучкова Л.И. // Ползуновский альманах №4 «Виртуальные и интеллектуальные системы». — Барнаул: Издательство АлтГТУ. — 2018. — С. 69-72.
6. Kabáč M., Consel C., Volanschi N. Designing parallel data processing for enabling large-scale sensor applications // *Personal and Ubiquitous Computing*. – 2017. – Т. 21. – №. 3. – С. 457-473.
7. Alpuente, M., del Mar Gallardo, M., Pimentel, E., Villanueva, A. (2008). An abstract analysis framework for synchronous concurrent languages based on source-to-source transformation. *ENTCS*, 206, 3–21.
8. Carl Friedrich Bolz, Laurence Tratt, The impact of meta-tracing on VM design and implementation. *Science of Computer Programming*, Volume 98, Part 3, 2015, Pages 408-421, ISSN 0167-6423, <https://doi.org/10.1016/j.scico.2013.02.001>.

Шестакова А.Ю.

Применение архитектурного метода при проектировании информационной модели умного города

*ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики»
(Россия, Пермь)*

doi:10.18411/spc-22-06-2020-05

idsp: sciencepublic-22-06-2020-05

Аннотация

На современном этапе развития научного знания актуальными являются разработки в рамках концепции «умного города». В рамках статьи раскрыта сущность данной концепции как многоуровневой системы, состоящей из множества элементов. Обоснована целесообразность применения архитектурного метода для моделирования подобной системы. Разработана архитектурная модель, на основе которой проведен анализ основных бизнес-процессов «умного города».

Ключевые слова: умный город, архитектурный метод описания информационной системы, инструмент Archi, бизнес-процессы, бизнес-сервисы, цифровой двойник.

Abstract

At the present stage of the development of scientific knowledge, developments in the framework of the concept of a “smart city” are relevant. The article reveals the essence of this concept as a multi-level system consisting of many elements. The expediency of applying the architectural method for modeling such a system is substantiated. An architectural model was developed, on the basis of which an analysis of the main business processes of the “smart city” was carried out.

Keywords: smart city, architectural method for describing an information system, Archi tool, business processes, business services, digital double.