

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕЛКОЗЕРНИСТЫХ АЛГОРИТМОВ В СИСТЕМЕ WINALT

М.Б. Остапкевич

Введение

Проблемная область. Мелкозернистый параллелизм (МЗП), наряду с крупноблочным, является одним из двух основных видов параллелизма. Его отличительная черта - огромное число простых параллельно работающих процессорных элементов. К широко известным классам алгоритмов с мелкозернистым параллелизмом относятся классический клеточный автомат (КА) и его расширения (КА с окрестностью Марголуса [1], асинхронный КА, вероятностный КА), сеть КА, клеточно-нейронная сеть, дискретная динамическая сеть, системы Линденмайера. Алгоритмы и структуры с МЗП используются для моделирования физических [1], химических, биологических и социальных процессов, разнообразных вычислительных устройств (ассоциативный процессор, систолическая структура, однородная среда, универсальная вычислительная среда, конвейеры арифметических устройств).

Объем данных и сложность правил их трансформации в МЗП моделях делает практически невозможным их исследование без системы моделирования. Известно несколько десятков систем моделирования МЗП [2, 3]. Их общий недостаток - работа только с классическим КА и некоторыми его расширениями. Для моделирования всего спектра МЗП алгоритмов была построена среда моделирования WinALT [4]. Цель этого проекта – разработка системных алгоритмов и реализация системы моделирования WinALT на широком спектре доступных параллельных вычислительных средств. Такая реализация обеспечит возможность работы с большими объемами данных и вычислений, характерными для большинства практически полезных моделей.

В статье дана общая характеристика среды моделирования WinALT, её текущее состояние, описана существующая подсистема параллельного исполнения WinALT (ParSIM/Win32) и сформулированы требования к проектируемой подсистеме параллельного исполнения (ParSIM/Omni), предложена ее архитектура и сформулированы планы развития системы.

Общая характеристика среды моделирования WinALT

Среда моделирования WinALT реализует набор пользовательских функций, позволяющих создавать, редактировать, отлаживать и исполнять модели МЗП алгоритмов. Среда предоставляет возможности как визуального проектирования моделей, так и построения их аналитических описаний. Визуальный подход позволяет представлять графически не только данные моделей, но и правила их трансформации. Аналитический подход позволяет описать любые аспекты работы моделей (как содержательные, так и технические) на языке моделирования системы.

