ГЛАВА 5

МОДЕЛЬ ПЛАНИРОВАНИЯ В НЕОДНОРОДНЫХ Р С О Д

ПРЕДИСЛОВИЕ

Повышение производительности вычислительных систем является предметом пристального внимания исследователей на протяжении всего развития средств вычислительной техники.

Эти исследования можно условно разделить на два основных направления:

Повышение производительности вычислительной системы за счет технического совершенствования самих вычислительных средств:

* быстродействия процессора;
* структурной организации ресурсов вычислительной системы;
* создания многопроцессорных параллельных вычислительных систем;
* создания распределенных систем обработки информации на базе вычислительных сетей.

Повышение производительности вычислительной системы за счет оптимальной организации, планирования и диспетчеризации работ в вычислительной системе:

* реализации истинного или кажущегося совмещения работ на однородных или неоднородных ресурсах однопроцессорной вычислительной системы;
* организации параллельного выполнения отдельных частей программ;
* эффективного использования ресурсов в параллельной вычислительной системе при многопрограммном режиме работы с распараллеливанием работ и программ;
* организации эффективных процедур распределенных операционных систем при динамической организации вычислительных процессов в вычислительных сетях любого класса.

Во втором направлении наиболее важными являются два подхода к повышению эффективности организации вычислительных процессов: *распараллеливание* и *распределение* (*диспетчеризация*) процессов. Известно множество работ по теории распараллеливания и практически работающих систем распараллеливания алгоритмов и программ, позволяющих выполнить предварительную подготовку работ, реализация которых требует многопрограммного режима работы однопроцесорной, многопроцессорной или распределенной вычислительной системы. Однако исследования в области теории построения эффективных систем планирования множества работ в вычислительной среде еще не дали практических алгоритмов и программ, позволяющих с малыми накладными расходами применять их для создания операционных систем многопроцессорных вычислительных средств. Практически отсутствует программная интерфейсная среда работы пользователя с многопроцессорными рабочими станциями, имеющими небольшое число процессорных элементов (10 -20). Отсутствуют системы автоматической подготовки параллельных программ для работы в параллельной вычислительной системе. Широкое применение таких систем — как PVM, MPI, Express, P4, Linda, Panda, сдерживается достаточно большой долей высококвалифицированного ручного труда при подготовке программ для исполнения в параллельной ВС. Кроме этого, используемые в них планировщики не гарантируют оптимального или оптимизированного использования оборудования. Также практически отсутствуют эффективные средства, позволяющие программисту проверить или смоделировать погружение распараллеленной программы в вычислительную среду высокопараллельных систем.

Анализ тенденций развития средств связи (коммуникаций) и применения их для параллельной распределенной обработки информации позволяет определить тенденции в изменении подходов к организации вычислительного процесса в таких системах. Все более широко для связи вычислительных машин применяются высокоскоростные средства связи, такие как: FDDI (100Мбит/сек), HiPP (800Мбит/сек/ 1.61Гбит/сек), SONET (OC-3—155.521Мбит/сек, OC-12 — 622.081бит/сек. Вычислительную систему можно рассматривать как совокупность неоднородных ресурсов с различной степенью детализации, а объекты планирования вычислений или заявки на ресурсы могут быть представлены как программы, процедуры, параллельные участки программ и т.д.

Существуют глобальные системы, объединяющие различные мощные удаленные вычислительные машины для эффективной и экономной обработки различных задач. Для таких распределенных систем при решении задач планирования и диспетчеризации целесообразно введение новой трактовки понятия неоднородности. Вычислительные системы с точки зрения управления и организации вычислительных процессов в них можно разделить на *"полунеоднородные"*, "*неоднородные*" и *"строго—неоднородные"*. Примерами таких ВС являются PVM [84, 91] и SmartNet системы [80].

5.1. ХАРАКТЕРИСТИКА РЕСУРСОВ И ЗАДАЧ В НЕОДНОРОДНЫХ РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

5.1.1. Характеристика неоднородности НРС

Понятие о строго неоднородных распределенных системах обработки данных (СНРС) является новым понятием. Это понятие введено в работах [125, 126] по исследованию задач планирования в параллельных многомашинных системах, которые имеют две особенности — это *распределенность* обработки информации и характеристику *неоднородности* не только для ресурсов, но и так же для заданий. В задаче планирования для таких систем нельзя не учитывать время передачи данных (из-за распределенности), а также проблему выбора оптимального расписания из различных вариантов (из-за неоднородности) [125, 126]. Этот фактор еще больше усложняет задачу планирования и поиска оптимизированного решения по сравнению с задачей планирования в однородных системах массового распараллеливания, которые относятся к классу NP-полных [10].

- *Общая постановка задачи динамического планирования в НРС*

Вычислительная система состоит из **N** различных ресурсов. На момент выполнения процедур планирования **t,** в системе имеются **R** независимых (распараллеленных) заданий. Каждое назначение задания на ресурс должно учитывать время выполнения $\mathbf{T_e}$ задания (а они независимы в момент t) и время коммуникации $\mathbf{T_c}$ (зависит от другого назначения, выполненного на предыдущем шаге t-1). Нужно составить расписание выполнения этих заданий на имеющихся ресурсах так, чтобы оно было оптимальным, где критерий оптимизации может быть один или несколько: минимальное время чистого выполнения заданий (Scheduling Length) имеющимися ресурсами, минимальная сумма времени коммуникации, максимальное использование ресурсов.

