

Анализ рис. 4 показал, что область максимальных напряжений располагается вблизи оси симметрии оправки, примерно на расстоянии 0,4 м от свободного торца оправки. Максимальные напряжения достигают значения 497,41 МПа.

Анализ рис. 5 показал, что перемещения вдоль оси ОУ не превышают значений $1,399 \times 10^{-3}$ м. В заделке перемещения равны нулю.

Полученные перемещения и напряжения меньше допустимых, а масса является минимальной.

П.Н. Полежаев

Оренбургский государственный университет

**АЛГОРИТМЫ ПЛАНИРОВАНИЯ ЗАДАЧ
ДЛЯ ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА С УЧЕТОМ СЕТИ
И МНОГОПРОЦЕССОРНОСТИ УЗЛОВ**

Существующие алгоритмы планирования задач для кластерных вычислительных систем, используемые в современных управляющих системах, демонстрируют неплохую эффективность работы. Однако имеется возможность дальнейшего роста их производительности за счет учета топологии вычислительной системы и многопроцессорности узлов.

Действительно, если алгоритм планирования будет назначать процессы параллельной задачи на топологически близкие вычислительные ядра, то это приведет к снижению времени ее исполнения в силу сокращения коммуникационных задержек при передаче данных между ее процессами. Это, в свою очередь, увеличивает производительность всей вычислительной системы.

Учет многопроцессорности вычислительных узлов при планировании также положительно сказывается на показателях работы кластерной системы, так как время выполнения коммуникационных операций между процессами параллельной программы, исполняющимися на соседних процессорах (или ядрах) одного узла, гораздо меньше, чем между процессами, исполняющимися на разных узлах.

Кроме того, при назначении параллельных программ на свободные вычислительные ядра необходимо учитывать сетевую конкуренцию между процессами одновременно исполняющихся задач. Ее снижение приводит к уменьшению времени выполнения сетевых коммуникаций, а это ведет к росту производительности кластерной системы.

Настоящее исследование направлено на разработку эффективных алгоритмов планирования задач для высокопроизводительных кластерных систем, учитывающих топологию, сетевую конкуренцию и многопроцессорность вычислительных узлов.

Модель вычислительного кластера. Разработанная модель вычислительного кластера [1] позволяет формировать аппаратную конфигурацию узлов вычислительной системы и соединяющей их высокопроизводительной сети с помощью взвешенного ориентированного графа. Множество вершин данного орграфа помимо вычислительных узлов также включает сетевые коммутаторы. Веса вершин-узлов представляют собой их статические параметры (количество вычислительных ядер, их производительность, объемы оперативной и дисковой памяти), а веса дуг – скорости передачи данных по соответствующим сетевым связям.

С помощью данной модели можно описать вычислительную систему произвольной топологии. В частности, в рамках данной работы она используется для задания вычислительных кластеров топологии толстого дерева, двумерного тора и звезды с однородными и неоднородными узлами.

Исследуемые алгоритмы планирования задач на вычислительном кластере. В рамках проводимого исследования были рассмотрены алгоритмы планирования задач, представляющие собой сочетания алгоритма выбора следующей задачи из очереди Most Processors First Served Scan (MPFS Scan) или алгоритма обратного заполнения Backfill с методами ее назначения на вычислительные ядра, учитывающими топологию вычислительной системы [1]: для топологии толстого дерева – Sorting Nodes by Performance (SNP), Fat Tree Sorting Commutators by Performance (FTSCP), Fat Tree Sorting Commutators by

Cores (FTSC); для топологии двумерного тора – модифицированные варианты алгоритмов Minimizing message-passing Contention 1x1 (MC1x1) и Minimizing message-passing Contention 1x1 Incremental (MC1x1+Inc); для топологии звезды – Sorting Nodes by Performance (SNP) и Sorting Nodes by Speed (SNS).