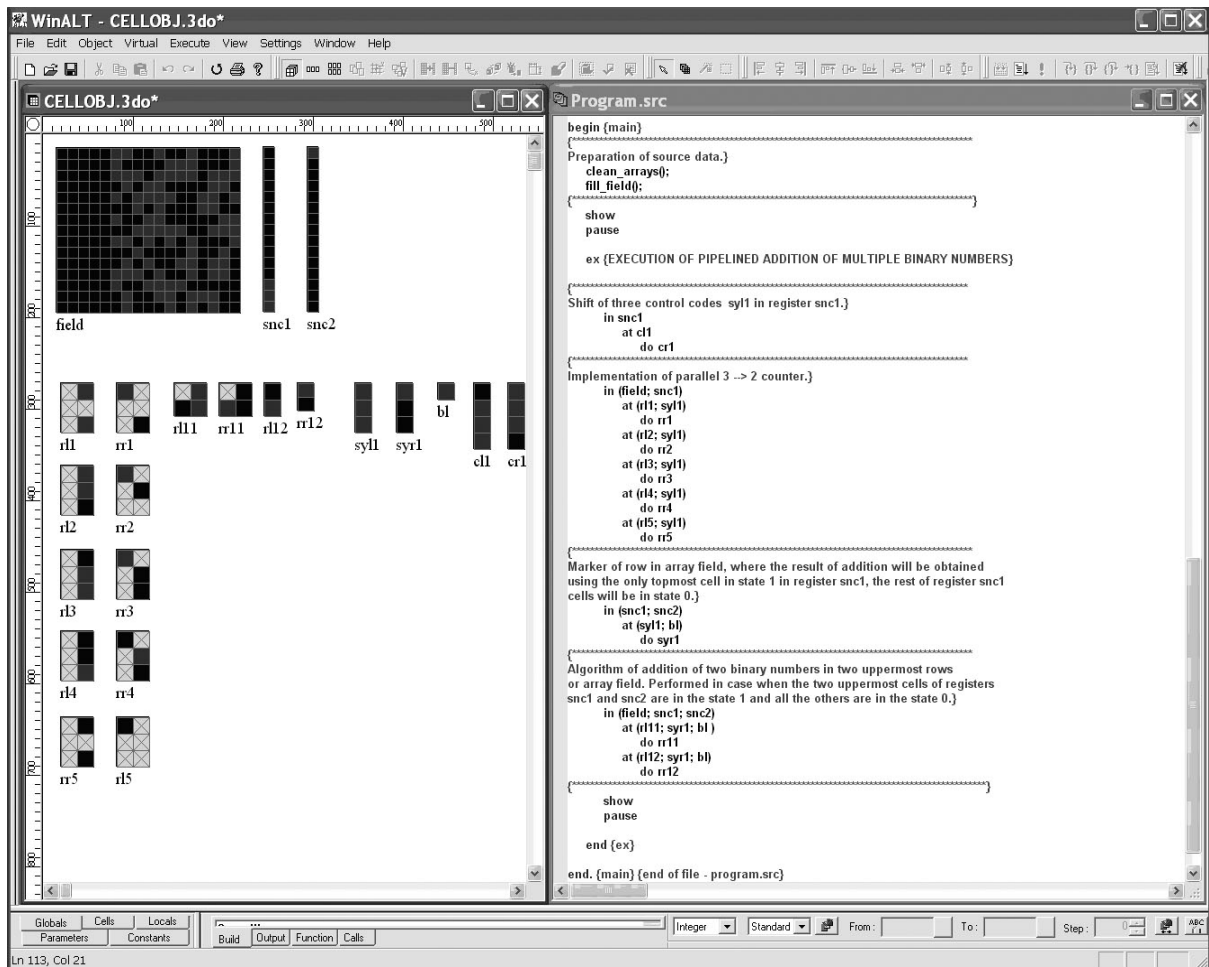


Рис. 1. Снимок экрана с примером модели сумматора многих чисел 3 в 2

Модель на рис.1 имитирует работу сумматора многих чисел. В левой части окна показаны объекты данных (клеточные массивы и шаблоны подстановок) модели. До исполнения модели строки клеточного массива field содержат двоичные числа. После исполнения все строки клеточного массива field, кроме верхней, содержат нули. Верхняя строка содержит результат суммирования. Клеточные массивы snc1 и snc2 - статусные регистры, показывающие для каких строк выполнение сложения завершается. Все остальные объекты данных описывают левые и правые части подстановок (шаблоны подстановок). Левая часть определяет условие применимости подстановки, а правая - новое состояние клеток, изменяемых подстановкой. В правой части окна показана моделирующая программа сумматора. Она состоит из двух частей - сервисной и содержательной. Сервисная служит для заполнения обрабатываемых объектов начальными данными (вызов процедур clean_arrays и fill_field). Содержательная часть описывает необходимые подстановки и режим их применения. Оператор ex задает синхронный режим исполнения подстановок до их неприменимости. Оператор in указывает имя клеточного объекта (или имена нескольких клеточных объектов для более сложных подстановок), который будет трансформироваться описываемой подстановкой. Оператор at указывает шаблон (или шаблоны) левой части подстановок. Оператор do задает шаблон (или шаблоны) правой части подстановки.

Дистрибутив среды моделирования и коллекция примеров моделей доступны на [2]. Среда имеет открытую архитектуру, что позволяет пользователям добавлять в нее новые функции, форматы данных, режимы отображения, режимы моделирования и фрагменты моделей и формировать из них библиотеки.