Чтобы разработать стратегию планирования для данной модели РСОД, надо точно определить 2 ключевых свойства системы: *тип* (или степень) *неоднородности* и *динамичность* (или реконфигурируемость) системы. Рассмотрим их влияние на требования к диспетчеру, реализующему функции планирования.

Понятие "неоднородность" системы можно рассматривать для трех случаев:

- В полунеоднородной системе, когда или задания, или ресурсы являются однородными или неоднородными. Число различных расписаний (до оптимизации) равно размещению из R элементов по N без повторения и без перестановки, то есть число возможных вариантов равно $\frac{R!}{(R-N)!}$ [21, 59, 134].
- В неоднородной системе и задания, и ресурсы являются неоднородными, и любое назначение задание—ресурс всегда является возможным (лишь имеют разные значения $\mathbf{T_e}$ и $\mathbf{T_c}$). Число различных расписаний (до оптимизации) равно размещению из R элементов по N без повторения, но с перестановками, тогда число возможных вариантов равно $\frac{R!N!}{(R-N)!}$ [72].
- В строго неоднородной системе имеет место понятие совместимости ресурса—задания, при этом некоторые назначения ресурс—задание могут быть *невозможными*. Поэтому планирование для таких систем состоит из двух этапов: на первом этапе ищутся расписания возможных назначений максимальной мощности, а на втором этапе требуется найти оптимальное расписание из найденных на предыдущем этапе. Число возможных вариантов на первом этапе равно число различных и возможных расписаний (до

оптимизации), или равно **числу** *максимальных паросочетаний* двух множеств **N** и **R** $V_1 \leq \frac{R!N!}{(R-N)!}$ [13, 40]**.** Число возможных вариантов на втором этапе $V_2 \leq V_1$.

Таким образом, число возможных вариантов в строго неоднородной системе, меньше чем в системах другого вида, но решение задачи планирования в данном случае усложняется из-за возможности появления конфликтных ситуаций, т.к. при этом следует искать паросочетание максимальной мощности с максимальным значением коэффициентов оптимизации.

Очевидно, что 2 первых варианта являются частным случаем последнего. Это означает, что алгоритмы, разработанные для строго неоднородных систем, будут работать в остальных случаях, хотя и не так эффективно, как специализированные.

5.1.2. Динамичность РСОД

Здесь РСОД понимается как общая система организации обслуживания **N** заданий на **R** ресурсах, а не только как обычная система ресурсов. Поэтому следует учитывать как свойства *динамичности* ресурсов, так и *динамичности* заданий.

* *Динамичность ресурсов*

Система ресурсов может быть фиксированной или реконфигурируемой.

- В том случае, когда РСОД является ***фиксированной,*** то число вычислительных узлов и связей между ними являются неизменными, метод планирования обычно является статическим и можно использовать, так называемую, технику моделирования процесса методом Отжега — SA (SA — Simulation Annealing) [51, 101]. Эта схема планирования требует полного и точного знания о состоянии системы (ресурсах), а также о заданиях до начала планирования. Она не может реагировать ни на какое изменение системы, включая отказ ресурсов (узлов или каналов связи).
- Однако, на практике ***распределенная*** обработка часто сопровождается необходимостью ***реконфигурации*** системы, а именно, способностью системы реагировать на отказ узла или отдельного ресурса от обработки задания (отказ, связанный не только с невозможностью функционирования, но и с временным отключением вычислительного ресурса от системы планирования при выполнении другой работы на момент решения задачи планирования). В этом случае для обеспечения живучести ВС, безопасности выполнения заданий и эффективной организации работы ресурсов, планировщик должен быть способен на обработку новой поступающей информации о состоянии системы и разработку *нового расписания*. Таким образом, используемый метод планирования в такой системе должен быть динамическим. Однако для динамического планирования нельзя принимать технику SA из-за неприемлемой временной сложности. В таких системах более важным является время планирования, а требования к качеству планирования несколько

снижаются.

* *Динамичность заданий*

Здесь под динамичностью заданий понимается то, как они поступают в систему, как выполняются и как удаляются из системы. Именно эти характеристики заданий и определяют вид управления при организации вычислений. Различают 3 вида управления: синхронное [49], асинхронное [115] и в реальном времени [146], хотя третий вид является частным случаем второго.