Также рассматривались сочетания алгоритмов MPFS Scan и Backfill с методами назначения, не принимающими топологию во внимание [2]: First Fit (FF), Best Fit (BF), Fastest Node First (FNF) и Random First (RF).

В ходе исследования были предложены собственные алгоритмы планирования, представляющие собой сочетания алгоритма MPFS Scan или Backfill с разработанными методами назначения Summed Distance Minimization (SDM) и Maximum Distance Minimization (MDM). Данные методы учитывают топологию вычислительной системы, сетевую конкуренцию между одновременно исполняющимися задачами и многопроцессорность вычислительных узлов.

Пусть методу назначения передается выбранная из очереди задача J_j , требующая для своего исполнения n_j вычислительных ядер. Далее опишем принцип работы алгоритмов SDM и MDM.

Метод назначения SDM для каждого допустимого окна планирования $W \in WL'$ перебирает все сетевые устройства d кластера (коммутаторы и узлы) и определяет для каждого из них n_j ближайших вычислительных ядер окна W , подходящих по конфигурации для задачи J_j . Эти вычислительные ядра формируют возможное окно запуска $R_{W,d}$ задачи J_j . Для назначения в качестве результата R выбирается окно $R_{W,d}$ с минимальным суммарным попарным расстоянием, если таких несколько, то выбирается окно с большей суммарной производительностью вычислительных ядер.

Данный метод является обобщением алгоритма Manhattan Median для случая произвольных топологий. За счет компактного размещения процессов параллельной задачи алгоритм SDM пыта-

ется минимизировать суммарное попарное расстояние между выделенными ей вычислительными ядрами. Это позволяет снизить сетевую конкуренцию между одновременно выполняющимися в системе задачами и уменьшить степень распределения процессов параллельных задач по вычислительной системе.

Метод назначения MDM для каждого допустимого окна планирования $W \in WL'$ перебирает все сетевые устройства d кластера и для каждого из них запускает алгоритм поиска в ширину. В процессе его работы формируется множество достигнутых вычислительных ядер узлов окна W , которые подходят по конфигурации для задачи J_j , до тех пор, пока не будет получено n_j ядер. При этом среди ядер, находящихся на одинаковом расстоянии от начального сетевого устройства в первую очередь выбираются те, которые имеют большую скорость. Результатом работы поиска в ширину является сформированное возможное окно запуска $R_{W,d}$. Для назначения задаче J_j в качестве результата R выбирается окно $R_{W,d}$ с минимальным максимальным расстоянием от первоначального сетевого устройства d .

Данный метод является аналогом алгоритма MC1x1 для произвольных топологий. За счет компактного размещения процессов параллельной задачи алгоритм MDM пытается минимизировать максимальное расстояние от выбранного центрального сетевого устройства до назначенных задаче вычислительных ядер.

Симулятор вычислительного кластера и его управляющей системы. Для экспериментального сравнительного исследования эффективности работы предложенных методов назначения с существующими вариантами использовался созданный в рамках настоящей работы симулятор вычислительного кластера и его управляющей системы.

Имитационная схема его работы представлена на рис. 1. Источник I генерирует поток параллельных задач, отправляемых пользователями в очередь Q управляющего узла p_0 . Канал S представляет собой планировщик, который в соответствии с заложенным в него алгоритмом осуществляет извлечение

задач из Q и назначение их на свободные вычислительные ядра подходящих по конфигурации узлов.