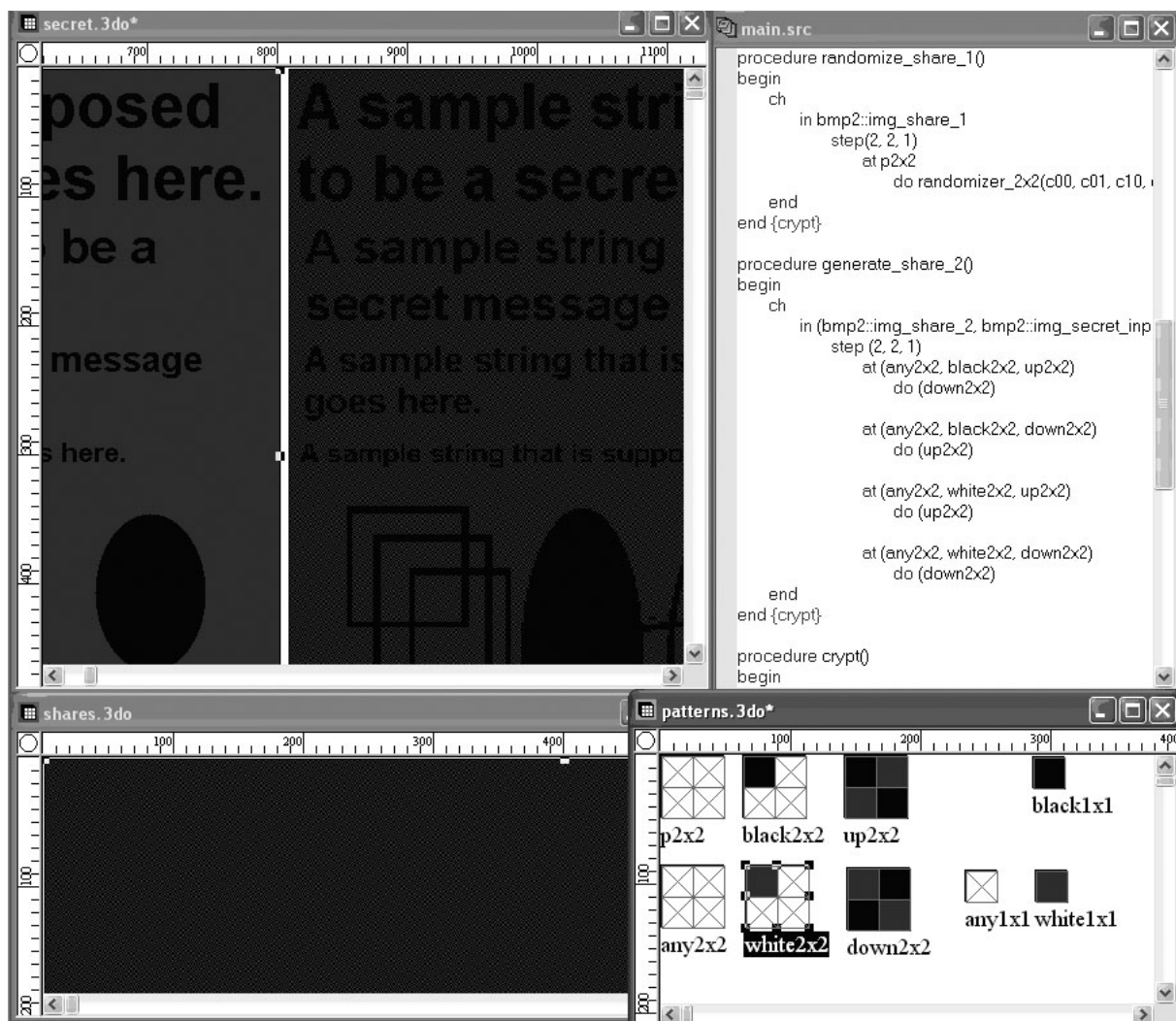


Рис. 2. Пример модели визуальной криптографии

В частности, библиотека форматов данных содержит поддержку форматов растровых изображений. На рис. 2 показана простейшая модель визуальной криптографии [5], которая использует их для хранения начальных, промежуточных и конечных данных. Левая часть окна secret.3do содержит исходное изображение, которое требуется зашифровать. Два изображения в окне shares.3do, одно из которых видно на рисунке, хранят изображение в зашифрованном виде. Изображение в правой части окна secret.3do содержит результат дешифрации.