- В *синхронной* организации вся система работает по тактам, которых может быть несколько типов: макро-такт для заданий и микро-такт для задач. Новые задачи поступают в начале каждого микро-такта. Все последующие действия процесса планирования и самого вычисления выполняются по макро-тактам, то есть каждое действие системы управления и планирования воздействует на соответствующие объекты (ресурсы) в ***одно и то же время*** по завершению макро-такта, и заканчиваются одновременно. Это означает, что в таких неоднородных системах имеет место *простой ресурсов,* так как задания требуют разного времени выполнения на разных ресурсах. В таких системах требуется решение задачи *балансирования* загрузки при создании расписания. Однако схема планирования при этом проще, по сравнению с асинхронной организацией и, кроме того, имеется возможность применять статический подход планирования, в котором допускаются и медленные алгоритмы.
- В *асинхронной* организации время планирования измеряется квантами (1 Квант = время одного шага планирования). Состояние системы считается неизменным в течении одного кванта. В асинхронной организации каждое планирующее действие системы воздействует на соответствующие объекты, участвующие в планировании в ***разное время*** (кванты) в зависимости от момента окончания предыдущего действия на том же объекте. Наличие свободного ресурса в системе активизирует действие планировщика, который анализирует очередь заданий и выбирает задание для загрузки на освободившийся ресурс. *Простой ресурсов* в этом случае уменьшается, т.к. ресурс завершивший выполнение задачи активизирует планировщик для получения очередного задания. Схема планирования в этом случае более сложная (по сравнению с синхронной организацией), так как планирование должно быть динамическим, хотя и не требуется решения задачи *балансирования* загрузки. При этом можно применять только быстрые алгоритмы. Время решения задачи планирования должно быть меньше 1 кванта.
- При управлении в *реальном времени* управление вычислениями является аналогом асинхронного типа. Разница между ними заключается в наличии временных ограничений на времена поступления и завершения для каждой задачи в системе реального времени. Поэтому главным критерием оптимизации является обеспечение выполнения этих временных ограничений. Для этого вида работ планирование заданий часто выполняется по приоритетной схеме.

Планирование в таких системах требует специализированных алгоритмов.

Из вышесказанного, очевидно, что синхронная огранизация удовлетворяет фиксированные РСОД, а реконфигурируемые РСОД требуют асинхронной организации.

5.2. ОБЩАЯ МОДЕЛЬ НЕОДНОРОДНОЙ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ

В распределенной неоднородной реконфигурируемой системе обработки данных (РНРС), имеются **N** узлов—ресурсов, образующих множество ресурсов $\mathbf{SR=\{V_1,..,V_N\}}$. Связи между этими узлами характеризуются матрицей связности ресурсов (МСР) **MCR[1..N, 1..N]**, где MCR[**i,j**] есть вес связи между узлами $\mathbf{V_i}$ и $\mathbf{V_j}$, и MCR[**i,j**]$\in\Re^+$ (множество реальных чисел) рис. 5.1. Мы говорим "нет связи" или связь неприемлема между $\mathbf{V_i}$ **и** $\mathbf{V_j}$,когда MCR[**i,j**] больше некоторого заданного числа $\mathbf{\Omega_0}$. В системе имеются **Q** задач, образующих множество задач $\mathbf{ST=\{T_1,..,T_Q\}}$. Эти задачи (процессы) могут поступать извне или активизироваться самой ВС. Каждая исходная задача $\mathbf{T_i}$ имеет свои характеристики и может представляться некоторым распараллеленным графом. Все значения N, Q, MCR являются динамическими.

Данная система характеризуется, кроме *общей модели,* еще следующими свойствами:

- распределенная.
- неоднородная.
- реконфигурируемая.
- асинхронная.
- с централизованным управлением.

5.2.1. Требования к планировщику в НРС

Соответственно высшеописанным характеристикам, можно сформулировать требования к планировщику НРС:

- Он должен быть *динамичным* из-за свойства реконфигурируемости системы. Соответственно, время перепланирования (реконфигурации) или реакции планировщика должно быть меньше периода времени, за которое система должна реконфигурироваться еще раз.
- Критериями *оптимизации* являются:
    1. *Максимальная степень использования ресурсов*, то есть надо учитывать совместимость ресурсов—заданий так, чтобы максимально эффективно использовать все ресурсы системы в целом (из-за неоднородности системы).
    2. *Минимизация коммуникационных затрат*, то есть следует учитывать число и время передачи данных между заданиями, выполняющимися на разных вычислительных узлах ВС.

5.2.2. Схема и стратегия планирования НРС

На основе общей модели и требований к системе планирования, можно представить схему планирования для НРС (рис. 5.2), в соответствии с семиуровневой моделью [39], описанной выше.