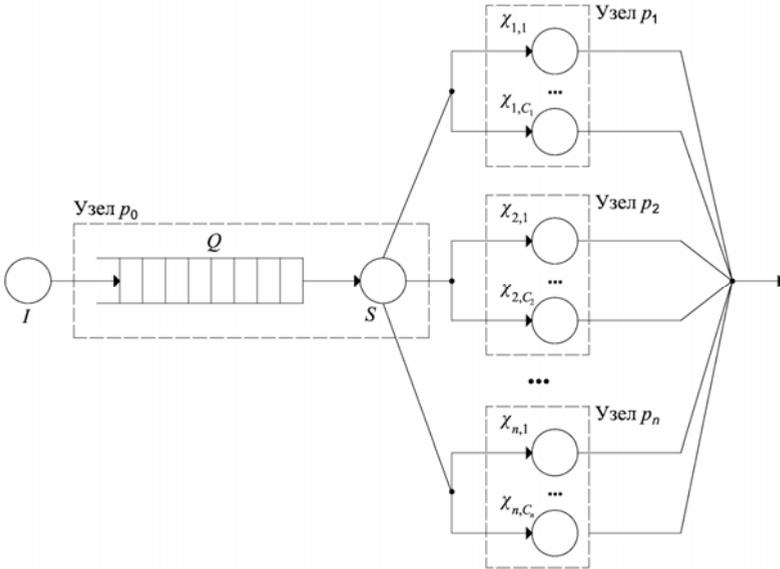


Рис. 1. Имитационная схема управляющей системы вычислительного кластера

Схема работы вычислительного узла p_i приведена на рис. 2, p_i соединен с другими узлами и коммутаторами вычислительной сети с помощью r_i дуплексных связей. Все входящие пакеты, а также пакеты сообщений, генерируемых процессами, выполняющимися на локальных вычислительных ядрах, сначала поступают в очередь $Q_{inp,i}$, а затем маршрутизируются каналом обслуживания R_i . Если соответствующий пакет предназначен для локального вычислительного ядра, то он передается ему непосредственно, иначе – помещается в одну из очередей $Q_{out,i,1}, Q_{out,i,2}, \dots, Q_{out,i,r_i}$, соответствующую выбранной алгоритмом маршрутизации исходящей связи. При выполнении маршрутизации каналом R_i моделируется временная задержка величиной $I_{routing}$. Имитационная схема коммутатора представ-

ляет собой урезанный вариант имитационной схемы узла. Отличие только в том, что коммутатор не может выполнять вычисления, и поэтому у него отсутствуют вычислительные ядра.

На рис. 2. также приведена схема работы дуплексной связи $L_{ij} = \{E_{ij}, E_{ji}\}$, соединяющей сетевые устройства $d_i \in D$ и $d_j \in D$. Каналы обслуживания E_{ij} и E_{ji} добавляют к времени передачи пакета задержку величиной $b(E_{ij}) = b(E_{ji})$.

В качестве алгоритма работы симулятора используется алгоритм моделирования по принципу особых состояний (событий).

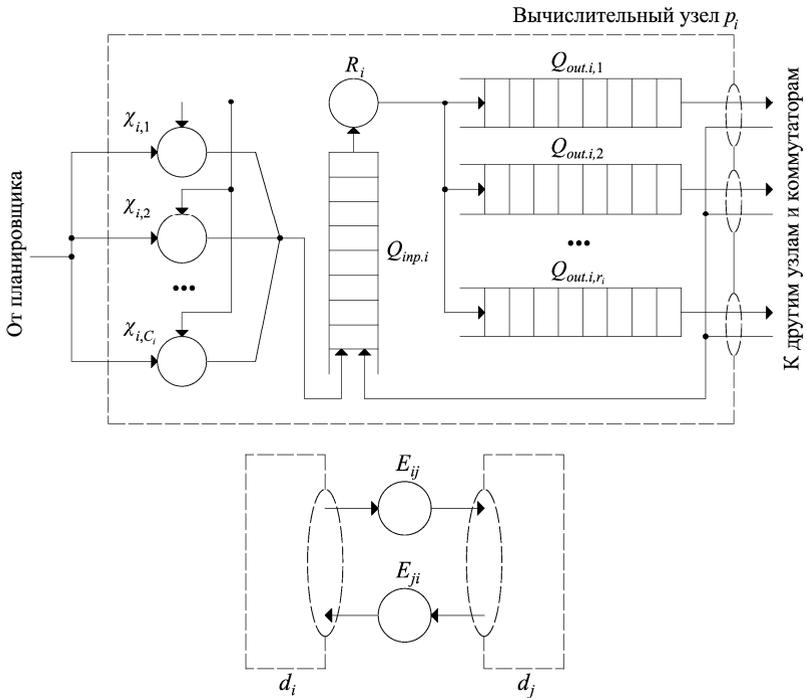


Рис. 2. Имитационная схема узла p_i и дуплексной сетевой связи L_{ij}

Разработанная в настоящем исследовании модель загрузки вычислительной системы потоком задач использует для описания параметров генерируемых задач случайные величины с за-