Среда моделирования WinALT представлена следующими основными подсистемами:

- 1) консольная версия системы моделирования (исполняет модели без отладки, визуализации и взаимодействия с пользователем);
- 2) графическая среда WinALT (обеспечивает визуальное построение, отладку и исполнение моделей, имеет развитые средства взаимодействия с пользователем);
- 3) Web среда (содержит дистрибутив системы, документацию, коллекцию моделей, массив информации по тематике МЗП);
- 4) Подсистема параллельного исполнения ParSIM/Win32 (исполнение моделей на параллельных вычислительных системах).

ParSIM/Win32 реализована как приложение, исполняемое на невыделенном кластере компьютеров с 32 битной платформой Windows (Win32). Для передачи данных и координации работы был разработан собственный протокол. Его реализация базируется на Windows socket library. ParSIM/Win32 обеспечивает параллельное исполнение без каких-либо изменений в моделирующих программах для моделей, в которых все постановки имеют локальные окрестности 3x3 клетки;

ParSIM/Win32 с одной стороны показала большую полезность для уменьшения времени моделирования, увеличения допустимых размеров объектов данных в моделях. С другой стороны ее использование выявило недостатки, главные из которых:

- 1) реализация на платформе Windows, перенос которой на другие платформы трудоемок;

- 2) негибкая архитектура виртуальной машины (модуля, которые управляет исполнением любой модели в системе), которая не допускает ее перенос на ряд современных архитектур (GPU) при сохранении достаточной эффективности;
- 3) поддержка распараллеливания только такого класса моделей, в которых все постановки имеют локальные окрестности 3x3 клетки;
- 4) привязка к единственному собственному протоколу передачи данных и координации работы удаленных процессов распределенного приложения.

С учетом указанных недостатков ParSIM/Win32 актуально построение новой подсистемы параллельного исполнения (ParSIM/Omni), преодолевающей их.

Главные требования к ParSIM/Omni следующие:

- 1) эффективное исполнение широкого спектра МЗП моделей на мультикомпьютерах в целом и на отдельных счетных узлах в частности;
- 2) использование существующей сетевой инфраструктуры и компьютеров с разнообразными конфигурациями, архитектурой и набором системного программного обеспечения без перенастройки WinALT;
- 3) поддержка различных протоколов доступа и обмена данными для взаимодействия с кластерами и между вычислительными устройствами,
- 4) возможность исполнения моделей, содержащих объекты данных, не уместяющихся целиком в основной/виртуальной памяти.

Архитектура ParSIM/Omni

Предлагаемая архитектура с одной стороны призвана удовлетворить предъявленным требованиям, а с другой стороны в максимальной степени использовать уже существующие реализации модулей среды моделирования.



Рис. 3. Основные компоненты ParSIM/OMNI

На верхнем уровне можно выделить следующие основные компоненты (см. рис. 3):

- 1) виртуальная машина;
- 2) менеджер объектов;
- 3) менеджер режимов моделирования;
- 4) менеджер протоколов и интерфейсов.

Первые два из них имеются в существующей среде WinALT и, в частности, в ParSIM/Win32, но их свойства и набор функций претерпел изменения. Остальные являются новыми компонентами.

Виртуальная машина управляет исполнением моделирования, пользуясь операциями, реализованными во всех остальных основных компонентах. Новая версия виртуальной реализует не только операцию запуска модели на исполнение, но также и операции приостановки, возобновления и прекращения исполнения, посылки сигналов о событиях исполняемой модели. Кроме этого, новая версия позволяет одновременное исполнение нескольких процессов внутри одной виртуальной машины. Хотя набор инструкций виртуальной машины максимально приближен к уже имеющемуся, минимальный набор обязательных инструкций существенно сокращен, что делает ее реализацию на новых платформах, включая GPU, менее трудоемкой.

