

**Отчет по этапам НИР, завершённым в 2012 г.
в соответствии с планом НИР института**

Проект НИР 1.3.2. «Параллельные и распределенные вычисления в задачах математического моделирования»

Проект НИР 1.3.2.1. «Параллельные и Грид-технологии реализации задач математического моделирования на супер-ЭВМ» (№ госрегистрации 01201002444).

Научный руководитель – д.т.н. В.Э. Малышкин.

Раздел 1. Алгоритмы и средства конструирования параллельных программ численного моделирования. Разработка наукоёмких программных систем.

Руководитель – д.т.н. В.Э. Малышкин.

Этап 1.3. Разработка сборочного программного обеспечения

Разработка прототипа пользовательского интерфейса на основе веб-технологий.

Разработаны модели командного пользовательского интерфейса и веб-интерфейса для пользователей и администраторов системы. Разработана архитектура сервера управления задачами для метакомпьютеров NumGRID (вычислительных систем, построенных на основе объединения вычислительных кластеров с помощью инструментария NumGRID). Программное обеспечение NumGRID реализует единую коммуникационную среду для процессов, выполняемых на вычислительных узлах объединяемых кластеров. Сервер хранит информацию о кластерах, входящих в состав объединения, способах доступа к ним и периодически опрашивает их состояние, на основании полученных данных формируя состояние метакомпьютера.

Административный интерфейс является типовым графическим Web-интерфейсом, через который администратор метакомпьютера может добавлять кластеры к интерфейсу, управлять пользователями интерфейса и выдавать/отбирать разрешения на запуск задач на определённых кластерах.

Разработан язык описания задачи на основе синтаксиса, принятого в PBS-подобных системах управления прохождением задач. Задача описывается как набор пронумерованных «частей». Основные параметры части: количество групп процессоров и количество процессорных ядер в каждой группе для исполнения данной части, ее связь с другими частями: указывается тип связи (hard – обязательно размещение в одном кластере, soft – желательное размещение в одном кластере) и номера связанных частей. Разработка методики адаптации программ численного моделирования к работе в среде NumGRID.

При разработке методики адаптации к неоднородным вычислительным системам типа NumGRID учитываются различия в производительности процессорных элементов, в устройстве вычислительных узлов (количество процессоров, ядер, наличие сопроцессоров), а также в пропускной способности каналов связи между вычислительными узлами. В ходе выполнения программы может изменяться соотношение долей работы в соответствии с природой моделируемых явлений и особенностями применяемых вычислительных методов. Так, в ходе моделирования горения газа на равномерной прямоугольной сетке обсчет узлов сетки, примыкающих к фронту горения, может занимать больше времени, чем обсчет узлов сетки, где горения уже есть или еще нет. Для поддержания равномерности распределения работы требуется применение алгоритмов динамической балансировки вычислительной нагрузки. Решение волнового уравнения на сетке относится к другому классу задач, где обеспечение баланса распределения работы осуществляется правильным распределением в начале работы программы.

Существуют задачи, которые не могут быть эффективно решены на NumGRID из-за высоких требований к скорости и задержке передачи данных между процессами. Для решения таких задач полезно развивать механизмы, которые автоматизируют обеспечение динамических свойств параллельных программ. Это делается в системе фрагментированного программирования, разрабатываемой в лаборатории. Система обладает высоким потенциалом автоматизации динамического распределения вычислений, реализации коммуникаций на фоне счета и может использоваться для разработки эффективных программ для NumGRID.

Разработка алгоритмов использования профиля для оптимизации исполнения фрагментированных программ. Разработка алгоритмов адаптации прикладной программы к вычислителю и алгоритмов преобразования профильной информации в индексированную форму.

В системе фрагментированного программирования LuNA исполнение прикладной программы автоматически оптимизируется по времени путём её настройки на конфигурацию аппаратного обеспечения с учётом структуры вычислений прикладной программы (самообучение программы). Для осуществления такой настройки используется профилирование, которое заключается в том, что по ходу исполнения фрагментированной программы фиксируется информация о времени исполнения фрагментов вычислений, называемая профилем исполнения фрагментированной программы. Выполняется анализ профиля с целью определения простоев оборудования, вызванного недостатком нагрузки на процессоры, после чего выявляются причины возникновения простоев и осуществляется корректировка исполнения прикладной программы при последующих запусках.

Разработан алгоритм свертки профильной информации в индексную форму. При профилировании реальных фрагментированных программ размер профиля исчисляется миллионами записей, что затрудняет анализ профиля. В связи с этим был разработан алгоритм, осуществляющий свёртку записей профиля заданного вида в ограниченный набор индексированных записей, каждая из которых задаёт множество записей исходного набора. Алгоритм свертки реализован в виде отдельного модуля для системы фрагментированного программирования LuNA.

