ПРОГРАММНЫЙ КОМПЛЕКС «ПИРАМИДА» ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С РАСПАРАЛЛЕЛИВАНИЕМ ПО ДАННЫМ

А.В. Баранов, А.В. Киселёв, Е.А. Киселёв, В.В. Корнеев, Д.В. Семёнов

В последние годы в развитии параллельных вычислений наметились две тенденции. Во-первых, увеличивается число пользователей, решающих прикладные задачи на параллельных вычислительных системах. При этом далеко не все пользователи обладают достаточными навыками программирования для эффективной организации параллельных вычислений. Часто возникает ситуация, когда необходимо осуществить распараллеливание по данным с отсутствием информационных обменов между процессами. Например, реализуя типовую схему «master-slave» средствами MPI, пользователь-программист тратит значительное время на организацию вычислений (порождение процессов и распределение данных) в MPI-программе.

Вторая тенденция – увеличение числа процессоров в параллельных вычислительных системах. При решении крупномасштабных задач, использующих большое число процессоров в течение длительного периода времени, на первый план выходит проблема обеспечения отказоустойчивости MPI-программы. Известно, что при отказе во время вычислений хотя бы одного процессора, велика вероятность аварийного завершения всей MPI-программы. Для обеспечения надежности вычислений пользователь должен предусмотреть периодическое сохранение контрольных точек и возможность рестарта программы, что требует дополнительных усилий при разработке программы.

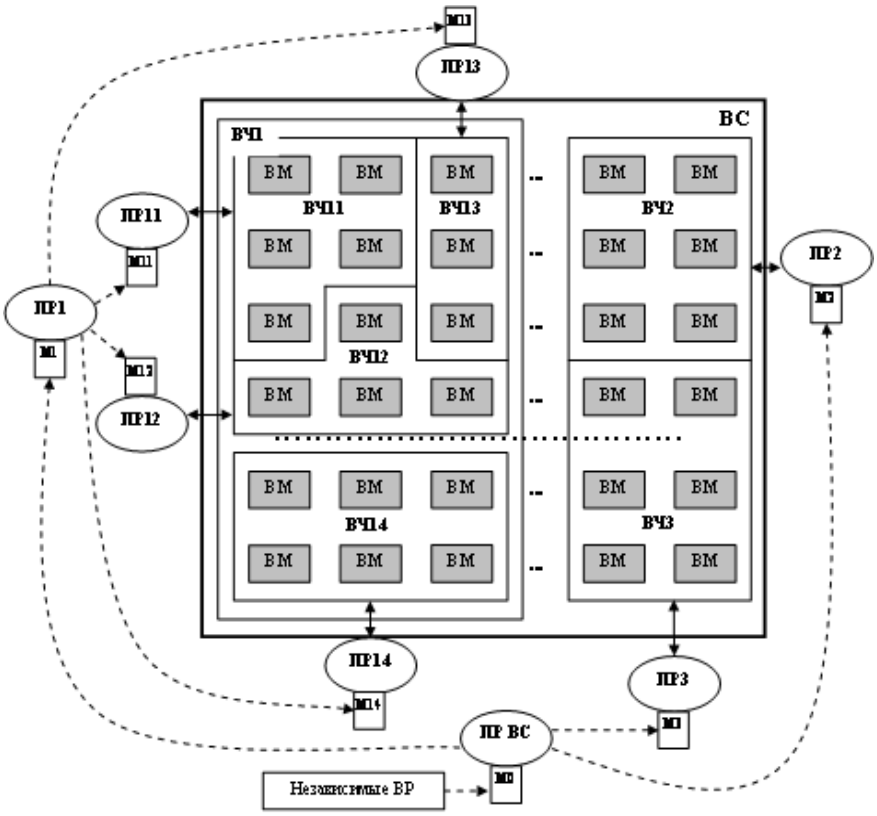

Рис. 1. Архитектура иерархической системы управления ресурсами и вычислениями в массово-параллельной ВС

Авторами разработан способ организации выполнения в массово-параллельной ВС независимых вычислительных работ (независимых заданий или параллельной программы, со схемой вычислений «распараллеливание по данным»), сокращающий время планирования вычислений и распределения вычислительных ресурсов, а также повышающий отказоустойчивость вычислений. В основу предложенного способа положен принцип распределенного управления в иерархически организованной системе вычислителей.