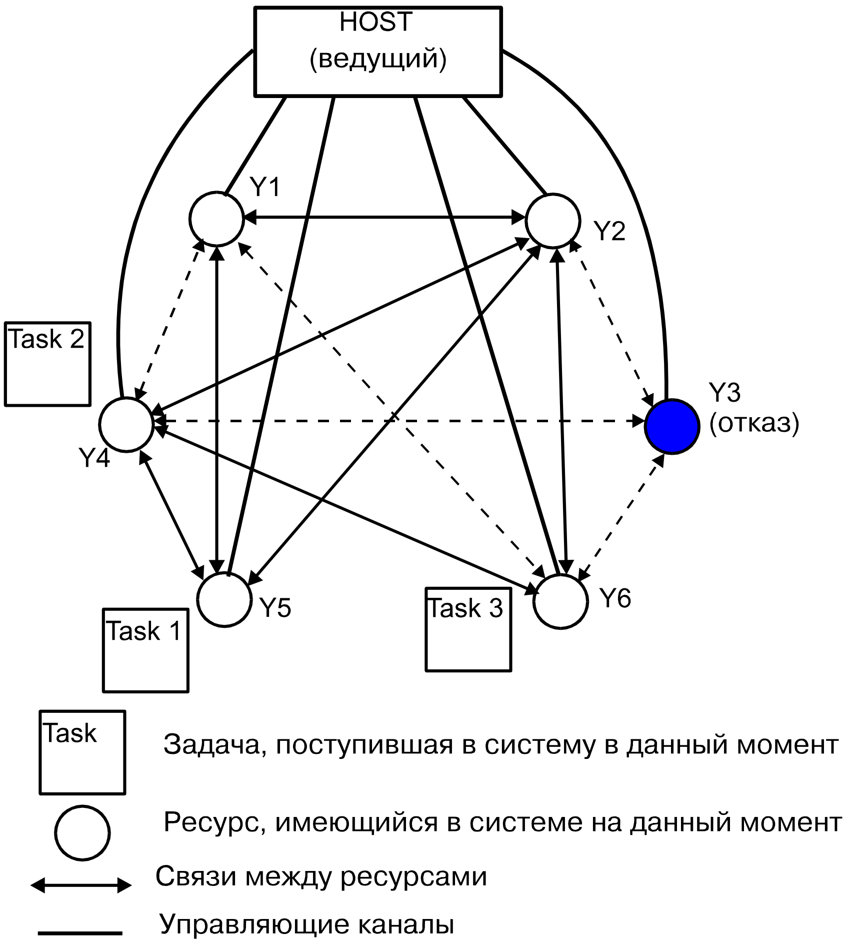


Рис. 5.1. Общая схема ВС

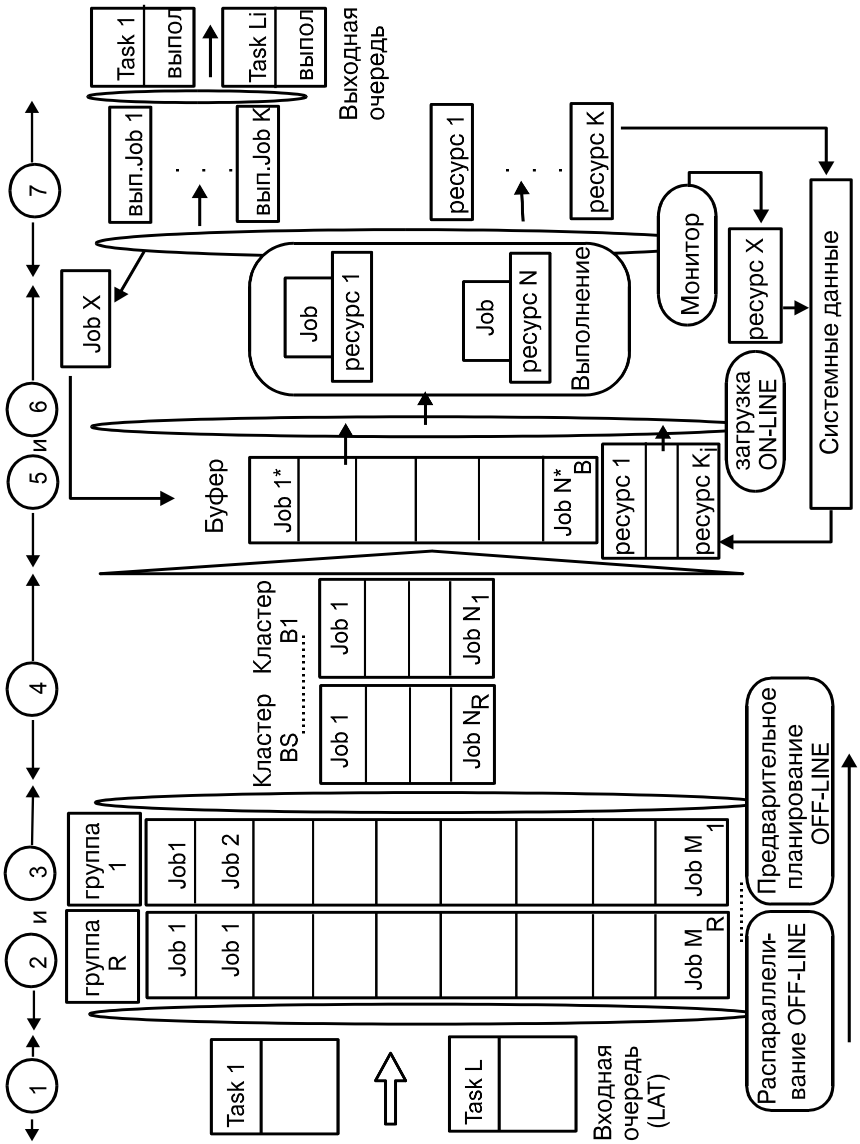


Рис. 5.2. Общая схема планирования для НРС

1. ***Ввод задач:*** прием заявок на выполнение задач $\mathbf{T_1,..,T_Q}$, поступающих из вычислительных узлов или на вход системы. Формирование входной очереди задач системы LAT (рис. 5.2.).

