

СИСТЕМА АВТОМАТИЧЕСКОГО МАСШТАБИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ ДЛЯ РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

*НИИ многопроцессорных вычислительных систем имени академика
А.В. Каляева Южного федерального университета, г. Таганрог
E-mail: zakhar@mvs.tsure.ru*

В статье рассматривается система автоматического масштабирования параллельных программ для реконфигурируемых вычислительных систем. Данная система является основой для построения многозадачной операционной системы для реконфигурируемых систем с целью минимизации прохождения потока заданий. Приведена структура системы масштабирования параллельных программ, взаимодействие ее компонентов.

Научно-технический прогресс требует решения все более сложных вычислительных задач за все более короткие промежутки времени. Производительности однопроцессорных систем уже недостаточно для обеспечения постоянно растущего спроса на высокопроизводительный вычислительный ресурс. С этой целью создаются новые архитектуры многопроцессорных вычислительных систем (МВС), способы их технической реализации, а также методы организации вычислений и программирования. Множество организаций и суперкомпьютерных центров на сегодняшний день имеют в своем распоряжении высокопроизводительные многопроцессорные системы и большинство из них сталкиваются с важной проблемой эффективного использования вычислительного ресурса. Рациональное использование МВС диктует необходимость решения самых различных задач: от трудоемких вычислительных, требующих полной загрузки вычислительного ресурса, до задач, требующих небольшого количества процессоров и решаемых за относительно короткие промежутки времени. В связи с этим важным является эффективное использование и планирование вычислительного ресурса с целью минимизации времени прохождения неоднородного потока заданий.

Существуют два основных метода разделения вычислительного ресурса: метод разделения времени и метод разделения ресурса. Первый метод заключается в выделении квантов времени на использование ресурса системы потребителям последовательно, и эффективно используется ОС однопроцессорных систем, когда процессорное время делится между потоками. Основным недостатком данного метода

является существенное падение производительности при росте числа потребителей из-за накладных расходов на переключение между ними.

Метод разделения ресурса предполагает “пространственное” разделение ресурса между потребителями. Качество разделения выполняет специальная процедура планирования. Данный метод преимущественно применяется в многопроцессорных системах. Однако на этапе разработки параллельных программ отсутствует информация программисту о количестве ресурса, которое будет выделено планировщиком заданий ОС для выполнения его программы. Традиционный подход решают данную проблему путем перекомпиляции исходного кода программ, учитывая на этапе трансляции количество выделенного ресурса непосредственно перед подстановкой на исполнение. Однако за время перекомпиляции, исчисляемое минутами, может измениться динамическая карта состояния ресурса и, как следствие, решение о постановке задания на выбранный ресурс станет неактуальным. При этом возможны конфликтные ситуации, и неизбежно неэффективное использование ресурса МВС.

В связи с этим функции масштабирования параллельных программ целесообразно вынести на уровень операционной системы и выполнять их в автоматическом или полуавтоматическом режимах. Уровень операционной системы подразумевает работу с исполняемым модулем программы без модификации ее исходного представления. В то же время средств автоматического распараллеливания уровня ОС на сегодняшний день практически не существует. Имеющиеся средства существенно снижают производительность МВС.

Еще более остро ситуация с распределением ресурсов в многозадачном режиме стоит с наиболее перспективными и бурно развивающимися в последнее время реконфигурируемыми вычислительными системами. Такие системы потенциально имеют высокий запас производительности за счет более рационального использования ресурсов системы, адаптированных под структуру решаемой задачи и оригинальных методов создания параллельных программ [1]. Структурная организация вычислительного процесса дают возможность создавать масштабируемые параллельные программы для реконфигурируемых систем, а функции масштабирования выполнять без коррекции исходного кода программ.

Для реконфигурируемых систем существует технология индуктивных параллельных программ [1]. Она подразумевает включения в параллельную программу помимо основной вычислительной компоненты еще и правила модернизации их в зависимости от степени распараллеливания. В этом случае программы

разрабатываются независимо от степени распараллеливания, а масштабирование выполняется процедурой посттрансляции, входящей в состав ОС и выполняющей модернизацию исполняемого кода параллельных программ. Основной недостаток данной технологии состоит в существенном повышении требованиям к квалификации программистов, разрабатывающих индуктивные программы, т.к. трудоемкость написания правил масштабирования соизмерима, а зачастую и превышает трудоемкость создания основной вычислительной компоненты программы.

