

Секция X
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И МОДЕЛИРОВАНИЕ

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ
ОРГАНИЗАЦИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ**

И. Л. Стефановский

*Гомельский государственный технический университет
имени П. О. Сухого, Беларусь*

Научный руководитель К. С. Курочка

При решении широкого круга практических задач необходимо выполнять большой объем вычислений. Суперкомпьютеры, традиционно применявшиеся для решения этих задач, очень дороги, поэтому в последнее время все большее применение находят кластерные и GRID-системы, построенные из дешевых рабочих станций и объединенные стандартным сетевым оборудованием.

Применение подобных систем выдвигает ряд определенных требований к программному обеспечению, которое должно решать следующие задачи:

1. Распределять вычислительную нагрузку.
2. Организовать информационное взаимодействие (передачу данных) между процессорами, что приводит к усложнению ПО.

Наиболее популярные технологии для организации распределенных вычислений: PVM, MPI, CORBA и технология клиент-серверных СУБД.

MPI – это стандарт взаимодействия параллельных процессов с помощью механизма передачи сообщений.

Возможности MPI:

Во-первых, MPI поддерживает несколько режимов передачи данных, важнейшими из которых являются: синхронная передача, которая не требует выделения промежуточных буферов для данных и обеспечивает надежную передачу данных сколь угодно большого размера, и асинхронная передача, при которой посылающий сообщение процесс не ждет начала приема, что позволяет эффективно передавать короткие сообщения. Во-вторых, MPI позволяет передавать данные не только от одного процесса к другому, но и поддерживает коллективные операции: широковещательную передачу (broadcasting), разборку-сборку (scatter и gather), операции редукции. В-третьих, MPI предусматривает гетерогенные вычисления. Вычислительная система может включать разные процессоры, в том числе, имеющие различные наборы команд и разное представление данных. В-четвертых, программа, использующая MPI может быть написана на разных языках (C/C++, Fortran, C# и т. д.). Синтаксис MPI облегчает создание приложений в модели SPMD (single program multiple data) – одна программа работает в разных процессах со своими данными. Одна и та же функция вызывается на узле-источнике и узлах-приемниках, а тип выполняемой операции (передача или прием) определяется с помощью параметра. Такой синтаксис вызовов делает SPMD-программы существенно компактнее.

Основная особенность стандарта MPI – понятие коммуникатора. Все операции синхронизации и передачи сообщений локализуются внутри коммуникатора. С коммуникатором связывается группа процессов. В частности, все коллективные операции вызываются одновременно на всех процессах, входящих в эту группу. Поскольку взаимодействие между процессами инкапсулируется внутри коммуникатора, на базе MPI можно создавать библиотеки параллельных программ.

Архитектура системы на основе технологии MPI представлена на рис. 1.

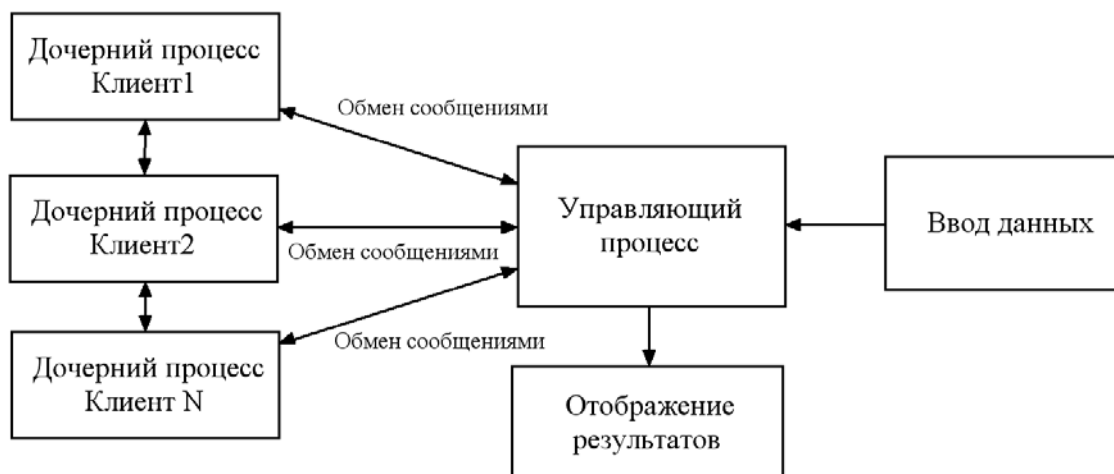


Рис. 1. Архитектура системы на основе технологии MPI

Системы, построенные на основе технологии клиент-серверных СУБД состоят из следующих частей:

1. Планировщика задач.
2. Базы данных (БД) задач.
3. Клиентов.

Архитектура системы на основе технологии клиент-серверных СУБД представлена на рис. 2.