2. ***Анализ:*** предварительно анализируются требования задач $\mathbf{T_1,..,T_Q}$ и устанавливаются нужные параметры для дальнейшего планирования. Подготавливаются данные для распараллеливания задач на следующем шаге.

3. ***Распараллеливание*** задач $\mathbf{T_1,..,T_Q}$ (например для 3-х задач $\mathbf{T_1,T_2,T_3}$) на максимально параллельное выполнение заданий $\mathbf{J_{i,j}}$ (без ограничений на количество ресурсов). После распараллеливания задачи представляются в виде ориентированных ациклических графов $\mathbf{VT_1,..,VT_Q}$, где $\mathbf{VT_1=\{J_{1,1},..,J_{1,L1}\},..,VT_Q=\{J_{Q,1},..,J_{Q,LQ}\}}$. Все $\mathbf{J_{i,j}}$ каждого графа $\mathbf{VT_i}$ разделяются на $\mathbf{S_i}$ кластеров по степени предшествования. Пример кластеризации для 3-х задач $\mathbf{T_1}$, $\mathbf{T_2}$, $\mathbf{T_3}$ показан на рис. 5.3.

4. ***Адаптация:*** в каждый момент времени τ можно выделить "семью" задач T_i^τ, где после распараллеливания они представляют множество кластеров как показано на рис. 5.4.

   Эти вновь полученные кластеры присоединяются к текущим кластерам $\mathbf{B1,B2,..,BS\tau}$, причем все задания из **1-ых** кластеров (всех задач в LAT) — к кластеру **B1,** все задания из **2-ых** кластеров (всех задач в LAT) — к кластеру **B2,** и т.д.: $\mathbf{B1=\{J^{1,1},..,J^{1,M1}\},..,BS_\tau=\{J^{S\tau,1},..,J^{S\tau,MS\tau}\}}$.

   В том случае, если в данном кванте планирования количество заданий, готовых к выполнению намного больше числа свободных ресурсов, выполняется ***предварительная оптимизация***, результатом которой является *маркирование* заданий μ_{ij}. Например, для минимизации пересылок маркируются задания, находящиеся на критическом пути, так, чтобы потом обеспечить их предпочтительное выполнение на одном и том же процессоре, если это возможно и не влияет (не увеличивает временные затраты) на другие назначения.

   Задания из кластера **B1** могут поступить в буфер (где находятся задания готовые для выполнения) через фильтр только тогда, когда все **его** предшествующие задания (результаты которых являются нужными для данного задания) выполнены. По такому же принципу задания из кластера $\mathbf{B_i}$ сдвигаются в кластер $\mathbf{B_{i-1}}$ (рис. 5.5). Внутренние задания каждого кластера и, особенно, буфера всегда являются абсолютно независимыми друг от друга.