В связи с этим актуальной является задача создания методов и средств автоматического масштабирования параллельных программ для реконфигурируемых вычислительных систем.

На основе принципов организации вычислений в реконфигурируемых вычислительных системах и технологии индуктивных параллельных программ предлагается структура системы автоматического масштабирования параллельных программ (система посттрансляции) для реконфигурируемых систем (см. рисунок 1). Рассмотрим структуру системы подробнее.

Исходный код параллельной программы поступает в транслятор. На этапе формирования объектного представления программы, транслятор определяет фрагменты программы, возможные для распараллеливания. Определение производится на основе требований к их описанию в библиотеке схем распараллеливания. Каждой схеме распараллеливания в библиотеке сопоставлено соответствующее значение типа распараллеливания. При формировании исполняемого кода программы фрагменты необходимые для модернизации помечаются определенными маркерами с целью дальнейшего анализа процедурой посттрансляции. Маркер, в первую очередь, несет в себе информацию о типе схемы распараллеливания.

Далее сформированный транслятором исполняемый код анализируется процедурой посттрансляции. Фрагменты кода, помеченные маркером, модернизируются соответствующей процедурой из библиотеки схем распараллеливания и параметром распараллеливания, выделенным планировщиком заданий [2]. Модернизированный (масштабированный) исполняемый код поступает на загрузчик программ. Загрузчик заполняет сегменты памяти базовых модулей соответствующими командами и данными. В память модулей загружена уже распараллеленная программа.



Рисунок 1 – Структура системы посттрансляции

Схема взаимодействия процедуры посттрансляции и библиотеки схем распараллеливания приведена на рисунке 2. Исполняемый код программы поступает в процедуру посттрансляции, разбивается на секции. Далее, если секция помечена как необходимая для модернизации транслятором, то в библиотеку схем распараллеливания передается адрес секции и ее тип. В соответствии с типом секции в библиотеке находится соответствующая процедура модернизации секции. Модернизация производится на основе параметра распараллеливания выделенным планировщиком заданий и с помощью служебных функций из процедуры посттрансляции. В результате модернизации библиотека возвращает масштабируемый код секции, готовый для загрузки в память базовых модулей. Данные действия повторяются для всех секций в исполняемом файле программы.



Рисунок 2 – Схема взаимодействия библиотеки схем распараллеливания и процедуры посттрансляции

Алгоритм процедуры посттрансляции в общем случае для автоматически масштабируемых программ приведен ниже.

1° Открытие исполняемого модуля программы.

2° Получение начального адреса очередной секции в исполняемом модуле.

3° Если секция помечена как необходимая для модернизации, то считается значение типа секции.

4° По значению типа секции производится запрос о наличии соответствующей процедуры масштабирования в библиотеке.

5° Если соответствующая типу секции процедура масштабирования найдена, то производится ее вызов. Основными параметрами вызова процедура являются:

- количество модулей выделенных для решения задачи;
- адрес модернизируемой секции.

6° Результатом работы процедуры является модернизированная секция в исполняемом файле параллельной программы.

7° Переход к п.2.

Система посттрансляции дает возможность построить многозадачную операционную систему с целью минимизации прохождения потока заданий через реконфигурируемые вычислительные системы. Кроме того, система существенно снижает требования к программистам для создания масштабируемых параллельных программ для реконфигурируемых систем. Анализ показывает, что задержки вносимые системой посттрансляции составляют не более 10% от общего времени постановки задания на решение и оцениваются как не существенное.

1. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Изд-во ООО «Янус-К», 2003. – 325 с.
2. Каляев З.В. Структура многозадачной распределенной операционной системы // Материалы седьмой Международной научно-технической конференции «Искусственный интеллект. Интеллектуальные и многопроцессорные системы». – Таганрог: Изд-во ТРГУ, 2006. – Т. 2. - С. 102-103.