РОССИЙСКАЯ АКАДЕМИЯ НАУК
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

СООБЩЕНИЯ ПО ПРИКЛАДНОЙ МАТЕМАТИКЕ

М.В. КОЗЛОВ, Ю.Е. МАЛАШЕНКО,
И.А. НАЗАРОВА

ГАРАНТИРОВАННЫЕ ОЦЕНКИ РАСПРЕДЕЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР РАН
МОСКВА 2011

УДК 519.85

Ответственный редактор
член-корр. РАН, доктор физ.-матем. наук
Ю.А. Флеров

Исследуется проблема управления ресурсами вычислительного комплекса при неизвестной длительности выполнения заданий, поступивших на обработку. Предложена модель, использующая методы оптимизации и принцип гарантированного результата, позволяющая строить априорные экспертные оценки разделения ресурсов между пользователями.

Ключевые слова: распределенные вычисления, математическое моделирование, принцип гарантированного результата.

Публикуется в авторской редакции.

Рецензенты: М.Г. Коновалов,
А.В. Лотов

Научное издание

Введение

В последние годы существенно возрос интерес к выполнению сложных задач, допускающих эффективное решение при использовании многопроцессорных специализированных вычислительных систем (СВС) и комплексов. Гетерогенные многопроцессорные системы предполагают совместное использование вычислительных ресурсов, включая программное обеспечение, базы данных, а также вычислительные модули, группами конкурирующих, сотрудничающих или взаимосвязанных потребителей. Состав ресурсов внутри СВС может варьироваться от отдельных переносных компьютеров до мощных кластеров, содержащих большое число процессоров и специальные ускорители.

Для всех гетерогенных СВС без исключения существует общая научно-техническая проблема, которая заключается в отыскании способов эффективного распределения ресурсов между потребителями [1]. Ряд аспектов указанной проблемы анализируется в [2], где содержится, не претендующий на полноту, обзор литературы по данному вопросу.

В современной отечественной практике применения высокопроизводительных вычислительных систем большую группу составляют потоковые задачи обработки информации [3], для решения которых используются массово-параллельные системы [4]. Имеются примеры успешных реализаций программных комплексов [5, 6], обязательным элементом которых является пакетирование фрагментов, что сокращает процедуры обмена, повышает надежность и упрощает управление в условиях гетерогенности аппаратно-программной среды [7-11].

Практическая проблема управления процессом коллективного использования гетерогенных вычислительных ресурсов является одной из ключевых для повышения эффективности последних, однако учет интересов конкретного пользователя и анализ показателей выполнения каждого вычислительного задания зачастую остаются за рамками изучаемых вопросов.

В настоящей работе, продолжающей исследования начатые в [12-14], анализируется специальный класс ресурсоемких вычислительных заданий, выполняемых в условиях неопределенности и допускающих распараллеливание по данным. В [15] алгоритмические процедуры решения соответствующих задач определены как случайный перебор с неизвестным исходом. К важнейшим особенностям рассматриваемого класса задач следует отнести их равнозначность (или важность) для пользователей, которые, в свою очередь равноправны в рамках СВС коллективного пользования и требуют недискриминирующего распределения ресурсов.

Для разработки алгоритмов управления СВС, в данной работе предлагается модель, базирующаяся на методах оптимизации [16] и гарантированного результата [17]. В разделе 2 процесс обработки заданий описывается и анализируется с помощью системы динамических уравнений и неравенств, записанных в дискретном времени. В разделе 3 для неизвестных величин строятся гарантированные оценки, на основании которых и принимается решение о распределении вычислительных ресурсов.

1. Специализированные вычислительные задания и схема их выполнения

Рассмотрим специальный класс ресурсоемких заданий, допускающих распараллеливание по данным и выполняющихся в условиях неопределенности. Вычислительная задача состоит в следующем: из имеющегося набора данных требуется выделить фрагмент с уникальными свойствами. Для достижения поставленной цели необходимо реализовать некоторую алгоритмическую процедуру для массива начальных данных, разбитых на неделимые, содержательно значимые, фрагменты. Если при выполнении задания, удается выявить уникальный фрагмент, то говорят, что задача решена, и обработка задания прекращается. Если же в ходе вычислительного процесса просматривается весь массив предъявленных данных и найти уникальный фрагмент не удается, то задание считается выполненным, но поставленная задача не решена.

Процесс выделения уникального фрагмента проходит следующие основные этапы:

— создание и/или рекуррентное описание большого массива отдельных фрагментов данных;

— формулировка правила, по которому некоторый фрагмент объявляется уникальным;

— выбор наугад фрагмента из массива, и, наконец, проверка его уникальности.

При реализации алгоритмических процедур поиска уникального фрагмента необходимо учитывать следующие особенности исходной проблемы:

— допускается разбиение массива данных на отдельные, даже пересекающиеся подмножества, таким образом, что решение исходной задачи может быть получено в результате обработки подмассива;

— для завершения задачи не всегда требуется обработка всех наборов данных, зачастую достаточно найти один уникальный фрагмент;

— исходная содержательная постановка задач предусматривает проведение всех заданных работ в срок и возможность планирования при распределении вычислительных ресурсов во времени.

Задачи поступают в произвольные моменты времени, без каких-либо особых признаков, отношений предпочтения или предшествования в обслуживании. Обработка соответствующих массивов данных требует большого, точно не известного, объема вычислений. В общем случае можно указать только верхнюю оценку для величины затрат вычислительных ресурсов необходимых для выполнения конкретного задания; имеет ли решение соответствующая содержательная задача — неизвестно. Таким образом, при планировании вычислительных работ возникает неопределенность, связанная как с моментами поступления и составом заданий, так и с длительностью их обработки.

Рассматриваемый специальный класс заданий будем далее для краткости обозначать: citu-задания, citu-работы, citu-задачи; от английского термина: computationally intensive task (under) uncertainty, который можно перевести как ресурсоемкие вычислительные задания, выполняемые в условиях неопределенности.

Данный класс citu-задач обладает целым рядом существенных особенностей. В частности, все задания выполняются в реальном времени, решения одинаково важны и должны быть получены — каждое в отдельности и все в совокупности — как можно быстрее.

Для всех citu-заданий задаются директивные сроки окончания (ДСО). Превышение ДСО приводит к тому, что решение соответствующей задачи теряет актуальность, а процессорное время, затраченное вычислительными ресурсами записывается в производственные потери.

С содержательной точки зрения, в рассматриваемом классе citu-задачи в определенном смысле персонифицированы: все они *равнозначны*, а пользователи, предоставляющие задачи для обработки, *равноправны* в рамках правил распределения вычислительных ресурсов, поэтому стратегии управления их выполнением не должны быть дискриминирующими по отношению к той или иной группе заявок.

Для рассматриваемой модели под *недискриминирующим дележом* ограниченного ресурса между равноправными пользователями будем понимать распределение, при котором ни один из партнеров не может улучшить временные характеристики выполнения своей заявки, не ухудшив соответствующие показатели своих коллег. Для данного класса citu-работ считается, что пользователи не являются антагонистами и при обработке пакетов взаимосвязанных заданий стремятся к достижению некоторой общей корпоративной цели.

Для целей модельного описания предполагается, что citu-задания выполняются на специализированной вычислительной системе (СВС), которая состоит из центрального управляющего устройства (ЦУП-устройства) и набора независимых исполняющих единичных вычислительных модулей (ЕВМ).

Схематично процесс обработки citu-заданий в СВС можно описать следующим образом. Все citu-задания анализируются и подготавливаются к исполнению на входе в СВС. Далее вся информация о citu-заданиях, вновь поступивших или находящихся в обработке, заносится и далее хранится в так называемой базе данных заданий (БД-заданий). Информация из последней поступает в реальном времени в программный комплекс анализа и планирования (ПК ПЛАН), где формируется пакет текущих работ (ПАКЕТ-ТР).

ПАКЕТ-ТР состоит из подмассивов (поднаборов) неделимых фрагментов данных тех citu-заданий, список которых составляется также в ПК ПЛАН.

В контрольный момент времени ПАКЕТ-ТР поступает в промежуточную базу данных текущих работ (БДТ-работ) и начинает обрабатываться на всех работоспособных ЕВМ.

В основу ПК ПЛАН положено несколько взаимосвязанных математических моделей, в которых процесс обработки citu-заданий в СВС описывается и анализируется с помощью системы динамических уравнений и неравенств, записанных в дискретном времени. В частности, для изначально неизвестных величин строятся гарантированные оценки, на основании которых последовательно решается ряд оптимизационных задач, что позволяет

— максимизировать суммарную эффективность гетерогенных вычислительных ресурсов;

— получить недискриминирующее распределение ресурсов между заданиями;

— определить неулучшаемые относительные показатели времени пребывания каждого конкретного задания в СВС с учетом имеющейся информации об остальных заданиях;

— в условиях неопределенности выполнить достаточные условия завершения каждого citu-задания к назначенному сроку.