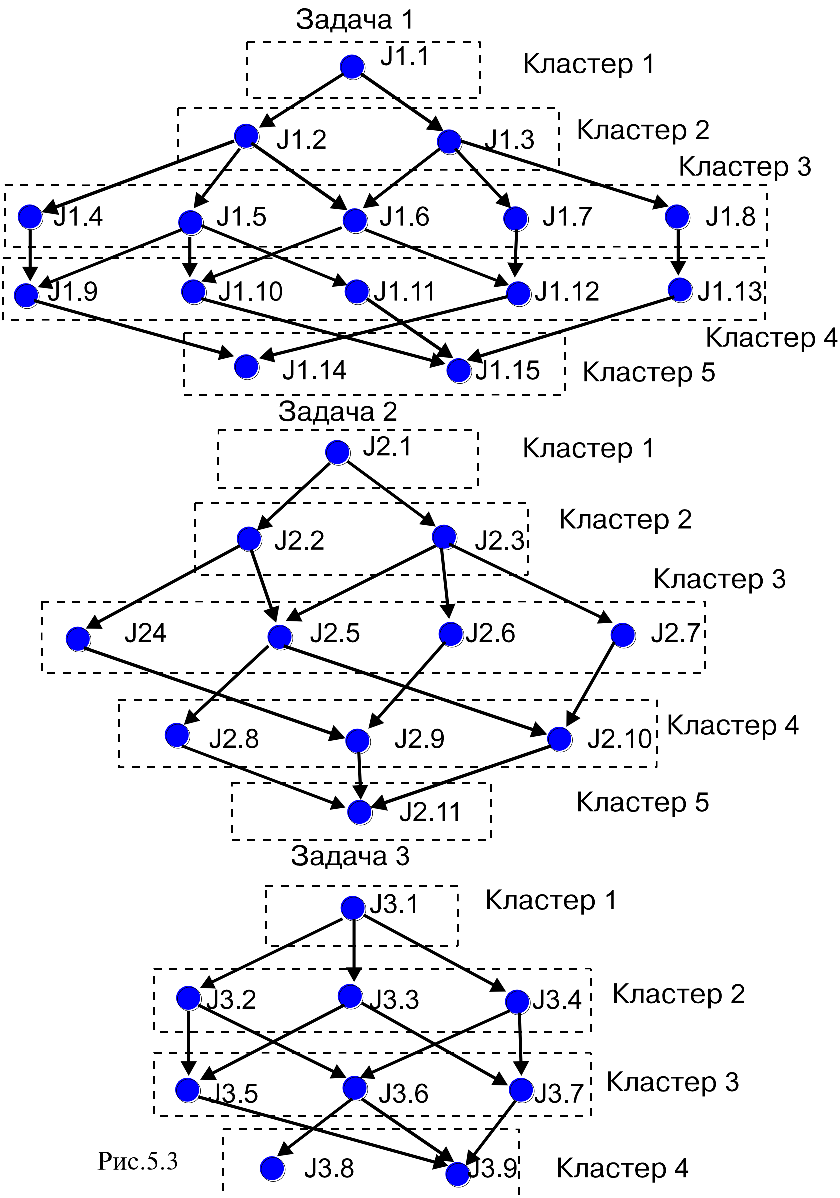

Задача 1
J1.1
Кластер 1
J1.2
J1.3
Кластер 2
Кластер 3
J1.4
J1.5
J1.6
J1.7
J1.8
J1.9
J1.10
J1.11
J1.12
J1.13
Кластер 4
J1.14
J1.15
Кластер 5
Задача 2
J2.1
Кластер 1
J2.2
J2.3
Кластер 2
Кластер 3
J24
J2.5
J2.6
J2.7
J2.8
J2.9
J2.10
Кластер 4
J2.11
Кластер 5
Задача 3
J3.1
Кластер 1
J3.2
J3.3
J3.4
Кластер 2
J3.5
J3.6
J3.7
Кластер 3
J3.8
J3.9
Кластер 4

Рис.5.3

Кроме этого, для избежания случая, когда задачи накапливаются в системе из-за того, что все задания 1-ого кластера всех новых задач имеют прямой доступ к буферу, возможно присваивание *приоритетов* ρ_{ij} заданиям, так, чтобы задания задач находящихся "долго" в системе, имели предпочтение и выполнялись раньше других. Таким образом, в этой схеме имеется возможность для перехода на системы реального времени. Тогда приоритеты ρ_{ij} заданий становятся важнейшим фактором в принятии решения о распределении заданий—ресурсов в алгоритме планирования. Эти приоритеты вычисляются на основе временных характеристик задач (например, на сколько времени задача может задержаться в системе).

5. ***Распределение:*** из **M** заданий буфера нужно выбирать **N'** заданий для назначения на **N** имеющихся на данный момент ресурсов (узлов) так, чтобы:
   - эти **N'** назначений были **бесконфликтными;**
   - **N'=N** или было **максимальным**.

6. ***Оптимизация:*** из всех найденных вариантов расписания (состоящего из **N'** назначений) нужно выбирать один оптимальный вариант, который удовлетворял бы следующим требованиям:
   - сумма чистого времени выполнения **Te** и времени коммуникации **Tc** этих **N'** назначений должна быть минимальной (по сравнению с другими вариантами от **N'** назначений, то есть должна быть достигнута *локальная оптимизация*);
   - выбранный вариант расписания должен учитывать и другие факторы, влияющие на эффективность вычислительной системы в целом, как μ и ρ**,** в зависимости от степени реконфигурируемости системы (то есть, желательно обеспечить так же *глобальную оптимизацию*).

7. ***Перераспределение***: в случае отказа узлов—ресурсов или каналов связи нужно корректировать расписание соответственно новому состоянию системы. Однако, при организации вычислительного процесса по этой схеме, это не вызывает трудности, т.к. на новом кванте планирования эти изменения фиксируются как состояние ресурсов.

