

РАЗРАБОТКА СИСТЕМЫ ПРОЕКТИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ СТАНДАРТНЫХ ИНТЕРФЕЙСОВ

А.И. Прокопчик

Введение

Распределенная обработка на кластерных архитектурах обеспечивает высокоскоростную среду для решения фундаментальных научных и инженерных задач. Главная проблема – разработка распределенных приложений для таких сред. Использование коммуникационных библиотек при проектировании является трудоемкой задачей. В работе предложена высокоуровневая среда проектирования распределенных вычислений на основе секционной модели параллельных вычислений.

1. Модель кластерной архитектуры

Предлагается структура кластерной архитектуры на рис. 1.

Модель имеет трехуровневую структуру:

1. Физический уровень. Данный уровень выполняет функции физического соединения компьютеров – это физическая часть локальной вычислительной сети.

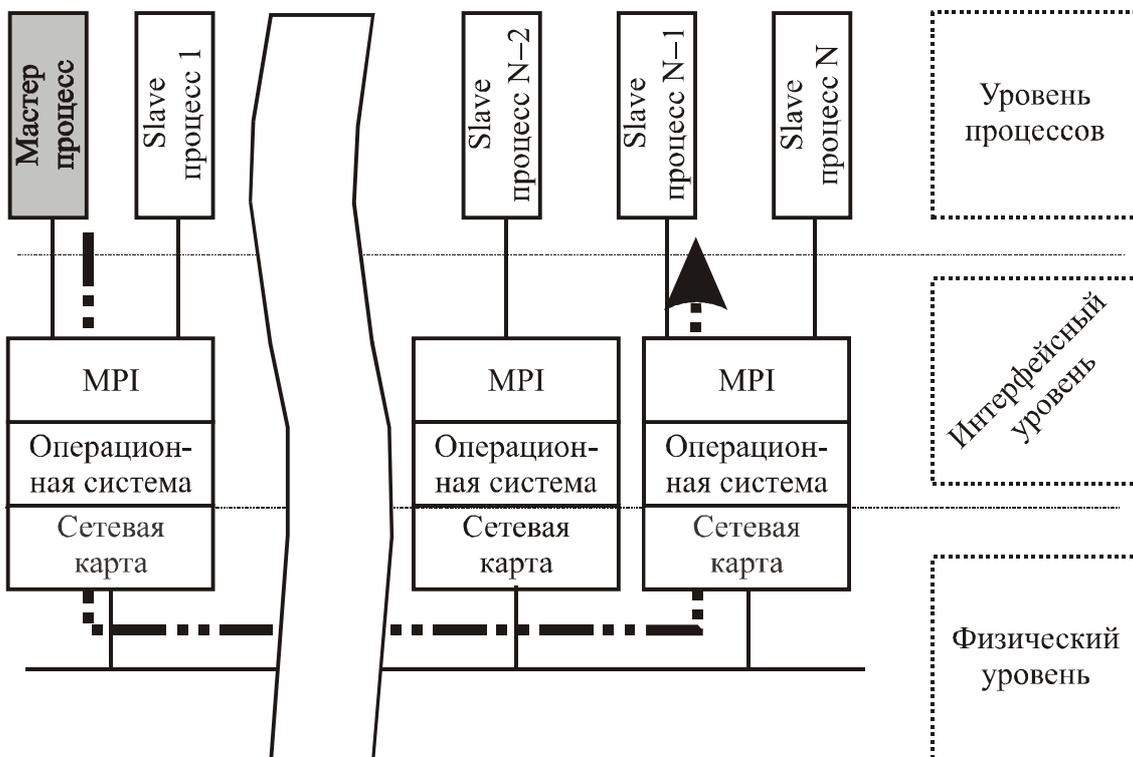


Рис. 1. Структура кластерной архитектуры.

2. Интерфейсный уровень включает программную поддержку обмена информацией между компьютерами по физической среде (операционная система) и между процессами (MPI) [1].