2. Общее описание модели

2.1. Задания и задачи

Обозначим:

z_n — задание-задачу с собственным идентификационным номером n;

$\mathbf{Z}_n$ — общее (нормативное) число неделимых фрагментов данных которые необходимо будет обработать для задания z_n, если соответствующая ему задача не имеет решения, т.е. $\mathbf{Z}_n$ — число вычислительных операций для полного перебора всего исходного массива фрагментов данных, если среди них нет уникального;

t — текущий момент календарного времени, а также контрольная точка — момент принятия решения об изменении управления выполнением заданий;

$\mathcal{N}(t)$ — множество номеров (индексов) заданий, находящихся в СВС в момент t;

$N(t) = |\mathcal{N}(t)|$ — общее число заданий, находящихся в СВС в момент t;

$\Delta(t)$ — плановый период с началом в момент времени t; промежуток (интервал) времени, длительность которого определяется заранее или выбирается при рассчетах в ПК-ПЛАН; $\Delta(t)$ измеряется в единицах календарного времени.

В ПК-ПЛАН формируется ряд гарантированных оценок и ограничений, на основе которых вычисляются параметры управления. Фактически в ПК-ПЛАН составляется список заданий, для них определяется размер текущих подмассивов неделимых фрагментов данных, которые и составляют пакет текущих работ. Далее ПАКЕТ-ТР размещается в БДТ-работ, и в контрольный момент t начнется его обработка, которую планируется завершить не позднее $t + \Delta(t)$.

Обозначим:

$w_n(\Delta(t))$ — число неделимых фрагментов данных для задания z_n, которые войдут в состав ПАКЕТ-ТР, будут помещены в БДТ-работ и в момент времени t начнут обрабатываться по команде ЦУП-устройства на назначенных ЕВМ. Комплект подмассивов данных, составляющих ПАКЕТ-ТР, обозначим

$$\mathbf{w}(t) = \{w_1(\Delta(t)), w_2(\Delta(t)), ..., w_{N(t)}(\Delta(t))\}.$$

Пусть t_n^0 — момент поступления задания z_n в СВС, и в момент времени t оно еще не завершено. Обозначим:

$z_n^-(t)$ — число неделимых фрагментов данных, которые уже были обработаны для задания z_n от момента его поступления в СВС до момента времени t.

При составлении ПАКЕТ-ТР предполагается, что все $\mathbf{w}(t)$ будут завершены до окончания планового периода $\Delta(t)$. Если в поднаборе $w_n(\Delta(t))$ фрагментов данных из z_n не окажется уникального, то в момент $(t + \Delta(t))$ будет справедливо равенство

$$z_n^-(t + \Delta(t)) = z_n^-(t) + w_n(\Delta(t)).$$

Монопольным режимом обработки задания z_n будем называть такой способ организации работы СВС, при котором на всех имеющихся работоспособных ЕВМ одновременно просматриваются фрагменты данных единственного задания z_n. Обозначим:

τ_n — абсолютное время выполнения задания z_n в монопольном режиме СВС;

$R(t)$ — общее число работоспособных ЕВМ в СВС в момент времени t;

p_0 — производительность (быстродействие) ЕВМ — число неделимых фрагментов данных, которые могут быть обработаны одним ЕВМ в единицу времени. Тогда суммарная производительность $P(t)$ системы СВС в единицу времени составляет

$$P(t) = p_0 \cdot R(t).$$

2.2. Гарантированные оценки времени выполнения задания

Предположим, что в поднаборе $w_n(\Delta(t))$ либо находится уникальный фрагмент данных задания z_n, либо обработка $w_n(\Delta(t))$ завершает его выполнение. Тогда значение

$$z_n^-(t + \Delta(t)) = z_n^-(t) + w_n(\Delta(t))$$

является оценкой сверху для общего числа фрагментов данных, которые в действительности необходимо просмотреть для завершения задания z_n или для решения соответствующей ему задачи. Следовательно, величина

$$\overline{\tau}_n(\Delta(t)) = \frac{z_n^-(t) + w_n(\Delta(t))}{p_0 \cdot R(t)}$$

является оценкой сверху для τ_n в момент времени $t + \Delta(t)$, и

$$\overline{\tau}_n(\Delta(t)) \geq \tau_n.$$

Обозначим через $T_n(t)$ длительность пребывания z_n в СВС на момент времени t,

$$T_n(t) = t - t_n^0,$$

напомним, что здесь t_n^0 — момент поступления задания z_n в СВС.

Предположим, что выполнение задания z_n будет завершено к моменту $t + \Delta(t)$, тогда время пребывания z_n в СВС определяется равенством

$$T_n(t + \Delta(t)) = t + \Delta(t) - t_n^0.$$

Введем переменную

$$\chi_n(t + \Delta(t)) = \frac{\overline{\tau}_n(\Delta(t))}{T_n(t + \Delta(t))}, \qquad (2.1)$$

которую будем называть *показателем относительной задержки выполнения задания* z_n к моменту времени $(t + \Delta(t))$.

Действительно, величина $\chi_n(t + \Delta(t))$ показывает, какую долю составляет абсолютное время обработки z_n в монопольном режиме от фактического времени пребывания в СВС, при условии, что оно будет завершено к моменту $(t + \Delta(t))$.

Обратную величину $æ_n(t+\Delta(t)) = 1/\chi_n(t+\Delta(t))$ назовем коэффициентом относительной задержки. Последний показывает, во сколько раз время фактического пребывания в системе оказалось больше необходимого времени обработки, т.е. характеризует задержку выполнения z_n относительно абсолютного времени его обработки в монопольном режиме.

В ПК-ПЛАН в каждой контрольной точке t величины выполняемых подзаданий $w_n(\Delta(t))$ выбираются так, чтобы минимальная величина $\chi_n(t+\Delta(t))$ оказалась максимальной из возможных для всех заданий z_n. Формально ведется поиск

$$\theta^*(t) = \max_{w(t)} \min_{1\le n\le N(t)} \chi_n(t+\Delta(t)) \qquad (2.2)$$

Другими словами цель управления состоит в том, чтобы задачи z_n с небольшим абсолютным временем решения не находились в СВС очень долго и покидали систему в первую очередь.

Предположим, что в момент времени t максимальный объем ПАКЕТ-ТР задан. Обозначим через $W^+(\Delta(t))$ максимальное суммарное число неделимых фрагментов, которые будут помещены в БДТ-работ и начнут выполняться в момент времени t. При этом, если задано $\Delta(t)$, то

$$W^+(\Delta(t)) = p_0 \cdot R(t) \cdot \Delta(t),$$

здесь $R(t)$ — общее число работоспособных ЕВМ в СВС в момент t. В сделанных предположениях поиск решения (2.2) можно свести к следующей задаче линейного программирования:

найти

$$\max_{w(t)} \theta(t), \qquad (2.3)$$

при ограничениях:

$$\left.\begin{array}{l}\theta(t) \le \chi_n(t+\Delta(t)),\\ \chi_n(t+\Delta(t)) = \dfrac{z_n^-(t) + w_n(\Delta(t))}{P(t)\cdot T_n(t+\Delta(t))},\\ \sum\limits_{n\in\mathcal{N}(t)} w_n(\Delta(t)) \le W^+(\Delta(t)),\\ 0 \le w_n(\Delta(t)) \le \mathbf{Z}_n - z_n^-(t), \quad n\in\mathcal{N}(t).\end{array}\right\} \tag{2.4}$$

Обозначим оптимальное решение (2.3)-(2.4) через

$$< \theta^*(t), w_1^*(\Delta(t)), w_2^*(\Delta(t)), \ldots, w_{N(t)}^*(\Delta(t)) > .$$

Введем множества:

$\mathcal{N}^*(t)$ — множество номеров задач z_n таких, что
$n \in \mathcal{N}^*(t)$ при условии $w_n^*(\Delta(t)) > 0, \theta^*(t) = \chi_n^*(t+\Delta(t))$;
$\mathcal{N}^0(t)$ — множество номеров задач z_n таких, что
$n \in \mathcal{N}^0(t)$, при условии $w_n^*(\Delta(t)) = 0, \theta^*(t) \le \chi_n^*(t+\Delta(t))$.
Из определения следует $\mathcal{N}(t) = \mathcal{N}^*(t)\bigcup\mathcal{N}^0(t)$.

Следовательно, оптимальное решение (2.3)-(2.4) — размеры подзаданий для выполнения к моменту $t+\Delta(t)$, удовлетворяют равенствам

$$\theta^*(t) = \frac{z_n^-(t) + w_n^*(\Delta(t))}{P(t)\cdot T_n(t+\Delta(t))}, \quad n\in\mathcal{N}^*(t),$$

$$\sum_{n\in\mathcal{N}^*(t)} w_n^*(t) = W^+(\Delta(t)).$$

После преобразований получим:

$$\theta^*(t) = \frac{W^+(\Delta(t)) + \sum\limits_{n\in\mathcal{N}^*(t)} z_n^-(t)}{P(t)\cdot \sum\limits_{n\in\mathcal{N}^*(t)} T_n(t+\Delta(t))} \tag{2.5}$$

$$w_n^*(\Delta(t)) = \theta^*(t)\cdot P(t)\cdot T_n(t+\Delta(t)) - z_n^-(t) =$$

$$= T_n(t+\Delta(t)) \cdot \frac{W^+(\Delta(t)) + \sum\limits_{n\in\mathcal{N}^*(t)} z_n^-(t)}{\sum\limits_{n\in\mathcal{N}^*(t)} T_n(t+\Delta(t))} - z_n^-(t), n \in \mathcal{N}^*(t).$$

Вектор значений

$$\mathbf{w}^*(t) =< w_1^*(\Delta(t)), w_2^*(\Delta(t)), \ldots, w_{N(t)}^*(\Delta(t)) > \qquad (2.6)$$

будем называть равноправным недискриминирующим по критерию (2.2) распределением выполняемых подзаданий $w_n(\Delta(t))$. При этом для всех z_n, завершенных на интервале $\Delta(t)$, показатель относительной задержки выполнения — не меньше $\theta^*(t)$.