Менеджер объектов обеспечивает унифицированный доступ к клеточным объектам данных по имени клеточного объекта и координатам клетки вне зависимости от используемых форматов данных и протоколов доступа. Новыми свойствами менеджера являются: 1) возможность частичной загрузки объекта в виртуальную память (необходимо для работы со сверхбольшими объектами данных), 2) возможность загрузки и сохранения объектов, не являющихся файлами локальной файловой системы, а доступных по какому-либо протоколу.

Менеджер режимов моделирования реализует операции поиска и применения подстановок. Обеспечивает возможность пользователю:

- 1) выбирать более одного режима синхронности внутри одной модельной программы,

- 2) использовать в качестве объектов данных не только клеточные массивы, но и иные структуры, например, графы,
- 3) использовать разные режимы распараллеливания.

В ParSIM/Win32 есть только режим распараллеливания с разрезанием объектов данных и их распределением по счетным узлам (см. рис. 4). Он является примером параллелизма по данным и обеспечивает эффективное распараллеливание только для моделей с небольшими локальными окрестностями правил подстановки.

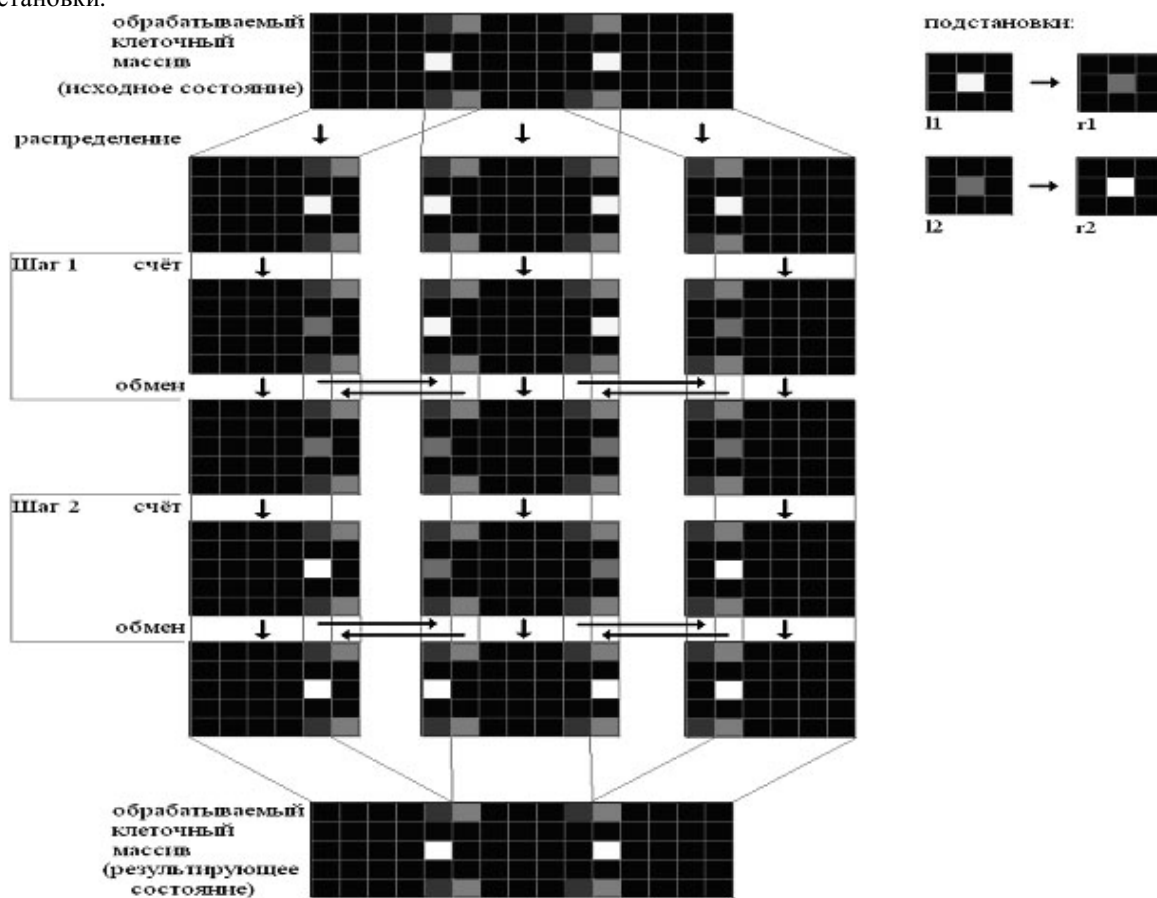


Рис. 4. Режим параллельного исполнения моделей с разрезанием объектов данных.