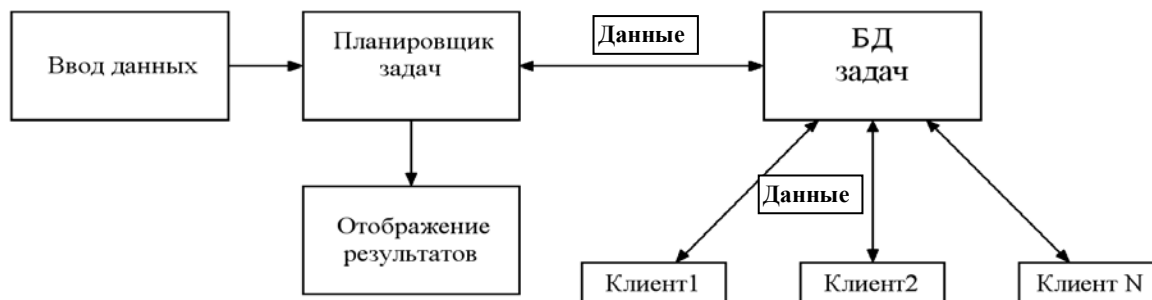


Рис. 2. Архитектура системы на основе технологии клиент-серверных СУБД

Планировщик задач служит для ввода исходных данных, необходимых для решения задачи, разделения задачи на составные части, в зависимости от количества клиентов, размещения этих частей и других необходимых данных в базе данных задач, получения результатов обработки данных и их объединение в единое целое.

База данных задач служит для хранения задач для клиентов, включая исходные данные и параметры обработки данных, а также для хранения результатов обработки.

Клиенты выполняют непосредственно процесс обработки данных, полученных из БД задач.

Данные технологии использовались при разработке ПО, осуществляющего обработку изображений с использованием алгоритма *ttest*, который служит для подавления шума и «несущественных» деталей в цифровых изображениях. В качестве реализации технологии MPI был выбран пакет *Mpich*, работающий на ОС семейства Windows. При реализации на базе технологии клиент-серверных СУБД в качестве СУБД использовалась MS SQL Server 2000. Вычислительные части были написаны на C++. Измерения проводились на изображении размером 3000x3074 точек. Маска фильтра была выбрана равной трем. Данные, полученные в результате измерений, представлены в виде графика на рис. 3.

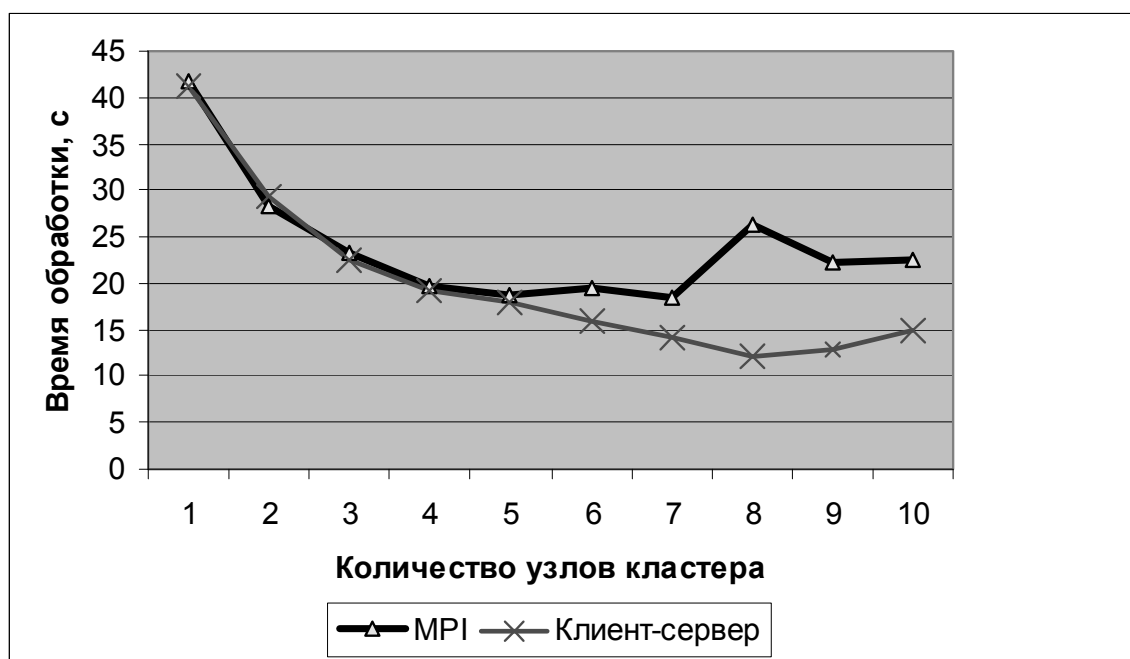


Рис. 3. Зависимость времени обработки от количества узлов кластера при размере маски, равной 3

Из графика хорошо видно, что время обработки, как правило, уменьшается при увеличении количества узлов кластера. Тем не менее, при определенном количестве узлов время обработки перестает уменьшаться, а в дальнейшем может даже увеличиться. Такая ситуация называется насыщением. Это объясняется увеличением накладных расходов, связанных с пересылкой информации по сети, по отношению к времени вычислений. При использовании технологии клиент-серверных СУБД насыщение происходит позднее. Таким образом, можно сделать выводы:

1. Технология MPI обладает большей чувствительностью к скорости передачи данных по сети, чем технология клиент-серверных СУБД, в связи с чем MPI можно рекомендовать использовать для программ, сильно загружающих процессор и передающих небольшой объем данных по сети.

390 Секция X. Информационные технологии и моделирование

2. Системы, построенные на основе технологии клиент-серверных СУБД, обладают более простой реализацией, чем системы, построенные с использованием МРІ, сложность которых связана с самой парадигмой передачи сообщений.

3. При построении приложений на основе технологии клиент-серверных СУБД необходимо вручную реализовывать сервисные возможности, уже реализованные в МРІ.