данными законами распределения, а также определяет SPMD-структуру задачи в виде цикла, на каждой итерации которого происходит выполнение фаз коммуникации и вычислений. Данная модель позволяет генерировать типичные потоки задач, поступающие в очереди управляющих систем большинства современных вычислительных кластеров.

Результаты экспериментального исследования. Исследование проводилось для двух групп сценариев: вычислительный кластер с однородными узлами и однородной высокопроизводительной сетью и вычислительный кластер с неоднородными узлами и однородной сетью.

В первом случае моделировался кластер из 12 узлов, каждый из которых имеет 2 вычислительных ядра, 2 Гб оперативной памяти, 40 Гб локальной дисковой памяти, единичную относительную вычислительную скорость. Во втором – использовалась конфигурация из 12 узлов с суммарным количеством вычислительных ядер 24, разными объемами оперативной и дисковой памяти, разной относительной вычислительной скоростью.

Каждая группа сценариев рассматривалась для трех топологий: толстое дерево (два 6-портовых корневых коммутатора и три 8-портовых листовых коммутатора (4 восходящих порта и 4 узловых порта на каждом)), звезда (один 12-портовый коммутатор), двумерный тор размером 3×4 узла.

Оценка эффективности работы алгоритмов планирования осуществлялась с помощью системы количественных критериев и метрик, которая охватывает все аспекты состояния вычислительной системы: производительность, сбалансированность загрузки вычислительной системы, используемость памяти вычислительных узлов, гарантированность обслуживания задач, честность по отношению к задачам, характер распределения процессов параллельных задач по вычислительной системе и сетевой конкуренции между ними.

В частности, критерий производительности включает следующие метрики: средняя загрузка вычислительных ядер узлов кластера U_{cores} , потеря производительности кластера CL , максимальное время завершения задачи C_{max} , среднее время

ожидания задач в очереди \bar{T}_{wait} и среднее ограниченное замедление задач S_{lim} .

Для получения достоверных результатов симуляция проводилась в каждом сценарии 100 раз для различных потоков из 500 случайно сгенерированных задач. Все вычисляемые значения метрик усреднялись по всем потокам.

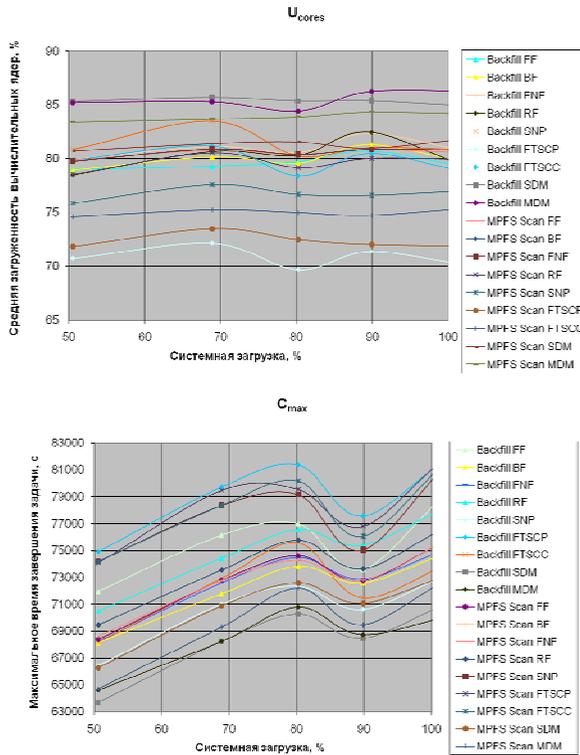


Рис. 3. Графики зависимости значений метрик производительности U_{cores} и C_{max} от величины системной загрузки L