Сущность способа заключается в следующем (рис. 1). Пусть единицей распределяемого ресурса в некоторой ВС служит вычислительный модуль (ВМ). Множество ВМ рекурсивно разбивается на непересекающиеся подмножества, каждое из которых может рассматриваться, как некоторый вычислитель (ВЧ), и, в свою очередь, разбивается на непересекающиеся подмножества ВМ – вычислителей и т.д. Таким образом, формируется древовидная структура вычислителей. С каждым уровнем дерева (вычислителем) связывается

управляющий процесс – менеджер (М), в функции которого входит равномерная загрузка вычислителей данного уровня множеством независимых вычислительных работ. Каждый из вычислителей выбирает и обрабатывает некоторое подмножество вычислительных работ (пул работ – ПР), выделенных для данного уровня иерархии ВС. Суть обработки сводится к распределению работ между вычислителями следующего уровня (выполняет менеджер вычислителя) или к непосредственно выполнению вычислительных работ ВМ, в случае нижнего уровня иерархии дерева вычислителей. Обработка осуществляется до завершения всего пула работ ВС (ПР ВС).

Такой подход к организации управления вычислениями обеспечивает распределение управляющих функций между менеджерами ВС и позволяет конвейеризировать их выполнение, сокращая тем самым время на управление вычислительным процессом.

Отказоустойчивость вычислений в подобной ВС обеспечивается независимостью функционирования вычислителей на каждом уровне иерархии: в случае выхода из строя некоторого подмножества вычислителей одного уровня вычислительная работа, выделенная для данного уровня ВС, будет выполнена оставшимися вычислителями. Система остается работоспособной при деградации производительности. Другим преимуществом рассматриваемой ВС является простота сохранения состояний вычислений для рестарта в случае аварии системы в целом. Рестарт системы можно осуществить, сохраняя невыполненные вычислительные работы для отдельных уровней иерархии ВС. Количество уровней иерархии, для которых выполняется сохранение, определяет интервал «доката» системы до состояния, предшествующего аварии.

Рассмотренный способ лег в основу разработанного авторами программного комплекса «Пирамида». Комплекс предназначен для использования на массово-параллельных и грид системах с большим числом процессоров.

Комплекс «Пирамида» автоматизирует процесс организации параллельных вычислений при выполнении операций, задаваемых последовательной программой (ПП), над множеством экземпляров данных. Множество экземпляров данных определяет пул работ ВС. Допустимые значения (диапазоны значений) входных параметров задаются пользователем в паспорте задания, представляющем собой XML-файл. Таким образом, комплекс «Пирамида» обеспечивает выполнение ПП со всевозможными комбинациями значений параметров. Результаты выполнения ПП для каждого экземпляра данных сохраняются для последующей обработки или анализа.

Вычислительная система, работающая под управлением комплекса «Пирамида», логически организована в иерархическую структуру вычислителей, содержащих отдельные ВМ или вычислители нижнего уровня иерархии. Для каждого вычислителя назначается сервер управления (сервер вычислителя), а для вычислительной системы в целом – центральный сервер управления.

Управляющие процессы комплекса «Пирамида» – менеджеры – выполняются на центральном сервере (центральный менеджер – менеджер задания), серверах вычислителей (менеджер вычислителя) и вычислительных модулях (менеджер модуля – ММ). Менеджеры комплекса «Пирамида», взаимодействуя друг с другом, осуществляют запуски ПП на вычислительных модулях системы со всеми допустимыми экземплярами данных, формируемыми на основе заданных пользователем значений параметров. Иерархическая структура системы менеджеров комплекса «Пирамида» приведена на рисунке 2.