Указанное распределение предполагает, что все пользователи являются равноправными по отношению к разделению вычислительных ресурсов. Все заявки-задания получают долю просмотренных фрагментов данных соответствующую их реальному, хотя и неизвестному в текущий момент, времени решения с учетом фактического пребывания в СВС. При таком распределении увеличение доли $w_n^*(\Delta(t))$ для любой заявки z_n приведет к уменьшению значения (2.2) для остальных пользователей. С формальной точки зрения предлагаемый способ выбора размеров подзаданий можно рассматривать как обобщение известного метода Round-Robin [18] разделения операционного времени центрального процессора.

2.3. Одновременное выполнение пакета текущих программ

Предположим, что решена оптимизационная задача (2.3)-(2.4) и полученный набор подзаданий

$$\mathbf{w}^*(t) =< w_1^*(\Delta(t)), w_2^*(\Delta(t)), \ldots, w_{N(t)}^*(\Delta(t)) >$$

фрагментов данных для выбранных заданий $z_n, n \in \mathcal{N}^*(t)$, т.е. ПАКЕТ-ТР, помещается в промежуточную базу данных текущих работ (БДТ-работ) и в момент t начинает обрабатываться системой.

Рассмотрим правило обработки пакета, при котором каждому подзаданию $w_n^*(\Delta(t))$ назначается некоторый набор ЕВМ. Например, диспетчер и/или некая программа просто формирует соответствующий список, где указывает какие ЕВМ будут выполнять данное подзадание $w_n^*(\Delta(t))$ начиная с момента t. При этом, если некоторая подзадача $w_n^*(\Delta(t))$ будет решена до истечения срока $\Delta(t)$, то освободившийся ресурс — ЕВМ — будет использован ЦУП-устройством для обработки других подзаданий $w_l^*(\Delta(t))$, которые еще не завершены. Описанную процедуру назовем параллельной обработкой ПАКЕТ-ТР.

Пусть $r_n^*(t)$ — планируемое число ЕВМ, которые в момент t должны начать выполнять подзадание $w_n^*(\Delta(t))$, и завершат его обработку в момент $(t+\Delta(t))$, если среди $w_n^*(\Delta(t))$ не будет найден уникальный. Тогда, поскольку, p_0 — число фрагментов, которые обрабатываются одним ЕВМ за единицу времени, то

$$w_n^*(\Delta(t)) = r_n^*(t) \cdot p_0 \cdot \Delta(t), \quad n \in \mathcal{N}^*(t).$$

При этом

$$\sum_{n \in \mathcal{N}^*(t)} r_n^*(t) = R(t),$$

$$r_n^*(t) = 0, n \in \mathcal{N}^0(t).$$

Из предположений п. 2.2 следует

$$W^+(t) = R(t) \cdot p_0 \cdot \Delta(t), \tag{2.7}$$

а из соотношений (2.5),(2.7) —

$$r_n^*(t) \cdot p_0 \cdot \Delta(t) = \theta^*(t) \cdot p_0 \cdot R(t) \cdot T_n(t + \Delta(t)) - z_n^-(t), \ n \in \mathcal{N}^*(t),$$

где
$$\theta^*(t) = \frac{R(t)\cdot p_0\cdot\Delta(t) + \sum\limits_{n\in\mathcal{N}^*(t)} z_n^-(t)}{R(t)\cdot p_0\cdot \sum\limits_{n\in\mathcal{N}^*(t)} T_n(t+\Delta(t))} =$$
$$= \frac{\Delta(t)}{\sum\limits_{n\in\mathcal{N}^*(t)} T_n(t+\Delta(t))} + \frac{\sum\limits_{n\in\mathcal{N}^*(t)} z_n^-(t)}{R(t)\cdot p_0\cdot \sum\limits_{n\in\mathcal{N}^*(t)} T_n(t+\Delta(t))}.$$

Вектор значений $\mathbf{r}^* =< r_1^*(t), r_2^*(t), \ldots, r_{N(t)}^*(t) >$ является оптимальным распределением вычислительных ресурсов $R(t)$ по критерию (2.2) при запуске параллельной обработки пакета в момент t. Если какие-то подзадачи $w_l^*(\Delta(t))$ будут решены на интервале $\Delta(t)$, то фактическое суммарное время выполнения данного ПАКЕТ-ТР $\mathbf{w}^*(t)$ окажется меньше $\Delta(t)$, и очередной контрольный момент наступит раньше $(t+\Delta(t))$.

2.4. Последовательное выполнение пакета текущих программ

Предположим, как и ранее, что в базе БДТ-работ в момент t находится пакет выбранных подзаданий $\mathbf{w}^* =< w_1^*(\Delta(t)), w_2^*(\Delta(t)), \ldots, w_{N(t)}^*(\Delta(t)) >$, который планируется выполнить в течение периода $\Delta(t)$. Правило обработки заданий из пакета будем называть последовательным, если подзадания $w_n^*(\Delta(t))$ выполняются одно за другим в монопольном режиме СВС, т.е. каждое сразу на всех работоспособных ЕВМ. Как только очередная подзадача $w_n^*(\Delta(t))$ завершена, сразу начинается обработка следующего поднабора данных.

Обозначим через Δ_n^* гарантированную оценку — промежуток времени, которое необходимо для обработки выбранного подзадания $w_n^*(\Delta(t))$ в монопольном режиме, при условии,что задача z_n не будет решена за $\Delta(t)$. Для всех подзаданий ПАКЕТА-ТР априори требуется выполнение равенства

$$\sum_{n \in \mathcal{N}^*(t)} \Delta_n^* = \Delta(t).$$

Поскольку $w_n^*(\Delta(t))$ — число неделимых фрагментов данных, которые будут обработаны в монопольном режиме на $R(t)$ работоспособных ЕВМ за время Δ_n^*, если в них нет уникального, то должно выполняться равенство

$$w_n^*(\Delta(t)) = p_0 \cdot R(t) \cdot \Delta_n^*, \ n \in \mathcal{N}^*(t).$$

Таким образом, из предыдущего равенства и (2.5) следует

$$\Delta_n^* = \frac{w_n^*(\Delta(t))}{p_0 \cdot R(t)} =$$

$$= \frac{1}{p_0 \cdot R(t)} \cdot \left[\theta^*(t) \cdot p_0 \cdot R(t) \cdot T_n(t + \Delta(t)) - z_n^-(t)\right],$$

где

$$\theta^*(t) = \frac{p_0 \cdot R(t) \cdot \Delta(t) + \sum\limits_{n \in \mathcal{N}^*(t)} z_n^-(t)}{p_0 \cdot R(t) \cdot \sum\limits_{n \in \mathcal{N}^*(t)} T_n(t + \Delta(t))} =$$

$$= \frac{\Delta(t)}{\sum\limits_{n \in \mathcal{N}^*(t)} T_n(t + \Delta(t))} +$$

$$+ \frac{1}{p_0 \cdot R(t)} \cdot \frac{\sum\limits_{n \in \mathcal{N}^*(t)} z_n^-(t)}{\sum\limits_{n \in \mathcal{N}^*(t)} T_n(t + \Delta(t))}.$$

Для соотношений, приведенных выше, справедливо

$$\frac{w_n^*(\Delta(t))}{W^+(t + \Delta(t))} = \frac{r_n^*(t)}{R(t)} = \frac{\Delta_n^*}{\Delta(t)}, n \in \mathcal{N}^*(t).$$

3. Модель управления поэтапным выполнением заданий

3.1. Общее описание

Рассмотрим динамическую оптимизационную модель (M-модель) управления работой гетерогенной вычислительной системы, в которой в условиях неопределенности выполняются citu-задачи — ресурсоемкие разнородные вычислительные работы, допускающие распараллеливание по данным.

Для формализации описания процедуры выполнения каждого citu-задания используем теоретико-вероятностную модель — случайную выборку без возвращений [20]. В рамках упомянутой модели каждому citu-заданию соответствует урна с шарами двух цветов, а процесс его выполнения рассматривается как извлечение шаров из урны и определение их цвета. Каждому неделимому содержательному фрагменту исходных данных ставится в соответствие отдельный шар, пусть белый, а уникальному фрагменту — например, черный, при этом поиск уникального фрагмента сводится к отысканию и извлечению из урны черного шара.

Под специализированной элементарной вычислительной операцией (СЭВ-операция, СЭВО) будем понимать извлечение из урны одного шара и идентификацию его цвета. Для исходной проблемы с вычислительной точки зрения СЭВ-операция сводится к обработке (просмотру) отдельного неделимого фрагмента данных и проверке его на уникальность.