3. Уровень процессов – это уровень взаимодействия между процессами в плоском пространстве (интерфейсная поддержка MPI).

2. Секционная модель

Пусть $P = \{p_i, i=1..n\}$ – множество процессов, реализующих вычисления некоторой задачи. Секция есть двойка $S=(a, \pi)$, где a – условие инициализации секции (УИ), π – тело секции, которое представляет собой набор программ, которые секцией ставятся в соответствие одному или нескольким процессам из P .

Параллельная программа представляется как множество секций с определенными УИ.

Пусть $S = \{s_i, i=1, \dots, m\}$ в общем случае множество иерархически вложенных секций. Состояние множества секций S в момент времени t может быть специфицировано как

$$Q_t(S) = \{(u_i, v_i), i=1, \dots, m, t \geq 0\},$$

где u_i и v_i соответственно общее число инициализаций и завершений секции s_i в момент времени t , начиная с момента начала работы всего множества секций S при $t=0$.

Каждая инициализация и каждое завершение секции вызывает изменение состояния Q_t . Выполнение параллельной программы обеспечивается монитором на основе аксиом [2].

3. Реализация

Основу системы составляет монитор, общая схема которого представлена на рис.2.

Монитор выполняет следующие функции:

- проверку условий инициализации секций и в случае необходимости исполнение последних;
- обработку завершений выполнения секций;
- поддержку работы с глобальными (управляющими) переменными;
- предоставление статистической информации работы секций (временные характеристики).

4. Эффективность модели

Эффективность реализации данной модели отражается полученными результатами выполнения теста: подсчета числа π в таблице.