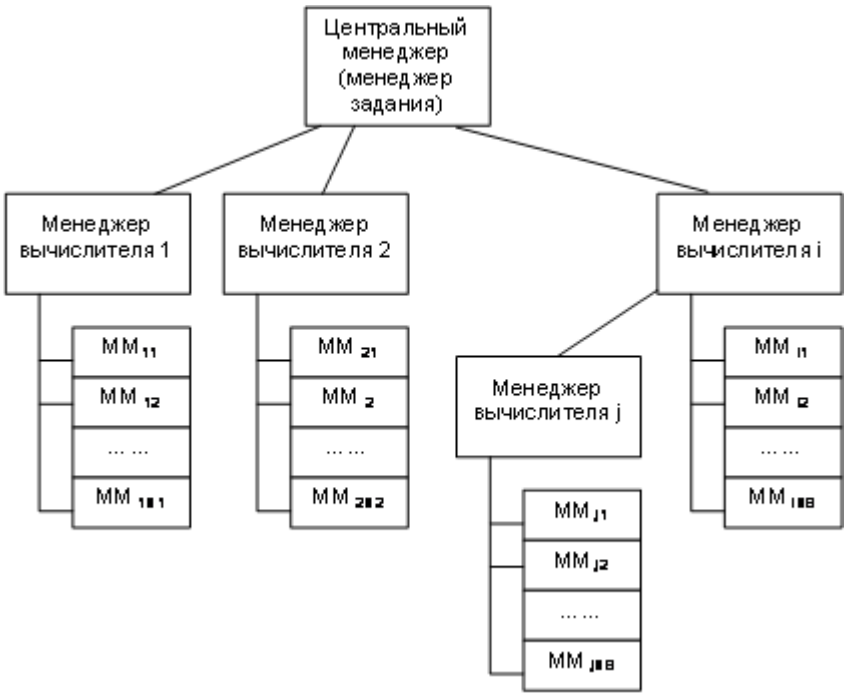

Рисунок 2 – Иерархическая структура системы менеджеров комплекса «Пирамида»

Единицей вычислительной работы в комплексе «Пирамида» является так называемый слайс, представляющий собой подмножество экземпляров данных, которое ПП обрабатывает за один цикл выполнения. Под размером пула работ понимается число слайсов, которое он содержит. Основой логики распределения работ в комплексе «Пирамида» является функция деления произвольного пула работ на более мелкие пулы работ произвольного размера.

Прикладная логика реализуется в пользовательской ПП, а менеджеры комплекса «Пирамида» решают задачи управления вычислительным процессом. К этим задачам следует отнести:

- запуск множества экземпляров ПП на всех доступных ВМ вычислительной системы;
- формирование подмножеств экземпляров данных (слайсов вычислительной работы) для распределения между экземплярами ПП;
- передача слайсов вычислительной работы экземплярам ПП и получение результатов вычислений;
- сбор результатов вычислений на центральный сервер (комплекс «Пирамида» способен осуществлять сбор результатов как при наличии, так и при отсутствии общей сетевой файловой системы для ВМ ВС);
- мониторинг состояния вычислительных ресурсов;
- перераспределение вычислительных работ между ВМ в случае выхода из строя некоторого множества ВМ или вычислителей ВС;
- автоматическое сохранение состояния вычислений в файле контрольных точек;
- восстановление состояния вычислений после перезапуска системы и продолжение решения задачи.

Инициализация работы программного комплекса «Пирамида» производится путем запуска менеджера задания на центральном сервере управления. На вход менеджера задания подается описание доступных вычислительных ресурсов ВС, описание конфигурации системы и XML-паспорт задания. Паспорт задания содержит сведения о ПП, входных данных и местоположении будущих результатов выполнения экземпляров ПП. На основании информации паспорта менеджер задания формирует пул работ ВС.

Далее менеджер задания порождает менеджеров вычислителей, передавая им на вход описание подчиненных ресурсов и паспорт задания. Менеджеры вычислителей, в свою очередь, порождают менеджеров подчиненных вычислителей или менеджеров вычислительных модулей.

Функционирование менеджеров комплекса «Пирамида» после инициализации определяется следующим образом. С помощью функции деления менеджер задания отделяет от пула работ ВС порции (пулы) вычислительной работы определенного размера. Отделенные порции передаются менеджерам вычислителей.

Менеджеры вычислителей производят дальнейшее деление полученной работы на порции (пулы) меньшего размера и передают отделенные порции своим менеджерам ВМ. Менеджеры ВМ делят каждый свой пул работ на отдельные слайсы и производят запуск ПП с отделенными слайсами в качестве входных данных.

Комплекс «Пирамида» обеспечивает отказоустойчивость в гетерогенной и ненадежной вычислительной среде. Алгоритмы работы менеджеров всех уровней рассчитаны на то, что подчиненные менеджеры могут обладать разной производительностью или выйти из строя в любой момент времени. Для обеспечения надежности вычислений используется следующий подход.

Получив на исполнение пул вычислительных работ, менеджер распределяет его между подчиненными менеджерами и ожидает от них результата выполнения работ. При получении результата подчиненному менеджеру выделяется и передается новая порция вычислительной работы. Процесс повторяется до тех пор, пока исходный пул работ не будет полностью исчерпан.

Пусть менеджер вычислителя распределяет пул работ для N подчиненных менеджеров. В некоторый момент возникает ситуация, когда некоторый подчиненный менеджер i ($1 \le i \le N$), выполнив очередную порцию работы, обнаруживает, что исходный пул работ исчерпан. В этом случае менеджер i получает от менеджера вычислителя порцию работы одного из оставшихся N-1 менеджеров, не закончивших к этому моменту расчет своих порций. Таким образом обеспечивается надежность вычислений – если какой-либо подчиненный менеджер вышел из строя, его порция вычислительной работы будет гарантированно выполнена другим менеджером.