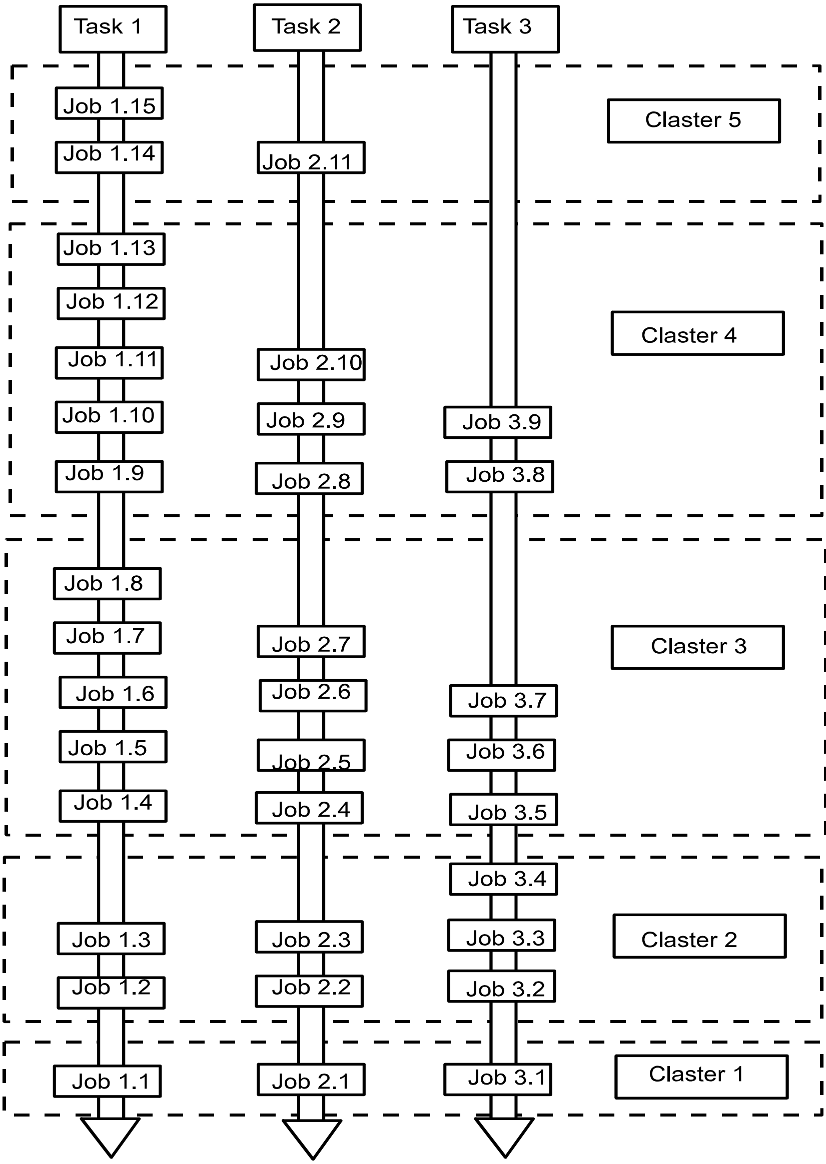


Рис 5.4. Схема распределения задач по кластерам

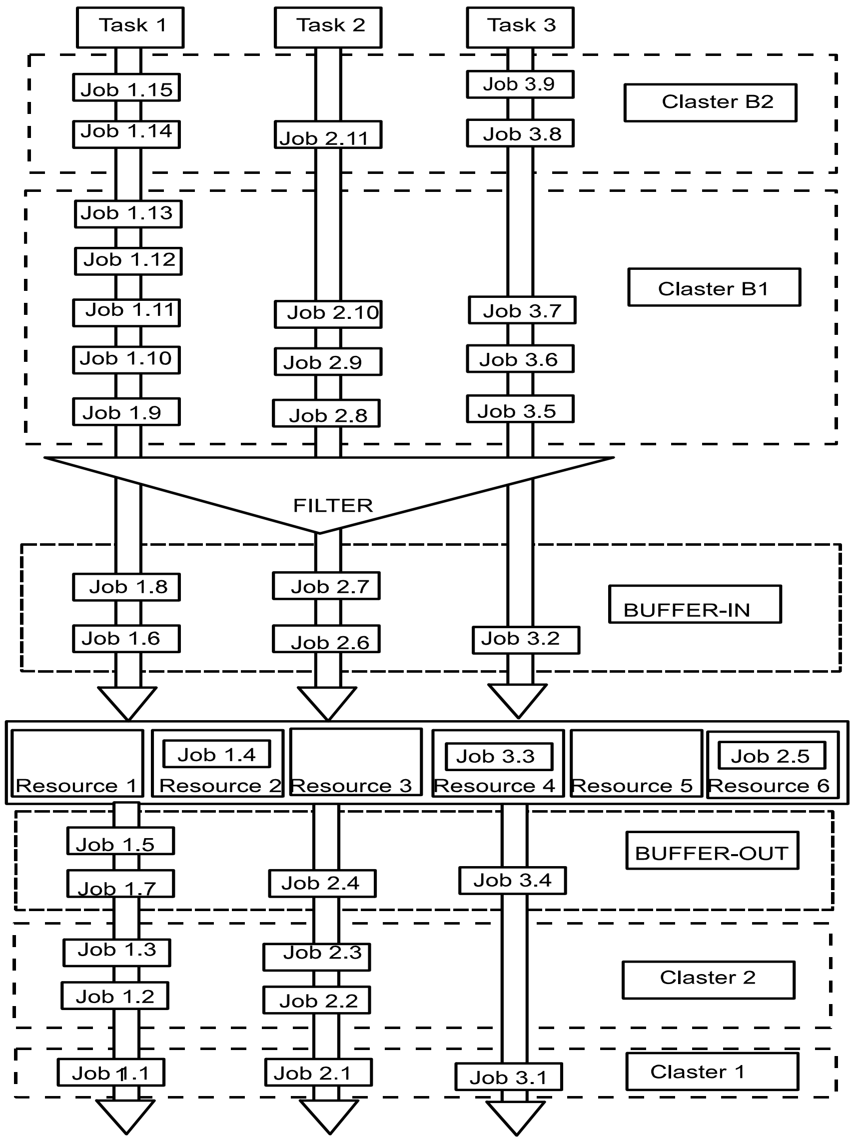

Task 1
Task 2
Task 3
Job 1.15
Job 1.14
Job 2.11
Job 3.9
Job 3.8
Claster B2
Job 1.13
Job 1.12
Job 1.11
Job 1.10
Job 1.9
Job 2.10
Job 2.9
Job 2.8
Job 3.7
Job 3.6
Job 3.5
Claster B1
FILTER
Job 1.8
Job 1.6
Job 2.7
Job 2.6
Job 3.2
BUFFER-IN
Resource 1
Job 1.4
Resource 2
Resource 3
Job 3.3
Resource 4
Resource 5
Job 2.5
Resource 6
Job 1.5
Job 1.7
Job 2.4
Job 3.4
BUFFER-OUT
Job 1.3
Job 1.2
Job 2.3
Job 2.2
Claster 2
Job 1.1
Job 2.1
Job 3.1
Claster 1

Рис. 5.5

5.3. МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ЗАДАЧ В НРС

Исследования управления, планирования и диспетчеризации в неоднородных распределенных системах обработки данных (НРС) активизировались в последнее время. Это связано с все более фундаментальными исследованиями в области распараллеливания и внедрением распределенных систем обработки информации. Самыми "горячими" точками процесса планирования в таких СНРС как PVM, P4 и SmartNet является решение задач *распределения* и *оптимизации* [80, 85]. Известно [3,4], что каждая, отдельно взятая, из них уже является NP-трудной.

Распределение в однородных системах не играет особую роль, так как все назначения являются возможными. Поэтому не имеют место конфликтные назначения, в результате выполнения которых некоторые ресурсы могут "зря" простоять, и ресурсы не сбалансированы по нагрузке. В НРС, где каждое задание $\mathbf{J_{i,j}}$ возможно выполнять только на некоторых, а не на всех ресурсах, задача распределения *играет особую роль,* так как она обеспечивает максимальное использование ресурсов и удовлетворение максимально возможного количества заявок.

Решение задачи *оптимизации* планирования в НРС должно обеспечить получение варианта расписания с максимальным выполнением требований, предъявляемых к расписанию и имеет два направления решения:

- Использование самых простых методов решения о назначении заявок на ресурсы [List Scheduling], которые имеют относительно малое время решения и временную сложность, приближающуюся к линейной (O(MxN)), но качество расписания далеко от оптимального варианта [141, 154].
- Использование сложных техник комбинаторной оптимизации [Genetic Algorithms, Simulated Annealing], которые имеют значительно большее время решения и временную сложность до ($O(n^5)$), не гарантирует получение оптимального варианта [92, 101].