При использовании режима разрезанием объектов данных перед параллельным исполнением происходит разделение обрабатываемого клеточного массива на части с перекрытием (с теньвыми клетками). Их число определяется числом параллельных процессов обработки. Во многих случаях оно совпадает с числом счетных узлов. Будем рассматривать именно такой случай. Область наложения задает те клетки, которые размещаются в двух или более соседних счетных узлах. Геометрия этой области определяется топологией разрезания клеточного массива (линейка, кольцо, 2D, 3D решетка или тор) и максимальным размером окрестности имеющихся в модели правил подстановки, применяемых к данному клеточному массиву.

После разрезания клеточного массива, его части передаются соответствующим счетным узлам, и начинается параллельное исполнение. Оно происходит итеративно до тех пор, пока есть клетки, значение которых менялось на последнем шаге. Каждый шаг состоит из двух фаз - счета и обмена. После завершения параллельного исполнения части клеточного массива собираются в целый массив в виде файла во внешней памяти.

В ParSIM/Omni вводятся дополнительные режимы распараллеливания, которые расширяют класс эффективно исполняемых МЗП моделей. Множество этих режимов может расширяться как разработчиком системы, так и ее пользователями имеющими навыки программирования, так как невозможно заранее предусмотреть все достаточные для моделей любого пользователя режимы. Изначально имеется реализация трех режимов параллельного исполнения. Первый из них заимствован из ParSIM/Win32 и рассмотрен выше. Два новых режима рассматриваются далее.

Режим с разбиением множества подстановок (см. рис. 5), являющийся примером функционального параллелизма, позволяет эффективно распараллеливать исполнение моделей, в которых есть большие окрестности в правилах подстановки, но при этом доля изменяемых клеток от общего их числа достаточно мала.