Построение событийно-ориентированного интерфейса, обеспечивающего встраивание в информационно-вычислительную среду пользователя системы WinALT внешних модулей.

Построен интерфейс для встраивания внешних модулей, реализующих фрагменты пользовательского интерфейса, в информационно-вычислительную среду пользователя системы имитационного моделирования мелкозернистых алгоритмов и структур WinALT. Разработаны внешние модули, реализующие фрагменты режимов отображения объектов: 1) с цветовым выделением, 2) с печатью числового значения клетки, 3) с показом значения в виде стрелки с направлением, 4) с показом клеточных объектов на прямоугольной сетке, 5) с показом клеточных объектов на гексагональной сетке, 6) композиционного режима отображения, позволяющего совмещать несколько других режимов для показа одного клеточного объекта. Интерфейс реализует встраивание в систему расширяемого множества внешних модулей, реализующих все основные аспекты реализации пользовательского интерфейса (режимы отображения и редактирования данных, диалоговые окна, меню, панели инструментов), что обеспечивает адаптацию среды пользователя к программно-аппаратной платформе, предметной области.

Раздел 2. «Разработка клеточной технологии моделирования физических и информационных процессов с использованием супер-ЭВМ».
Руководители – к.т.н. С.В. Пискунов, д.т.н. О.Л. Бандман.

Этап 2.3. Реализация крупномасштабных моделей на супер-ЭВМ.

Разработка и исследование методов распараллеливания асинхронных вероятностных клеточно-автоматных моделей и их многопроцессорная реализация.

Проведено исследование эффективности распараллеливания асинхронных вероятностных клеточно-автоматных моделей. Путем проведения серии вычислительных экспериментов по моделированию химической реакции окисления монооксида углерода на каталитической поверхности были получены конкретные значения эффективности параллельных реализаций на всех доступных параллельных архитектурах: кластере, 32-ядерной машине, графических ускорителях. Полученные результаты показали, что без проведения предварительных преобразований асинхронного клеточного автомата к блочно-синхронному виду эффективность распараллеливания достигает не более 70%, и применение графических ускорителей для повышения эффективности невозможно. При применении блочно-синхронных вариантов достигается высокая эффективность (\square 90%) на всех архитектурах, а применение графического ускорителя производительность увеличивает в 30-40 раз. При этом статистические исследования вносимой блочно-синхронным преобразованием погрешности для тестового примера показали ее практическое отсутствие, хотя известны такие асинхронные клеточные автоматы, для которых блочно-синхронные преобразования искажают их эволюцию. Возникла задача определения области применимости блочно-синхронных преобразований, которую пока не удалось решить.

Разработка условий корректности параллельных реализаций для сложных клеточно-автоматных моделей процессов самоорганизации. Получение оценок производительности и эффективности параллельных реализаций клеточных автоматов при разных режимах их функционирования.

Исследованы два вида клеточно-автоматных моделей процессов самоорганизации: 1) тоталистические клеточные автоматы со взвешенными шаблонами и 2) системы из взаимодействующих клеточных автоматов – параллельная композиция клеточных автоматов. Для первых определены свойства получаемых в результате их эволюции устойчивых структур и выявлен инвариант модели в виде отношения значений весов активаторных связей к ингибиторным (p/n). Доказана теорема о равенстве эволюций клеточных автоматов с одинаковыми значениями p/n . Показано, что для этого класса клеточных автоматов блочно-синхронные преобразования недопустимы. Исследования процессов самоорганизации, представленных системой взаимодействующих клеточных автоматов, производились на результатах моделирования динамики популяций некоторых организмов в оз. Байкал, где каждому виду организмов соответствует свой клеточный автомат. Исследована эффективность распараллеливания композиции в многоядерном компьютере с общей памятью путем размещения каждого автомата системы на своем ядре.

Разработка в системе имитационного моделирования параллельных мелкозернистых алгоритмов и структур WinALT средств сборки сложных моделей процессов с мелкозернистым параллелизмом из библиотечных элементов и средств их исполнения на параллельных ЭВМ (кластерах, графических картах).

Разработаны средства сборки сложных моделей процессов с мелкозернистым параллелизмом из библиотечных элементов. В языке системы имитационного моделирования параллельных мелкозернистых алгоритмов и структур WinALT

реализованы конструкции для подключения модулей из этих библиотек: директива препроцессора `include` для подключения фрагментов моделей и операторы `import` и `use` для загрузки внешних модулей из библиотеки функций. Реализованы модули системы моделирования WinALT, обеспечивающие исполнение моделей клеточных автоматов и их расширений на кластерах машин под ОС GNU Linux и на машинах с графическими картами NVIDIA. Параллельная реализация на кластере использует библиотеку MPI. Параллельная реализация для машин с графической картой основывается на библиотеке CUDA.