При такой организации надежность и масштабируемость достигаются автоматически. Если какой-либо вычислительный модуль выходит из строя, его менеджер перестает передавать ему вычислительную работу. В этом случае вычисления продолжаются другими ВМ с некоторым снижением общей производительности.

Менеджеры комплекса «Пирамида» имеют схожую структуру, которую мы рассмотрим на примере менеджера вычислителя (рис. 3).

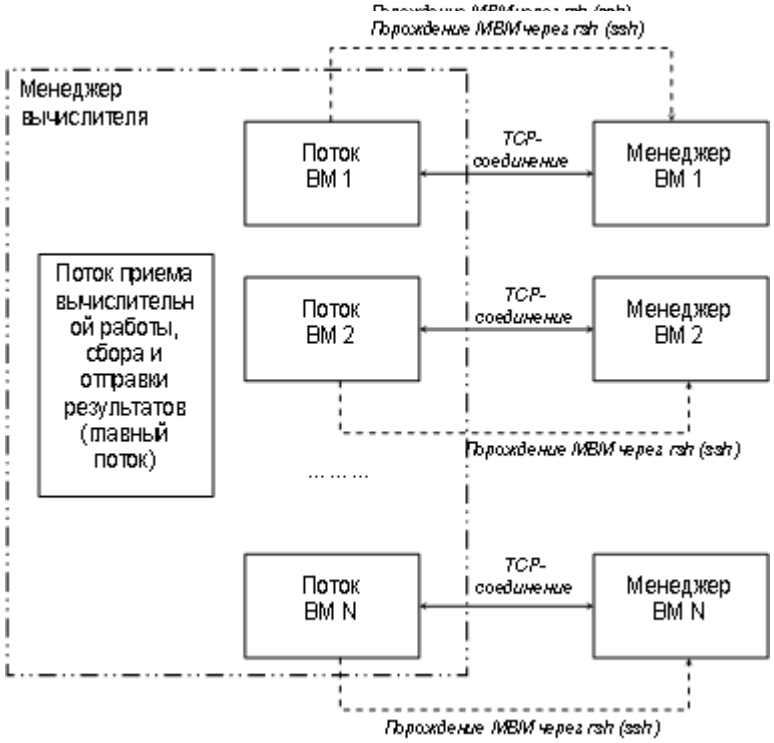

Рис. 3. Структура менеджера вычислителя и взаимодействие с менеджерами ВМ

Менеджер вычислителя реализован как многопоточное приложение. Главный поток осуществляет прием вычислительной работы от центрального менеджера, сбор и отправку результатов. По одному потоку выделяется на каждый подчиненный менеджер ВМ или вычислителя. Подчиненный менеджер создается и контролируется отдельным процессом удаленного выполнения команд (rsh или ssh в зависимости от конфигурации). Подчиненный менеджер выступает по отношению к породившему его потоку в роли сервера. После порождения подчиненного менеджера соответствующий поток менеджера кластера пытается соединиться с ним в режиме клиента. При успешном соединении поток менеджера кластера начинает передавать подчиненному менеджеру порции вычислительной работы и получать в ответ результаты. Заметим, что использование в качестве сервера именно подчиненного менеджера, а не менеджера вычислителя, позволяет избежать проблемы «узкого горла», когда большое число клиентов (подчиненных менеджеров) с различных ВМ одновременно пытаются соединиться с единственным сервером (менеджером вычислителя).

Целостность установленного соединения непрерывно контролируется. При разрыве соединения соответствующий поток ВМ менеджера вычислителя предпринимает меры по его восстановлению. При невозможности восстановления предпринимается попытка перезапуска подчиненного менеджера через процесс удаленного выполнения команд (rsh или ssh).

В случае выхода из строя произвольного числа вычислительных модулей или вычислителей выполнение задания не прерывается, и полученные ранее результаты не теряются. При восстановлении работоспособности вычислительных модулей производится их автоматическое подключение к процессу вычислений. Кроме этого, на центральном сервере осуществляется периодическое сохранение контрольных точек, что позволяет прервать и возобновить вычисления в произвольный момент времени.

Для совместимости программного комплекса «Пирамида» с различными программно-аппаратными платформами авторы использовали при разработке библиотеку Qt4, обеспечивающую переносимость приложений на уровне исходных кодов между различными ОС (Linux, MS Windows, Mac OS). Это позволяет использовать комплекс «Пирамида» в ВС с однородными и неоднородными вычислителями с разными ОС.

Опыт практического применения разработанного авторами комплекса в массово-параллельной ВС показал его устойчивое и эффективное функционирование при решении крупномасштабных вычислительных задач.