В рамках M-модели считается, что исходные citu-задания бывают разных видов, вернее исходные данные могут иметь разную физическую природу и содержательный смысл. Такие citu-задания требуют обработки отличающимися друг от друга алгоритмическими процедурами на единичных вычислительных модулях различных конструктивных типов. В дальнейшем, в качестве заданий разных видов рассматриваются урны с наборами шаров определенных цветов, а цвет шара, ассоциируемого с уникальным фрагментом, известен заранее. Например, в урнах с бело-черными шарами ведется поиск черного шара, в урне с желто-зелеными — зеленого, среди красно-коричневых уникальным пусть будет коричневый, а для сине-голубых задача считается решенной после извлечения синего шара и т.д.

Будем говорить, что задание завершено, но задача не имеет решения, если после извлечения из урны все шары оказались одного цвета. Последнее событие в реальности означает, что при обработке и просмотре всех предъявленных фрагментов данных уникальный среди них не обнаружен. При этом формально приходится признать вычислительную работу выполненной, хотя цель не достигнута — поиск ничего не дал. К такому результату может привести сбой в работе вычислительной системы, неправильная подготовка данных для citu-задания и ряд других причин.

Рассмотрим ситуацию, когда в урне содержатся шары двух цветов. В этом случае выполнение задания прекращается, а задача полагается решенной после извлечения первого шара, цвет которого объявлен соответствующим уникальному фрагменту. Так, например, для урны с черно-белыми шарами задача считается решенной после того, как обнаружен первый черный шар.

В рамках модели предполагается, что citu-работы различных видов поступают в систему последовательно, через случайные промежутки времени без каких-либо признаков предпочтения или отношений предшествования между ними. Соответствующие СЭВО выполняются гетерогенной специализированной вычислительной системой (СВ-системой, СВС), которая состоит из центрального управляющего устройства

(ЦУП-устройства) и набора единичных вычислительных модулей (ЕВМ). Диспетчер-оператор организует и контролирует процесс выполнения заданий, поступающих от пользователей, а с помощью ЦУП-устройства осуществляется координация работы всех ЕВМ.

Каждый ЕВ-модуль представляет собой некий абстрактный автономный вычислитель со своим блоком управления и памятью, без детализации архитектуры и конструктивных особенностей. Производительность ЕВМ определяется как число выполняемых СЭВО в единицу времени, которая для разных типов ЕВМ различна. Предполагается, что каждый конкретный ЕВМ может выполнять СЭВО по крайней мере одного вида, а его производительность при выполнении citu-заданий различных видов неодинакова.

Считается, что СВ-система может одновременно обрабатывать как одно, так и несколько заданий на всех работоспособных в данный момент ЕВМ. Предполагается, что отдельное задание может быть разделено на подзадания, каждое из которых может в свою очередь выполняться самостоятельно и/или в составе некоторого набора (пакета).

В рассматриваемой М-модели процесс выполнения подзаданий в СВС описывается системой конечно-разностных уравнений в дискретном времени. Шаг по времени называется плановым периодом и/или операционным окном. В каждой контрольной точке при оптимизации числа и размеров подзаданий предполагается, что весь пакет работ может быть выполнен за плановый период.

Заметим, что в силу объективной неопределенности, присущей проблеме, ПАКЕТ-ТР может быть выполнен за отрезок времени, длительность которого меньше планового. Именно поэтому для управления используется идея скользящего планирования, а именно, вся информация о заданиях обрабатывается в ходе вычислительного процесса. В результате, в момент завершения обработки ПАКЕТ-ТР — контрольной точке для запуска следующего пакета, последний оказывается сформирован и может поступить на обработку.

3.2. Основные понятия и обозначения

3.2.1. Задания и задачи

Далее в тексте верхний индекс будет всегда относиться к типу вычислительного устройства, а нижний, один или два, в зависимости от рассматриваемого случая, — отвечать номеру и виду задачи соответственно. Для того, чтобы избежать путаницы второй нижний индекс, соответствующий виду задачи, помещен в скобки.

Введем следующие обозначения:

K — общее число различных видов заданий, которые обрабатываются СВС;

$z_{n(k)}$ — задание-задача с собственным идентификационным номером n, для которой явно указан ее вид k, $k = \overline{1, K}$; обозначение будет использовано для удобства прочтения формул, описывающих процесс обработки заданий определенного вида;

z_n — задание-задача с собственным идентификационным номером n; обозначение будет использовано в случаях, когда вид задачи значения не имеет;

t — календарное время; фиксированный, определенный момент времени;

$\mathcal{Z}(t)$ — множество заданий z_n, находящихся в СВС в момент t;

$\mathcal{N}(t)$ — множество номеров (индексов) заданий z_n, находящихся в СВС в момент t.

$N(t)$ — общее число заданий z_n в СВС в момент t. Таким образом $|\mathcal{Z}(t)| = |\mathcal{N}(t)| = N(t)$;

$\mathcal{Z}_k(t)$ — множество заданий k-го вида, находящихся в СВС в момент t;

$\mathcal{N}_k(t)$ — множество номеров (индексов) заданий k-го вида, находящихся в СВС в момент t;

$N_k(t)$ — общее число заданий k-го вида в СВС в момент t.

Тогда

$$\mathcal{Z}(t) = \bigcup_{k=1}^{K} \mathcal{Z}_k(t), \ \mathcal{N}(t) = \bigcup_{k=1}^{K} \mathcal{N}_k(t), \ N(t) = \sum_{k=1}^{K} N_k(t).$$

3.2.2. Временные характеристики заданий

Введем следующие обозначения

t_n^0 — момент поступления задания z_n в СВС;

d_n — директивный срок окончания (ДСО) — календарный момент времени до наступления которого задание z_n должно быть завершено;

$T_n(t) = (t - t_n^0)$ — длительность промежутка времени, в течение которого, начиная с момента его поступления, задание z_n находится в СВС при условии, что на момент t оно еще не завершено т.е. $T_n(t)$ — число единиц времени, которое прошло от момента t_n^0 поступления z_n в СВС до текущего момента t;

t_n^+ — момент завершения задания z_n и его выхода из СВС;

$T_n^+ = (t_n^+ - t_n^0)$ — длительность промежутка времени, в течение которого задание z_n находится в СВС начиная с момента поступления t_n^0 до своего завершения, т.е. T_n^+ — время (длительность) пребывания z_n в системе.

В М-модели предполагается, что для каждого задания z_n в момент t_n^0 становится известна нормативная величина

$\mathbf{Z}_n$ — общее число СЭВО, которые будет необходимо выполнить для данного задания в случае когда соответствующая задача не имеет решения, т.е. $\mathbf{Z}_n$ — число операций для полного перебора всего исходного массива неделимых фрагментов данных при отсутствии среди них уникального, или общее число шаров в урне z_n.

3.2.3. СВС

Под специализированной вычислительной системой (СВС) $\mathcal{H} = \{\mathbf{H}, \mathbf{E}\}$ будем понимать:

H — ЦУП-устройство, распределяющее задания для обработки по имеющимся ЕВМ и осуществляющее контроль за их выполнением; и

E — набор единичных вычислительных модулей ЕВМ различных типов.

Обозначим:

M — число различных типов ЕВМ, из которых состоит рассматриваемая СВС;

e^m — единичное устройство m-го типа;

p_k^m — быстродействие и/или производительность ЕВМ m-го типа для заданий k-го вида, т.е. e^m выполняет p_k^m СЭВО k-го вида в единицу времени;

$R^m(t)$ — общее число работоспособных на момент времени t ЕВМ m-го типа. Тогда

$P_k(t)$ — максимальное число СЭВО k-го вида, которые можно выполнить в единицу времени на СВС, начиная с момента t на всех работоспособных ЕВМ, находится из соотношения

$$P_k(t) = \sum_{m=1}^{M} p_k^m \cdot R^m(t), \quad k = \overline{1, K}; \tag{3.1}$$

E^m — множество, состоящее из $R^m(t)$ работоспособных в момент t ЕВМ m-го типа $e^m, m = \overline{1, M}$, тогда множество $\mathbf{E}$ всех

ЕВМ в СВС можно записать в виде следующего объединения

$$\mathbf{E} = E^1 \bigcup E^2 \bigcup ... \bigcup E^M.$$

3.2.4 Параметры планирования и управления

В рамках рассматриваемой динамической оптимизационной М-модели функционирование СВС схематично описывается следующим образом. В контрольной точке принятия решения t анализируются данные о состоянии выполнения всех заданий находящихся в СВС. Далее Диспетчер, специалист и/или алгоритмическая процедура, составляет ПАКЕТ-ТР подзаданий, которые начнут выполняться в СВС в момент времени t. Считается, что диспетчер выбирает число единиц календарного времени $\Delta(t)$ — плановый период, или оперативное окно. Фактически через t в зависимости от контекста обозначим, либо текущий календарный момент времени, либо контрольную точку принятия решения о формировании пакета подзаданий, которые будут выполняться в течение следующего планового периода $\Delta(t)$.