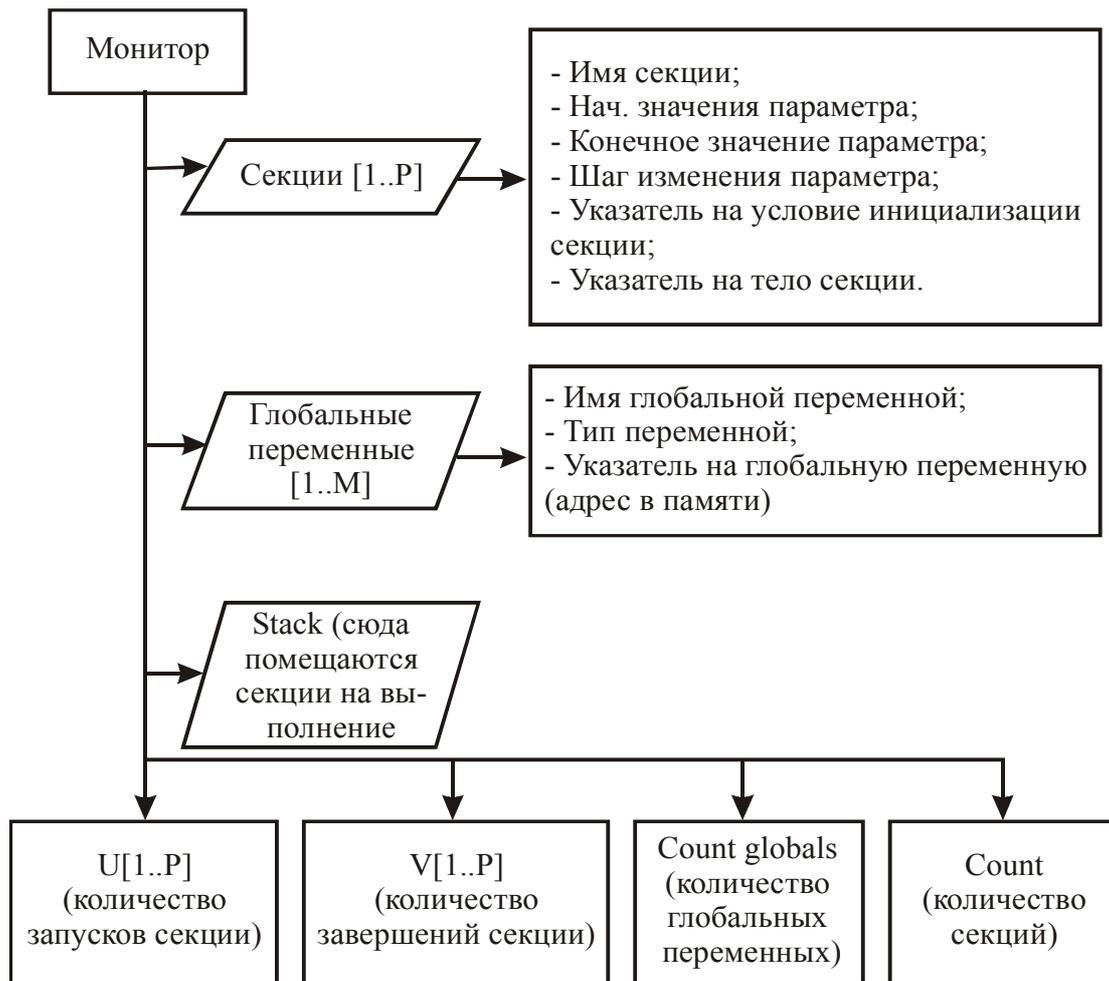


Рис. 2. Схема монитора

Для анализа эффективности работы вычислительной системы используется коэффициент увеличения производительности, определяемый как

$$\sigma = \frac{\text{время вычисления на одном компьютере}}{\text{время вычисления на кластере}}$$

Таблица

Результаты выполнения теста – подсчет числа π

Точность вычислений $e^{-N \ln(10)}, N$	Время вычисления на кластере, с	Время вычисления на 1-м компьютере (Celeron 500), с	σ
12	0,22	0,34	1,55
14	1,85	3,26	1,76
15	5,69	10,29	1,80
17	56,01	104,28	1,86
19	559,66	1046,56	1,87

Вывод. Как видно из полученных результатов, производительность при использовании кластера возрастает, максимальный прирост производительности составляет 87 % (при использовании 3-х компьютеров в кластере). Узкими местами в построенном кластере являются неоднородность последнего (разные компьютеры в кластере) и среда передачи данных (в данном тесте среда передачи данных не играет весомой роли). При высоком порядке точности время, необходимое для передачи данных, мало по сравнению с временем вычислений, поэтому с ростом порядка точности вычислений вырисовывается тенденция к росту производительности. Но неоднородность кластера на этом этапе является серьезным барьером (вследствие чего при тестировании не удалось получить прирост более 87 %). Обойти данный барьер можно либо грамотным перераспределением процессов между компьютерами кластера и перепроектированием программы в рамках данной модели, либо модернизацией кластера в целом.

Для увеличения эффективности данной модели необходимо модернизировать кластер: использовать более скоростную технологию передачи данных и использовать однородные компьютеры (однородное аппаратное обеспечение) для построения кластера.

Литература

1. Message Passing Interface Forum MPI: A Message Passing Interface Standard // International Journal of Supercomputer Applications 8 (3/4). 1994.
2. Буза М. К., Зимянин Л. Ф. Секционная модель параллельных вычислений // Программирование. 1990. №4. С. 54–62.

ОПТИМАЛЬНЫЕ ПРОГРАММНЫЕ И ТИПА ОБРАТНОЙ СВЯЗИ ОДНОМЕРНЫЕ УПРАВЛЕНИЯ ДЛЯ ДИНАМИЧЕСКИХ СИСТЕМ

О. В. Ридченко

Линейные задачи оптимального управления (ОУ) подробно изучены в литературе [1]. Вместе с тем до сих пор, несмотря на многочисленные предложения, нельзя указать достаточно эффективные численные методы построения их решений, в которых полно учитывалась бы специфика этих задач. Особенно это касается позиционных решений.

В работе реализован алгоритм нахождения оптимального программного и позиционного решения линейной задачи оптимального управления с учетом ее динамической специфики.