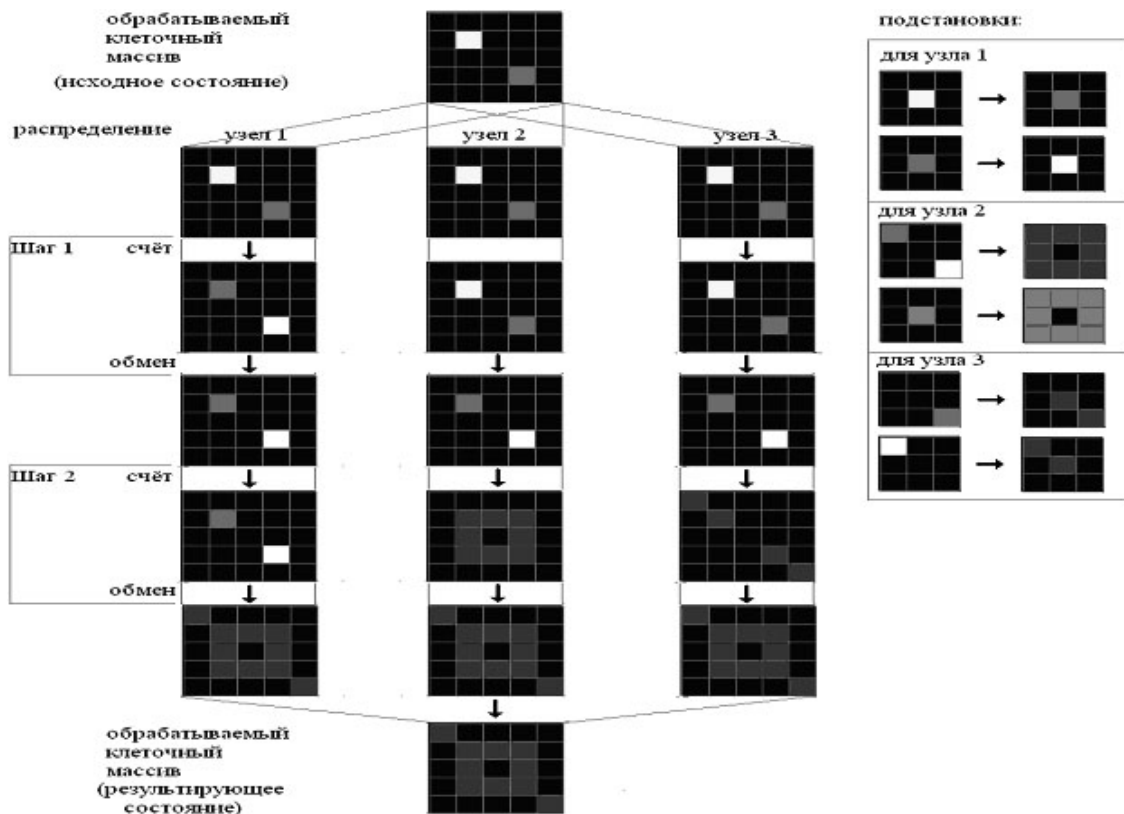


Рис. 5. Режим параллельного исполнения с разбиением множества подстановок.

В этом режиме перед параллельным счетом все узлы вычислителя получают одинаковую полную копию обрабатываемого массива. Множество всех подстановок делится на непересекающиеся подмножества. Их число определяется числом параллельных процессов обработки. Во многих случаях оно совпадает с числом счетных узлов. Будем рассматривать именно такой случай. Каждый узел получает свой комплект подстановок. Параллельное исполнение происходит итеративно с синхронизацией вычислений между итерациями для проведения обменов. На фазе счета каждый узел отыскивает все применимые подстановки из своего комплекта, вычисляет новые значения изменяемых клеток и формирует список всех измененных клеток. На фазе обмена каждый узел рассылает всем другим узлам значения всех модифицированных клеток. Если списки модифицированных клеток пусты на всех узлах, то результат получен, и параллельное вычисление прекращается.

Для случая, когда объем данных умеренный, но есть большой разброс времени обработки одной применимой подстановки, оправдано использование режима параллельного исполнения с разбиением обрабатываемых данных и вычислению по готовности данных (см. рис. 6).