Подзадания ПАКЕТ-ТР представляют собой различающиеся по объему наборы фрагментов данных для различных заданий $z_n, n \in \mathcal{N}(t)$. В рамках рассматриваемой модели отдельное подзадание ПАКЕТ-ТР — некоторый набор шаров из урны $z_n, n \in \mathcal{N}(t)$. Размер подзадания для $z_{n(k)}$ обозначим $w_{n(k)}(\Delta(t))$. Таким образом,

$w_{n(k)}(\Delta(t))$ — число СЭВО k-го вида, которые планируется выполнить на интервале времени $\Delta(t)$ для задания $z_{n(k)}$, или число шаров в поднаборе, взятом для обработки в течение планового периода $\Delta(t)$ из урны $z_{n(k)}$. Переменные $w_{n(k)}(\Delta(t))$, $n \in \mathcal{N}(t)$, являются управлениями или управляющими параметрами.

Обозначим:

$z^{-}_{n(k)}(t)$ — число СЭВО k-го вида, которые уже были выполнены для задания z_n от момента его поступления в СВС до момента времени t. В рамках модели $z^{-}_{n(k)}(t)$ — число шаров, которые уже были извлечены из урны z_n к моменту t;

$z^{+}_{n(k)}(t)$ — число шаров, которые осталось в урне z_n к моменту t. Тогда

$$z^{+}_{n(k)}(t) = \mathbf{Z}_n - z^{-}_{n(k)}(t), \quad n \in \mathcal{N}(t). \tag{3.2}$$

Очевидно, что управления $w_{n(k)}(\Delta(t))$ на интервале $\Delta(t)$ должны удовлетворять ограничениям

$$0 \leq w_{n(k)}(\Delta(t)) \leq z^{+}_{n(k)}(t), \quad n \in \mathcal{N}(t), \tag{3.3}$$

поскольку в момент t из урны z_n нельзя извлечь шаров больше, чем там осталось. Если среди фрагментов из поднабора $w_{n(k)}(\Delta(t))$ нет уникального, то в момент $t + \Delta(t)$

$$z^{-}_{n(k)}(t + \Delta(t)) = z^{-}_{n(k)}(t) + w_{n(k)}(\Delta(t)), n \in \mathcal{N}(t); \tag{3.4}$$

Обозначим $W_k(\Delta(t))$ — суммарное число СЭВО k-го вида, которые планируется выполнить на заданном интервале планирования $\Delta(t)$ для всех заданий. т.е.

$$W_k(\Delta(t)) = \sum_{n \in \mathcal{N}_k(t)} w_{n(k)}(\Delta(t)), \quad k = \overline{1, K}. \tag{3.5}$$

Введем переменную управления $r^m_k(t)$ — общее число ЕВМ m-го типа, которые начинают выполнять СЭВО k-го вида в момент времени t.

Предположим, что все поднаборы $w_{n(k)}(\Delta(t))$ не содержат шара, соответствующего уникальному фрагменту, но их планируется обработать за плановый период $\Delta(t)$. Тогда для выполнения всех СЭВО для подзаданий k-го вида необходимо выделить $r_k^m(t)$ ЕВМ m-го типа,

$$W_k(\Delta(t)) = \sum_{m=1}^{M} p_k^m \cdot r_k^m(t) \cdot \Delta(t), \quad k = \overline{1,K}. \qquad (3.6)$$

Поскольку число ЕВМ, распределяемых для выполнения заданий, не должно превышать общего числа работоспособных, то значения $r_k^m(t)$ должны удовлетворять следующим ограничениям:

$$\sum_{k=1}^{K} r_k^m(t) \le R^m(t), \quad m = \overline{1,M}, \qquad (3.7)$$

$$0 \le r_k^m(t), \quad k = \overline{1,K}, \quad m = \overline{1,M}.$$

3.2.5 Планирование с учетом времени

Соотношения (3.6) - (3.7) можно переписать иначе, используя понятие "время выполнения работы на некотором устройстве". Пусть $\delta_k^m(t)$ — временной интервал, в течение которого СЭВО k-го вида выполняются на ЕВМ m-го типа после момента t. В теории расписаний и/или календарного планирования [19] $\delta_k^m(t)$ — время выполнения работ (СЭВО) k-го вида на станке (ЕВМ) m-го типа. Если среди поднаборов $w_{n(k)}(t)$ нет ни одного шара, соответствующего уникальному фрагменту данных, то $\delta_k^m(t)$ — гарантированная оценка времени выполнения работ (СЭВО) k-го вида на ЕВМ m-го типа и при планировании выполнения всех подзаданий $W_k(\Delta(t))$ должны выполняться следующие равенства

$$W_k(\Delta(t)) = \sum_{m=1}^{M} p_k^m \cdot \delta_k^m(t) \cdot R^m(t), \; k = \overline{1,K},$$

$$\sum_{k=1}^{K} \delta_k^m(t) = \Delta(t), m = \overline{1, M}.$$

Откуда согласно (3.6) следует

$$\sum_{m=1}^{M} p_k^m \cdot \delta_k^m(t) \cdot R^m(t) = \sum_{m=1}^{M} p_k^m \cdot r_k^m(t) \cdot \Delta(t),\ k = \overline{1, K},$$

$$\sum_{m=1}^{M} r_k^m(t) \cdot p_k^m = \sum_{m=1}^{M} R^m(t) \cdot p_k^m \cdot \frac{\delta_k^m(t)}{\Delta(t)},\ k = \overline{1, K},$$

где $\dfrac{\delta_k^m(t)}{\Delta(t)}$ — доля времени планового периода, в течение которого ЕВМ m-го типа выполняют СЭВО k-го вида.

3.3. Анализ и формирование множества решений

3.3.1. Анализ эффективности работы СВС

Следует обратить внимание, что в реальной ситуации значения p_k^m для разных k и m могут сильно отличаться друг от друга, поскольку представляют собой число СЭВО разных видов, выполняемых в единицу времени на разнотипных ЕВМ. При формировании пакетов подзаданий — выборе значений вектора $\mathbf{w}^*(t) =< w_1^*(\Delta(t)), \ldots, w_{N(t)}^*(\Delta(t)) >$, в основном учитывается состояние выполнения заданий, поэтому в некоторые моменты может возникать недогрузка ЕВМ определенных типов. С формальной точки зрения, некоторое ограничение типа (3.7), например, для l-го типа ЕВМ,

$$\sum_{k=1}^{K} r_k^l(t) \leq R^l(t),$$

при оптимальных значениях $r^*(t)$ будет выполняться как строгое неравенство, что и означает недоиспользование ЕВМ соответствующего l-го типа. Для исследования проблемы эффективного использования СВС, в рамках М-модели решается следующая оптимизационная задача:

при заданных $\mathcal{Z}, p, \Delta(t), R(t)$ найти

$$\varphi^*(t) = \max_{w,r} \varphi(t), \tag{3.8}$$

при ограничениях

$$\varphi(t) \leq \sum_{n=1}^{N(t)} w_{n(k)}(\Delta(t)), \tag{3.9}$$

$$0 \leq w_{n(k)}(\Delta(t)) \leq z^+_{n(k)}(t), \quad n \in \mathcal{N}(t), \tag{3.10}$$

$$\sum_{n \in \mathcal{N}_k(t)} w_{n(k)}(\Delta(t)) \leq W_k(\Delta(t)) =$$

$$= \sum_{m=1}^{M} p_k^m \cdot r_k^m(t) \cdot \Delta(t), \quad k = \overline{1, K}, \tag{3.11}$$

$$\sum_{k=1}^{K} r_k^m(t) \leq R^m(t), \quad m = \overline{1, M}, \tag{3.12}$$

$$r_k^m(t) \geq 0, \quad k = \overline{1, K}, \quad m = \overline{1, M}.$$

Содержательный смысл задачи (3.8) состоит в следующем: при заданных ограничениях (3.9) — (3.12) требуется просмотреть за время $\Delta(t)$ максимально возможное суммарное число шаров независимо от их цвета для всех урн z_n. Таким образом, решение задачи (3.8) при ограничениях (3.9) — (3.12) позволяет максимизировать суммарное число обработанных уникальных фрагментов данных на отрезке планирования $\Delta(t)$ для всех задач из $\mathcal{Z}(t)$.

3.3.2. Анализ ограничений ДСО

Рассмотрим достаточные условия завершения заданий z_j, $j \in \mathcal{N}(t)$, до наступления их директивных сроков окончания. Каждому $d_j, j \in \mathcal{N}(t)$, поставим в соответствие множество (список) номеров $\mathcal{N}(t, d_j)$ всех заданий $z_i, i \in \mathcal{N}(t)$, которые должны быть завершены не позднее времени $d_j, j \in \mathcal{N}(t)$, т.е.

$$\mathcal{N}(t, d_j) = \{i \mid d_i \leq d_j,\ i \in \mathcal{N}(t)\}, j \in \mathcal{N}(t). \qquad (3.13)$$

Для каждого $z_n, n \in \mathcal{N}(t)$ и каждого m в момент времени t определим переменную (параметр) управления $\nabla^m_{n(k)}(t)$ как число единиц времени (единичных временных интервалов), в течение которых задание z_n выполняется на всех работоспособных ЕВМ m-го типа, $m = \overline{1, M}$.

