

Современное требование уменьшения времени, необходимого для выпуска конечного продукта, заставляет искать пути сокращения как времени выполнения любых отдельных этапов проектирования, так и устранения причин, вызывающих необходимость итерационных возвратов к предыдущим этапам проектирования.

Рассмотренные проблемы перевода программно реализованных алгоритмов в аппаратуру и современные требования к проектированию ПОВС привели к появлению таких новых подходов к проектированию, как сопряженное проектирование, верификация и моделирование.

Со-проектирование (сопряженное проектирование) – процесс параллельного проектирования аппаратных и программных средств, при котором осуществляется оценка целесообразности выбора аппаратной или программной реализации определенного фрагмента проекта. Этот процесс также дает проектировщикам возможность, увидев (на абстрактном уровне), как система могла бы работать с данным разделением аппаратных средств ЭВМ и программного обеспечения.

Уже существуют наборы программ, содержащие мощные инструментальные средства проектирования (например, Celoxica DK) и представляющие из себя системную среду разработки алгоритмических акселераторов и комплекса конструирования электронных устройств.

Предлагаемый маршрут проектировщика предоставляет разработчику комплексную интегрированную среду разработки, основанную на языках высокого уровня (ЯВУ), таких как C, C++, Pascal, Object Pascal, Java, и включает в себя полный процесс проектирования устройства от этапа ввода схемы, моделирования и заканчивая синтезом. Основной особенностью является совместимость с Transaction Level Model (TLM), что позволяет смешанное абстрактное моделирование и симуляцию программно-аппаратного окружения, которая позволяет итерационно уточнять уровень моделирования устройства.

Преимуществами использования данного подхода являются: увеличение производительности при конструировании аппаратно-программных комплексов, сокращение времени на разработку и улучшение общего качества проектирования. Все участники процесса конструирования устройства от проектировщиков системы и разработчиков аппаратных средств до разработчиков программного обеспечения и тестирующих могут четко разделить задачи от уровня системной спецификации до реализации.

**К.П. Цурко**

### **РЕАЛИЗАЦИЯ АЛГОРИТМА КОМПОНОВКИ НА ОСНОВЕ МЕТОДА ДИХОТОМИИ И МЕТОДА ВЕТВЕЙ И ГРАНИЦ**

Цель данной работы состоит в реализации и исследовании алгоритма дихотомического деления схемы, то есть разбиения графа на две равные части, с помощью метода ветвей и границ, который гарантирует нахождение оптимального решения, уходя от полного перебора всех вариантов решений.

Достижение указанной цели предполагает решение следующих основных задач:

- ◆ Построение графовых моделей схем ЭВА для этапа конструкторского проектирования.
- ◆ Исследование алгоритма дихотомического деления с помощью метода ветвей и границ.
- ◆ Реализация алгоритма на языке программирования высокого уровня.

Результатом проделанной работы является реализованный на языке программирования высокого уровня программный продукт, который позволяет решать

задачу дихотомического деления графа с помощью метода ветвей и границ. Были описаны способы перехода от схем к графовым моделям и выбран наиболее оптимальный способ для решения поставленной задачи. Были проведены исследования алгоритма и оценены его временная сложность и эффективность.

При проведении анализа и исследования характеристик алгоритма были использованы графы с различным количеством вершин и с 70-ти процентным заполнением рёбрами. В результате были проведены эксперименты, которые показали, что временная сложность алгоритма (ВСА) равна  $n^2$ .

**В.Е. Золотовский, В.Б. Резников**

### **РАСПАРАЛЛЕЛИВАНИЕ РАСЧЕТА СЛОЖНЫХ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ В РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ**

Для решения задачи автоматизации распараллеливания расчета математических моделей был разработан метод, основывающийся на статическом распределении иерархического описания сложных многокомпонентных моделей с учетом неоднородности производительности вычислительных узлов (распределенного вычислителя) кластера.

Эта проблема решается путем введения в систему моделирования, информации об архитектуре её базового вычислителя. С этой целью помимо модели, исследуемой физической системы, так же строится модель базового вычислителя, которая представляется при помощи неориентированного взвешенного графа. Для более точной оценки уровня равномерности распределения вычислительной нагрузки данная графовая модель вычислителя дополняется эвристической моделью, которая задается в виде двух таблиц: вычислительных затрат ( $T_c$ ) и обменных затрат ( $T_e$ ).

В качестве исходных данных, кроме модели распределенного вычислителя, предлагаемый метод рассматривает функциональный и математический уровни структурной модели. На функциональном уровне структурная модель представляется в виде иерархического графа. Подграф которого представляет собой направленный граф со взвешенными вершинами и ребрами, отображающими совокупность элементарных математических моделей и их взаимодействие.

На основе разработанных моделей были синтезированы оценки времени вычисления компонент структурной модели с учетом особенности методов численного решения. Данные оценки были использованы в алгоритме автоматического распределения компонент сложных моделей в неоднородной распределенной вычислительной сети.

Основу алгоритма составили два основных этапа: размещение всех компонент сложной модели в вычислительном узле, обладающим максимальной производительностью; последовательная декомпозиция структурной модели по другим вычислительным узлам, начиная с самых трудоемких компонент согласно полученным оценкам. На каждом шаге декомпозиции производится вычисление эффективности текущего варианта распределения. При необходимости наиболее сложные компоненты модели декомпозируются на составляющие их элементарные модели, которые подвергаются распределению до тех пор, пока будет возможно увеличить общую эффективность распределения.

Таким образом, представленный метод позволяет автоматизировать процесс распределения иерархических структурных моделей в неоднородной вычислительной среде, что достигается путем использования эвристических оценок вычислительной сложности математического описания и построению модели вычислителя.