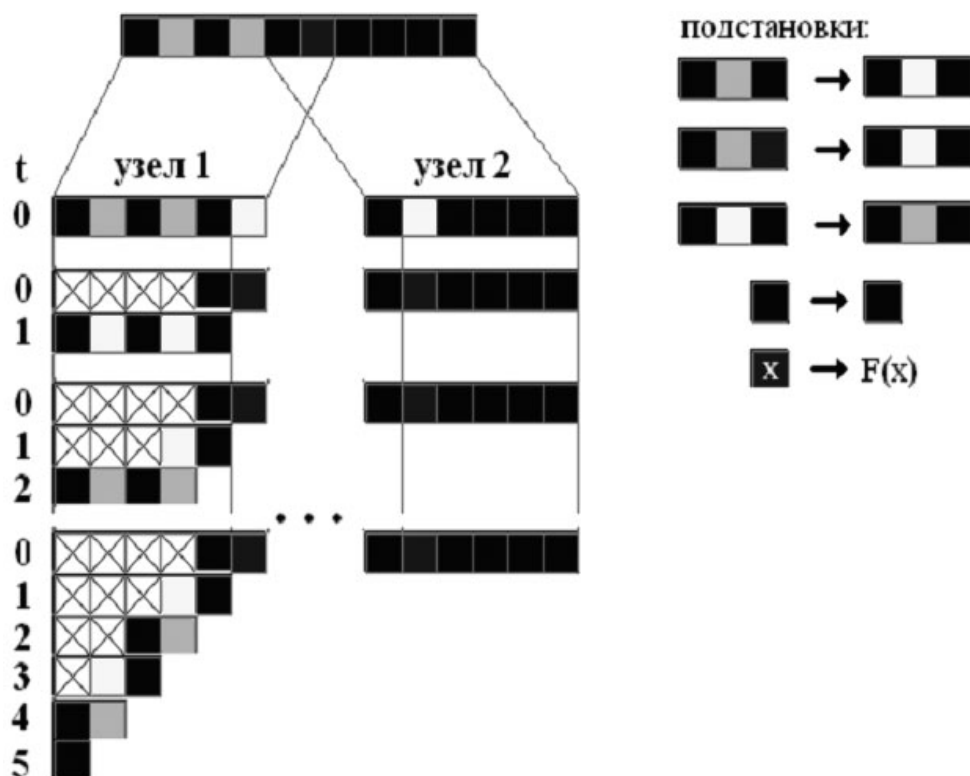


Рис. 6. Режим параллельного исполнения разбиением обрабатываемых данных и вычислению по готовности данных.

В этом режиме возможно хранение значений одного элемента данных для разных итераций. Это дает возможность продолжать счет для внутренних областей клеток даже в тех случаях, когда новые значения клеток в области пересечения еще не получены от тех узлов, которые ответственны за обработку этих клеток. В этом режиме нет жесткого чередования фаз счета и обмена. Они могут реализовываться отдельными потоками команд, исполняемыми на счетном узле в режиме разделения времени.

Заключение

На настоящий момент реализована новая версия виртуальной машины и разработаны базовые системные алгоритмы для реализации остальных компонентов. Разрабатываются остальные компоненты верхнего уровня ParSIM/Omni. В первую очередь запланировано завершение реализации ParSIM/Omni на базе OpenCL.

Реализация ParSIM/Omni позволит расширить класс моделируемых в среде WinALT явлений и повысить точность результатов моделирования за счет увеличения размеров обрабатываемых объектов данных. Приблизительная оценка предельных размеров клеточных массивов в ParSIM/Win32 составляет 30000x30000 клеток для 2D и 1000x1000x1000 для 3D. В ParSIM/Omni она составит 1000000x1000000 клеток для 2D и 10000x10000x10000 для 3D. Для многих моделей, например из области теории протекания [6], подобное увеличение размеров приведет к существенно более точным и качественным результатам моделирования. Для ряда других моделей, таких как каталитические реакции получение результата в принципе невозможно при малом числе клеток. В [7] указывается, что требуемое число взаимодействующих молекул составляет 1020.

Также, реализация ParSIM/Omni позволит увеличить объем производимых вычислений как за счет возможности использования высокопроизводительных параллельных вычислительных комплексов, так и за счет возможности виртуальной машины сохранять состояние процесса моделирования и возобновлять моделирование после сбоев различного характера.

ЛИТЕРАТУРА:

1. Т. Тоффоли, Н. Марголус. - Машины клеточных автоматов. - М. - Мир. - 1991
2. Cellular Automata Software. - <http://cafaq.cafaq.com/soft/index.php>
3. Сайт среды моделирования WinALT. - <http://winalt.sccc.ru/>
4. М. Ostapkevich, S. Piskunov. - The Construction of Simulation Models of Algorithms and Structures with Fine-Grain Parallelism in WinALT. - in PaCT 2011, V. Malyshevkin (Ed.). - LNCS 6873. - Springer-Verlag. - 2011. - pp. 192-203
5. М. Naor, А. Shamir. - Visual Cryptography. - Advances in Cryptology, EUROCRYPT'94. - 1995. - pp. 1-12
6. А.Л. Эфрос. - Физика и геометрия беспорядка. - Библиотека Квант. - вып. 19. - М. - Наука. - 1982

7. А.Е. Шарифулина. - Параллельная реализация каталитической реакции ($\text{CO} + \text{O}_2 \rightarrow \text{CO}_2$) / Pt110 с помощью асинхронного клеточного автомата. - Сборник трудов Международной научной конференции "Параллельные вычислительные технологии 2012". - С. 325-335