Если задача z_n не имеет решения, то для завершения оставшейся части $z^+_{n(k)}(t)$ необходимо просмотреть все оставшиеся в урне z_n шары. При этом

$$\mathbf{Z}_{n(k)} - z^-_{n(k)}(t) = \sum_{m=1}^{M} p^m_k \cdot R^m(t) \cdot \nabla^m_{n(k)}(t), \qquad (3.14)$$

$$n \in \mathcal{N}_k(t), k = \overline{1, K},$$

$$\nabla^m_{n(k)}(t) \geq 0, m = \overline{1, M}, n \in \mathcal{N}(t).$$

Рассмотрим величину

$$\nabla^m(t, d_j) = \sum_{k=1}^{K} \sum_{n \in \mathcal{N}(t,d_j) \bigcap \mathcal{N}_k(t)} \nabla^m_{n(k)}(t) =$$

$$= \sum_{n \in \mathcal{N}(t,d_j)} \nabla^m_{n(k)}(t), \quad m = \overline{1, M}, \quad j \in \mathcal{N}(t). \qquad (3.15)$$

Значение $\nabla^m(t, d_j)$ показывает, сколько единиц времени после момента t потребуется ЕВМ m-го типа для завершения всех заданий z_n, для которых $n \in \mathcal{N}(t, d_j)$, при условии, что ни одна из задач не имеет решения, т.е. ни в одной из урн z_n, $n \in \mathcal{N}(t, d_j)$ нет шара, соответствующего уникальному фрагменту данных. Для наихудшего случая $\nabla^m(t, d_j)$ является гарантированной оценкой временных затрат при использовании вычислительных ресурсов СВС. Достаточные условия соблюдения всех ДСО можно записать в виде неравенства

$$t + \nabla^m(t, d_j) \le d_j,\ j \in \mathcal{N}(t),\ m = \overline{1, M}. \tag{3.16}$$

Введем переменные

$$\omega^m(t, d_j) = \frac{t + \nabla^m(t, d_j) - t_j^0}{d_j - t_j^0},\ \ j \in \mathcal{N}(t),\ m = \overline{1, M}, \tag{3.17}$$

где t_j^0 — момент поступления z_j в СВС. Возможность соблюдения ДСО значение $\omega^m(t, d_j)$ характеризует следующим образом:

если $\omega_j^m(t, d_j) \le 1$ для всех $j \in \mathcal{N}(t, d_j), m = \overline{1, M}$, то все задания с индексами из $\mathcal{N}(t, d_j)$ гарантированно можно завершить до наступления их ДСО даже в наихудшем случае;

если же существует значение $\omega_j^m(t, d_j) > 1$, то хотя бы одно задание в худшем случае может быть завершено после назначенного срока.

Для проверки достаточных условий выполнения ДСО и поиска соответствующих $\nabla^m(t, d_j)$ в рамках М-модели решается следующая оптимизационная задача

при заданных $\mathbf{Z}_n, z_n^-(t), R^m(t), d_j, t_j^0$

найти

$$\omega^*(t) = \min_{\omega, \nabla} \omega(t) \tag{3.18}$$

при ограничениях (3.13)-(3.17) и

$$\omega^m(t, d_j) \le \omega(t), \quad m = \overline{1, M}, j \in \mathcal{N}(t). \qquad (3.19)$$

Оптимальное решение (3.18) обозначим через $< \omega^*(t), \nabla^{*m}_{n(k)}(t) >$. Если $\omega^*(t) \le 1$, то в момент времени t существует $\nabla^{*m}_{n(k)}(t)$ — стратегия выполнения заданий z_n (график использования ЕВМ), при которой ДСО не нарушаются. Полученные $\nabla^{*m}_{n(k)}(t)$ являются гарантированными оценками и показывают, сколько единиц времени будет затрачено ЕВМ m-го типа для выполнения задания $z_n, n \in \mathcal{N}(t, d_j)$ после момента t, если ни одна задача с индексом из $\mathcal{N}(t, d_j)$ не имеет решения, но при этом все задания гарантированно завершены в срок, до наступления d_j. Заметим, что решение (3.18) позволяет проверить существует ли в принципе для имеющегося множества $\mathcal{Z}(t)$ такое априорное распределение ресурсов, при котором будут соблюдены все ДСО. Ограничения (3.14) — (3.17), (3.19) используются далее в п. 3.4 для определения плана работ, или вектора $\mathbf{w}(t)$.

3.3.3. Показатель относительной задержки

Введем показатель относительной задержки выполнения разнородных заданий на момент времени $(t + \Delta(t))$ в гетерогенной СВС

$$\chi_n(t+\Delta(t)) = \frac{z_n^-(t) + w_n(\Delta(t))}{P_k(t)} \cdot \frac{1}{(T_n(t) + \Delta(t))}, \; n \in \mathcal{N}(t). \quad (3.20)$$

Следуя рассуждениям раздела 2, можно утверждать, что если поднабор $w_n(\Delta(t))$ содержит шар, соответствующий уникальному фрагменту, или урна z_n после извлечения $w_n(\Delta(t))$ окажется исчерпанной, то

$$\overline{\tau}_n(t) = \frac{z_n^-(t) + w_n(\Delta(t))}{P_k(t)}, n \in \mathcal{N}(t),$$

является оценкой сверху для абсолютного времени выполнения задания z_n в монопольном режиме. При этом фактически завершенное задание покинет СВС в момент $t_n^+ = t + \Delta(t)$, и, следовательно, длительность пребывания z_n в СВС составит $T_n^+ = t_n^+ - t_n^0 = T_n(t) + \Delta(t)$.

3.4. Параметризованная модель выполнения заданий

Рассмотрим М-модель выполнения заданий в СВС, в которой учитывается ДСО, на выбранном отрезке планирования определяется суммарное число шаров для обработки и минимизируется показатель относительной задержки выполнения. При формировании ПАКЕТ-ТР в контрольный момент времени t в ПК ПЛАН решается следующая

М-задача: при заданных $\mathbf{Z}_n, z_n^-(t), d_n, R(t), p_k^m, t_n^0, c_1 \geq 0$, $c_2 \geq 0, c_3 \geq 0$, найти:

$$\phi^*(t) = \max_{w,\nabla,r,\omega,\theta,\varphi} \left[c_1\theta(t) - c_2\omega(t) + c_3\varphi(t)\right], \tag{3.21}$$

при ограничениях:

$$\theta(t) \leq \chi_n(t + \Delta(t)), n \in \mathcal{N}(t), \tag{3.22}$$

$$\chi_n(t + \Delta(t)) = \frac{z_n^-(t) + w_n(\Delta(t))}{P_k(t)} \cdot \frac{1}{(T_n(t) + \Delta(t))}, \tag{3.23}$$

$$n \in \mathcal{N}_k(t), k = \overline{1, K},$$

$$\sum_{n \in \mathcal{N}_k(t)} w_n(\Delta(t)) = W_k(\Delta(t)),\ k = \overline{1, K}, \tag{3.24}$$

$$W_k(\Delta(t)) \leq \sum_{m=1}^{M} p_k^m \cdot r_k^m(t) \cdot \Delta(t),\ k = \overline{1, K}, \tag{3.25}$$

$$\sum_{k=1}^{K} r_k^m(t) \le R^m(t),\ m=\overline{1,M}, \tag{3.26}$$

$$0 \le w_n(\Delta(t)) \le \mathbf{Z}_n - z_n^-(t),\ n \in \mathcal{N}(t); \tag{3.27}$$

при ограничениях на эффективность использования ЕВМ (см. (3.9)):

$$\sum_{n\in\mathcal{N}(t)} w_n(\Delta(t)) \ge \varphi(t); \tag{3.28}$$

при ограничениях на ДСО, в соответствии с п. 3.3.2 достаточные условия выписываются для момента $t+\Delta(t)$:

$$\mathbf{Z}_n - z_n^-(t) - w_n(\Delta(t)) =$$

$$= \sum_{m=1}^{M} p_k^m \cdot R^m(t) \cdot \nabla_{n(k)}^m(t+\Delta(t)), \tag{3.29}$$

$$n \in \mathcal{N}_k(t),\quad k=\overline{1,K},$$

$$\nabla_{n(k)}^m(t+\Delta(t)) \ge 0,\ n\in\mathcal{N}(t),\ m=\overline{1,M},$$

$$\nabla^m(t+\Delta(t),d_j) =$$

$$= \sum_{n\in\mathcal{N}(t,d_j)} \nabla_{n(k)}^m(t+\Delta(t)),\quad m=\overline{1,M}, j\in\mathcal{N}(t). \tag{3.30}$$

$$\omega^m(t+\Delta(t),d_j) =$$

$$= \frac{t+\Delta(t)+\nabla^m(t+\Delta(t),d_j)-t_j^0}{d_j-t_j^0}, \tag{3.31}$$

$$j\in\mathcal{N}(t),\ m=\overline{1,M},$$

$$\omega^m(t+\Delta(t),d_j) \le \omega(t+\Delta(t)), \tag{3.32}$$

$$j\in\mathcal{N}(t),\ m=\overline{1,M}.$$

Решение М-задачи (3.21) при ограничениях (3.22)-(3.32) и различных, заранее заданных c_1, c_2, c_3 позволяет анализировать возможности управления в момент времени t.