Сравнение алгоритмов планирования осуществлялось путем построения графиков зависимости метрик от величины системной загрузки L . Например, на рис. 3 изображены графики зависимости метрик производительности U_{cores} и C_{max} от L для сценария кла-

стера топологии толстого дерева с неоднородными вычислительными узлами. Заметим, что алгоритмы планирования в первую очередь сравнивались по критерию производительности, остальные критерии носили вторичный характер.

Алгоритмы планирования, показавшие лучшие результаты в каждом сценарии

Группа сценариев	Топология	Наилучшие алгоритмы
Неоднородные вычислительные узлы и однородная сеть	Толстое дерево	Backfill SDM (при $L < 85-93$ %), Backfill MDM (при $L \geq 85-93$ %)
	Двумерный тор	Backfill SDM (при $L < 67-71$ %), Backfill MDM (при $L \geq 67-71$ %)
	Звезда	Backfill SDM, Backfill MDM
Однородные вычислительные узлы и однородная сеть	Толстое дерево	Backfill SDM (при $L < 75-79$ %), Backfill MDM (при $L \geq 75-79$ %)
	Двумерный тор	Backfill SDM (при $71-73$ %), Backfill MDM (при $L \geq 71-73$ %)
	Звезда	Backfill SDM, Backfill MDM, Backfill SNP

Полученные результаты экспериментального исследования сведены в таблицу. Для каждого сценарии может быть рекомендован свой алгоритм планирования, представляющий собой сочетания алгоритма выбора следующей задачи из очереди Backfill с разработанными методами ее назначения на вычислительные ядра MDM или SDM.

Список литературы

1. Гергель В.П., Полежаев П.Н. Теоретические основы экспериментального исследования алгоритмов планирования задач для вычислительного кластера с помощью симулятора // Вестн. Оренбург. гос. ун-та. 2010. – № 9(115). – С. 114–120.

2. Полежаев П.Н. Исследование алгоритмов планирования параллельных задач для кластерных вычислительных систем с помощью симулятора // Параллельные вычислительные технологии (ПАВТ–2010): тр. междунар. конф. – Челябинск: Изд-во ЮУрГУ. 2010. С. 287 – 298.

С.В. Роцин, И.И. Зиновьев

Владимирский государственный университет им. А.Г. и Н.Г. Столетовых

ЗАДАЧА ОБНАРУЖЕНИЯ ЛИЦ НА ИЗОБРАЖЕНИИ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ КОМПЬЮТЕРОВ

Задача обнаружения лица на изображении (face detection) часто является «первым шагом», предобработкой в процессе решения задачи «более высокого уровня» (например, узнавание лица, распознавание выражения лица). Однако и сама информация о присутствии и, возможно, количестве лиц на изображении или в видеопотоке может быть полезна для таких приложений, как охранные системы и содержательная индексация базы данных изображений или видеофрагментов.

Существующие алгоритмы обнаружения лиц можно разбить на две широкие категории. К первой категории относятся методы, отталкивающиеся от опыта человека в распознавании лиц и делающие попытку формализовать и алгоритмизовать этот опыт, построив на его основе автоматическую систему распознавания. Вторая категория опирается на инструментарий распознавания образов, рассматривая задачу обнаружения лица как частный случай задачи распознавания.

Основа методов первой категории – эмпирика является одновременно их сильной и слабой стороной. Трудно эффективно перевести неформальный человеческий опыт и знания в набор формальных правил, поскольку чересчур жесткие рамки правил приведут к тому, что в ряде случаев лица не будут обнаружены и, напротив, слишком общие правила приведут к большому количеству случаев ложного обнаружения. Методы вто-