Анализ достаточных условий выполнения ДСО

Положим в условиях М-задачи (3.21) $c_2 \gg c_1 = c_3$. Например, $c_2 = 10^6, c_1 = c_3 = 1$. Пусть $< \mathbf{w}^0(t), \omega^0(t), \varphi^0(t) >$ — ее оптимальное решение. Тогда:
— если $\omega^0(t + \Delta(t)) \leq 1$, то существует гарантированная стратегия управления выполнением заданий из $\mathcal{Z}(t)$, при которой все задания могут быть завершены до наступления своих ДСО, даже если ни одна задача $z_n, n \in \mathcal{N}(t)$, не имеет решения (наихудший случай);
— если $\omega^0(t+\Delta(t)) > 1$, то найденный вектор управлений $\mathbf{w}^0(t)$, т.е. состав ПАКЕТ-ТР на период $\Delta(t)$, является наилучшим решением, минимизирующим возможное относительное превышение некоторых ДСО на величину не более $\delta\omega = \omega^0(t + \Delta(t)) - 1$. В последнем случае следует взять вновь полученный вектор управлений $\mathbf{w}^0(t)$, сформировать соответствующий ПАКЕТ-ТР и приступить к его выполнению. Действительно, для некоторых $z_n, n \in \mathcal{N}(t)$ существует опасность не уложиться в ДСО, и поэтому все ресурсы должны быть направлены на выполнение таких заданий. Список задач и число шаров, которые необходимо срочно обработать, содержатся в векторе $\mathbf{w}^0(t)$, т.е. состав аварийного ПАКЕТ-ТР уже готов, поскольку получен в ходе решения (3.21) при $c_1 = 1$, $c_2 = 10^6, c_3 = 1$.

Анализ производительности

Пусть $\omega^0(t + \Delta(t)) \leq 1$. В условиях (3.32) зафиксируем значения $\omega(t + \Delta(t)) = 1$, и положим $c_3 \ggg c_1 > 0, c_2 = 0$, и вновь решим задачу (3.21). Полученное решение обозначим $< \mathbf{w}^*(t), \varphi^*(t) >$. Значение $\varphi^*(t)$ показывает, какое максимальное число СЭВО всех видов может быть выполнено за период

$\Delta(t)$ для всех $z_n, n \in \mathcal{N}(t)$, при условии, что ограничения на ДСО соблюдаются.

Оптимизация показателей задержки

Как и ранее, зафиксируем значение $\omega(t+\Delta(t)) = 1$ в (3.32), а также положим $\varphi(t) = \varphi^*(t)$ в (3.28), $c_1 = 10^6$, и решим (3.21) с этими значениями. Обозначим полученное оптимальное решение $< \theta^{**}(t), \mathbf{w}^{**}(t) >$. Можно утверждать, что

— для всех $z_n, n \in \mathcal{N}(t)$ показатель относительной задержки выполнения будет не меньше $\theta^{**}(t)$;

— выполнение всех задач $z_n, n \in \mathcal{N}(t)$ может быть завершено до наступления предписанных для них ДСО;

— для данного набора $\mathcal{Z}(t)$ достигается максимальная суммарная производительность СВС на интервале $\Delta(t)$.

Заключение

Следует обратить внимание, что для рассматриваемого специального класса citu-заданий проблема управления их выполнением по сути является многокритериальной. Действительно, от диспетчера-планировщика на каждом этапе требуется:

— эффективно использовать разнотипные ЕВМ и добиваться максимальной производительности СВС;

— завершать каждое конкретное задание до наступления ДСО;

— в условиях неопределенности распределять вычислительные ресурсы таким образом, чтобы при выполнении ни одно из *равнозначных* заданий *равноправных* пользователей не оказалось в худшей ситуации, чем остальные.

Если исходную проблему со случайными, априори неизвестными параметрами формализовать как задачу многокритериальной оптимизации, то могут возникнуть серьезные методологические трудности с определением понятия решений и, соответственно, с их последующей интерпретацией [21]-[24].

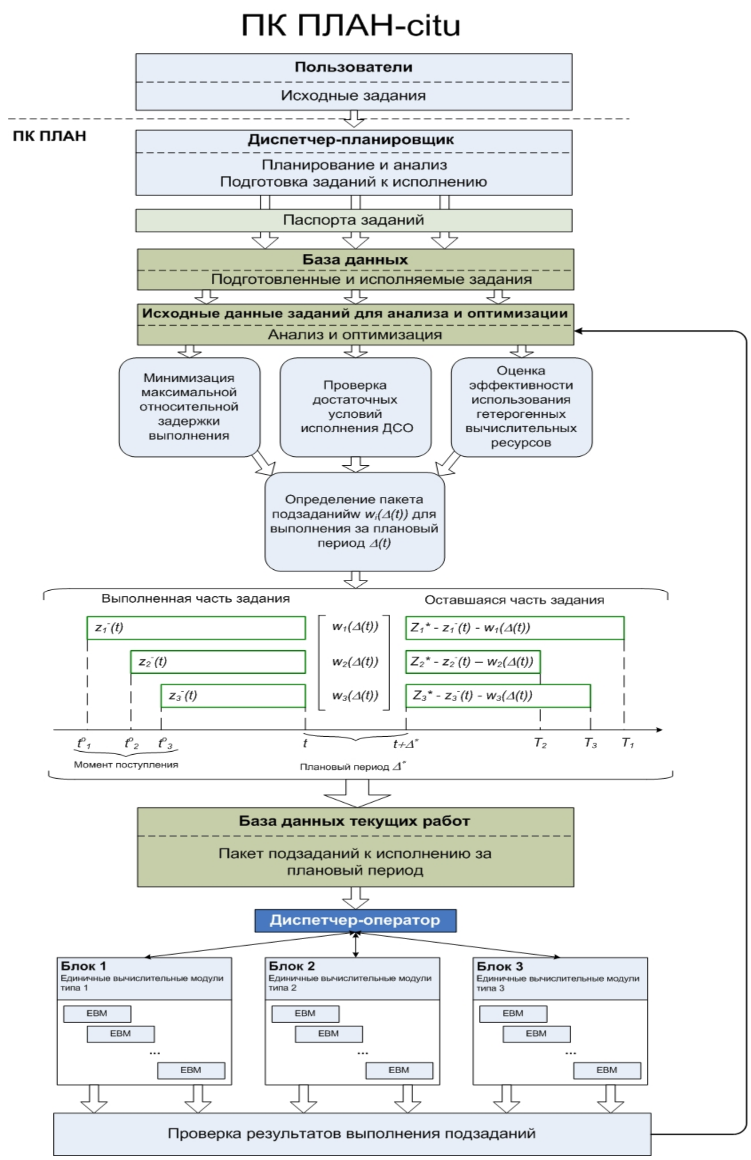
В рамках М-модели процесс управления динамической системой разбивается на отдельные этапы (см. рисунок), на каждом из которых:

— прежде всего строятся гарантированные оценки для целого ряда величин, которые изначально неизвестны;

— на основании гарантированных оценок находится вектор управления $\mathbf{w}^*(t)$, определяющий недискриминирующее распределение вычислительных ресурсов в условиях неопределенности;

— достигается максимальная эффективность использования СВС;

— для каждой задачи происходит проверка достаточных условий выполнения ДСО. В случае нарушения ограничений



ПК ПЛАН-citu
Пользователи
Исходные задания
ПК ПЛАН
Диспетчер-планировщик
Планирование и анализ
Подготовка заданий к исполнению
Паспорта заданий
База данных
Подготовленные и исполняемые задания
Исходные данные заданий для анализа и оптимизации
Анализ и оптимизация
Минимизация максимальной относительной задержки выполнения
Проверка достаточных условий исполнения ДСО
Оценка эффективности использования гетерогенных вычислительных ресурсов
Определение пакета подзаданийw $w_i(\Delta(t))$ для выполнения за плановый период $\Delta(t)$
Выполненная часть задания
Оставшаяся часть задания
Момент поступления
Плановый период Δ^*
База данных текущих работ
Пакет подзаданий к исполнению за плановый период
Диспетчер-оператор
Блок 1
Единичные вычислительные модули типа 1
Блок 2
Единичные вычислительные модули типа 2
Блок 3
Единичные вычислительные модули типа 3
ЕВМ
Проверка результатов выполнения подзаданий

осуществляется релейное переключение СВС на обработку задач, для которых срок завершения может быть нарушен.

В результате в первую очередь решаются короткие задачи, с небольшим абсолютным временем, а при пакетной обработке все задания, не содержащие уникальных фрагментов, завершаются в последнюю очередь.

Литература

1. Коновалов М.Г., Малашенко Ю.Е., Назарова И.А. Управление заданиями в гетерогенных вычислительных системах // Изв. РАН. ТиСУ. 2011. № 2. С. 72–90.

2. Коновалов М.Г., Малашенко Ю.Е., Назарова И.А. Модели и методы управление заданиями в системах распоеделенных вычислительных ресурсов. М.: ВЦ РАН, 2009.

3. Каляев И.А., Левин И.И. Реконфигурируемые мультиконвейерные вычислительные структуры для решения потоковых задач обработки информации и управления // Тр. междунар. научно-практической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный ун-т, 2010. Т. 1. С.100-102.

4. Забродин А.В., Левин В.К., Корнеев В.В. Массово-параллельные системы МВС-100 и МВС-1000. // Сб. научн. тр. научной сессии МИФИ - 2000. М.: МИФИ, 2000. Т. 2. С. 194-195.

5. Баранов А.В., Киселев А.В., Корнеев В.В. и др. Управление сетевой средой распределенных вычислений // Тр. Первой Всероссийской научн. конф. "Методы и средства обработки информации". М.: МГУ им. М.В. Ломоносова, 2003. С. 98-103.

6. Баранов А.В., Киселев А.В., Корнеев В.В. и др. Программный комплекс "Пирамида"организации параллельных вычислений с распараллеливанием по данным // Тр. междунар. суперкомпьютерной конф. и конф. молодых ученых "Научный сервис в сети Интернет: Суперкомпьютерные центры и задачи". М.: МГУ им. М.В. Ломоносова, 2010. С. 299-302.

7. Garonne V., Stokes-Rees I., Tsaregorodsev A. DIRAC: A Scalable Lightweight Architecture for High Throughput Computing// Proc. 5th IEEE/ACM Int. Worksh. on Grid Computing. IEEE Comp. Sc.: Washington, 2004. P. 19 - 25.

8. Garonne V., Tsaregorodtsev A., Caron E. A study of meta-scheduling architectures for high throughput computing: Pull versus Push // Proc. 4th Int. Symp. on Parallel and Distribut. Computing. IEEE Comp. Sc.: Washington, 2005. P. 226-233.

9. Dongarra J., Foster I., etc. edit. Sourcebook of parallel computing. San Francisco: Morgan Kaufmann Publ. Inc., 2003.

10. Casavant T., Kuhl J. Taxonomy of Scheduling in General Purpose Distributed Computing Systems // IEEE Transactions on Software Engineering. 1988. V. 14. N. 2. P. 141-154.

11. Matarneh R. Self-Adjustment Time Quantum in Round Ro-

bin Algorithm Depending on Burst Time of the Now Running Processes // Am. J. Applied Sci., 2009. V. 10. N. 6. P. 1831-1837.

12. Голосов П.С., Козлов М.В., Малашенко Ю.Е.и др. Анализ управления заданиями специализированными вычислительными заданиями в условиях неопределенности // Изв. РАН. ТиСУ. 2011. № 6. С. 63-79.

13. Голосов П.С., Козлов М.В., Малашенко Ю.Е.и др. Модель системы управления специализированным вычислительным комплексом. М.: ВЦ РАН, 2010.

14. Козлов М.В., Малашенко Ю.Е., Назарова И.А. и др. Анализ режимов управления вычислительным комплексом в условиях неопределенности. М.: ВЦ РАН, 2011.

15. Ронжин А.В., Суриков В.Н. О математических проблемах применения высокопроизводительных вычислительных систем для решения задач случайного поиска // Тр. междунар. научно-технической конфю "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный ун-т , 2010. Т. 2. С. 239-243.

16. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. - М.: Наука, 1986.

17. Гермейер Ю.Б. Введение в теорию исследования операций. операций. М.: Наука, 1971.

18. Rasmussen R.V., Trick M.A. Round Robin Scheduling - A

Survey // European Journal of Operational Research. 2008. V. 188. Iss. 3. P. 617-636.

19. Pinedo M. Scheduling: Theory, Algorithms, and Systems. 3rd ed. Heidelberg: Springer Verlag, 2002.

20. Феллер В. Введение в теорию вероятностей и её приложения. М.: Мир, 1984. Т. 1.

21. Li C., Li L. Utility-based QoS optimisation strategy for multi-criteria scheduling on the grid // J. of parallel and distributed computing. 2007. V. 67. N. 2. P. 142-153.

22. Salmani V., Ensafi R., Khatib-Astaneh N. et. al. A Fuzzy-Based Multi-criteria Scheduler for Uniform Multiprocessor Real-Time Systems // Proc. 10th Int. Conf. on Information Technology. IEEE Comp. Sc. Press, 2007. P. 179-184.

23. Salmani V., Naghibzadeh M., Kahani M. et al. Multi-criteria Scheduling of Soft Real-time Tasks on Uniform Multiprocessors Using Fuzzy Inference // Advances and Innovations in Syst. Comp. Sciences and Software Engineering. Springer, 2007. P. 439-444.

24. Baraglia R., Klusaček D., Rudova H. et al. Comparison Of Multi-Criteria Scheduling Techniques // Grid Computing. Springer, 2008. P. 173-184.

СОДЕРЖАНИЕ

Список литературы

[1] Коновалов М.Г., Малашенко Ю.Е., Назарова И.А. Управление заданиями в гетерогенных вычислительных системах // Изв. РАН. ТиСУ. 2011. № 2. С. 72–90.

[2] Коновалов М.Г., Малашенко Ю.Е., Назарова И.А. Модели и методы управление заданиями в системах распоеделенных вычислительных ресурсов. М.: ВЦ РАН, 2009.

[3] Каляев И.А., Левин И.И. Реконфигурируемые мультиконвейерные вычислительные структуры для решения потоковых задач обработки информации и управления // Труды международной научно-практической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 1. С.100-102.

[4] Забродин А.В., Левин В.К., Корнеев В.В. Массово-параллельные системы МВС-100 и МВС-1000. // Сборник научных трудов научной сессии МИФИ - 200. М.: МИФИ, 2000. Т. 2. С. 194-195.

[5] Баранов А.В., Киселев А.В., Корнеев В.В. и др. Управление сетевой средой распределенных вычислений // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации". М.: МГУ им. М.В. Ломоносова, 2003. С. 98-103.

[6] Баранов А.В., Киселев А.В., Корнеев В.В. и др. Программный комплекс "Пирамида"организации параллельных вычислений с распараллеливанием по данным // Труды международной суперкомпьютерной конференции и конференции молодых ученых "Научный сервис в сети Интернет:

Суперкомпьютерные центры и задачи". М.: МГУ им. М.В. Ломоносова, 2010. С. 299-302.

[7] Garonne V., Stokes-Rees I., Tsaregorodsev A. DIRAC: A Scalable Lightweight Architecture for High Throughput Computing // Proc. 5th IEEE/ACM Int. Workshop on Grid Computing. IEEE Computer Society, Washington, 2004. P. 19 - 25.

[8] Garonne V., Tsaregorodtsev A., Caron E. A study of meta-scheduling architectures for high throughput computing: Pull versus Push // Proc. 4th Int. Symp. on Parallel and Distributed Computing. IEEE Computer Society, Washington, 2005. P. 226-233.

[9] Dongarra J., Foster I., etc. edit. Sourcebook of parallel computing. San Francisco: Morgan Kaufmann Publishers Inc., 2003.

[10] Casavant T., Kuhl J. Taxonomy of Scheduling in General Purpose Distributed Computing Systems // IEEE Transactions on Software Engineering. 1988. V. 14. N. 2. P. 141-154.

[11] Matarneh R. Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes // Am. J. Applied Sci., 2009. V. 10. N. 6. P. 1831-1837.

[12] Голосов П.Е., Козлов М.В., Малашенко Ю.Е. и др. Анализ управления заданиями специализированными вычислительными заданиями в условиях неопределенности // Изв. РАН. ТиСУ. 2011. № 6. С. 63-79.

[13] Голосов П.С., Козлов М.В., Малашенко Ю.Е. и др. Модель системы управления специализированным вычислительным комплексом. М.: ВЦ РАН, 2010.

[14] Козлов М.В., Малашенко Ю.Е., Назарова И.А. и др. Анализ режимов управления вычислительным комплексом в условиях неопределенности. М.: ВЦ РАН, 2011.

[15] Ронжин А.В., Суриков В.Н. О математических проблемах применения высокопроизводительных вычислительных систем для решения задач случайного поиска // Труды международной научно-технической конференции "Суперкомпьютерные технологии: разработка, программирование, применение". Ростов-на-Дону: Южный федеральный университет, 2010. Т. 2. С. 239-243.

[16] Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. - М.: Наука, 1986.

[17] Гермейер Ю.Б. Введение в теорию исследования операций. М.: Наука, 1971.

[18] Rasmussen R.V., Trick M.A. Round Robin Scheduling - A Survey // European Journal of Operational Research. 2008. V. 188. Iss. 3. P. 617-636.

[19] Pinedo M. Scheduling: Theory, Algorithms, and Systems. 3rd ed. Heidelberg: Springer Verlag, 2002.

[20] Феллер В. Введение в теорию вероятностей и ее приложения. М.: Мир, 1984. Т. 1.

[21] Li C., Li L. Utility-based QoS optimisation strategy for multicriteria scheduling on the grid // J. of parallel and distributed computing. 2007. V. 67. N. 2. P. 142-153.

[22] Salmani V., Ensafi R., Khatib-Astaneh N. et. al. A Fuzzy-Based Multi-criteria Scheduler for Uniform Multiprocessor Real-Time Systems // Proc. 10th Int. Conf. on Information Technology, (ICIT 2007). IEEE Comp. Sc. Press, 2007. P. 179-184.

[23] Salmani V., Naghibzadeh M., Kahani M. et al. Multi-criteria Scheduling of Soft Real-time Tasks on Uniform Multiprocessors Using Fuzzy Inference // Advances and Innovations in Systems, Computing Sciences and Software Engineering. Springer Netherlands, 2007. P. 439-444.

[24] Klusáček D., Rudová H., Baraglia R. et al. Comparison Of Multi-Criteria Scheduling Techniques // Grid Computing. Springer, 2008. P. 173-184.