- Использование методов точного решения, которые хотя, и дают точные оптимальные расписания, но практически, непригодны, т.к. имеют экспоненциальную временную сложность [9, 59].

Таким образом, решение задач распределения и оптимизации в НРС одновременно является критически трудным. Очень часто критерии оптимизации их решения являются противоположными и можно получить решение только близкое к оптимальному. В этой работе предлагается схема планирования на основе комбинированного алгоритма, то есть алгоритма, который способен выдать оптимизированное решение при приемлемой временной сложности. Более подробно теоретическое обоснование этого алгоритма описывается в [32, 35, 38, 39].

Для решения *комбинаторных задач* распределения и оптимизации, необходимо определить модель и дисциплину обслуживания заявок в этой модели.

5.3.1. Модель и дисциплина обслуживания процессов

оптимизации и распределения

Решением задач оптимизации и распределения в общей схеме планирования занимается планировщик среднего уровня, который обеспечивает прохождение заданий из буфера к обслуживающим их ресурсам. Рассмотрим действия планировщика более детально.

Объектами планировщика данной задачи являются:

- **BUFFER-IN** (где находятся M_τ заданий на данный момент τ).
- **N ресурсов** системы на момент τ.

Единицей для измерения времени является квант (К) равный времени выполнения процедуры планирования.

ОПИСАНИЕ ДАННЫХ:

Определенные выше объекты характеризуются следующими параметрами (значения которые хранятся в ведущем узле—Host).

- **Задания в BUFFER-IN**: характерезуются содержимым регистров BIN.L[1..M_τ] и BIN.E[1..M_τ] (рис. 5.6), где:
  - BIN.L[i] — список ресурсов {Rp,..,Rq}, с которыми данное задание i требует коммуникацию, т.е. характерезует связь обьектов по данным {ϕ_p,..,ϕ_q}.
  - BIN.E[i] — объем работы ε_i (измеряемое в К единицах), который нужно выполнить данному заданию i .

Дополнительными характеристиками для заданий находящихся в **BUFFER-IN** могут быть:

- BIN.M[i] — маркировка задания i. Если задание i не находится на критическом пути задачи, то BIN.M[i]=0. В противном случае BIN.M[i]= номеру ресурса, где было обработано задание i-1, предшествующее заданию i на критическом пути.
- BIN.P[i] — приоритет задания i, который был определен на предыдущем уровне планирования (например по времени нахождения задания в системе).

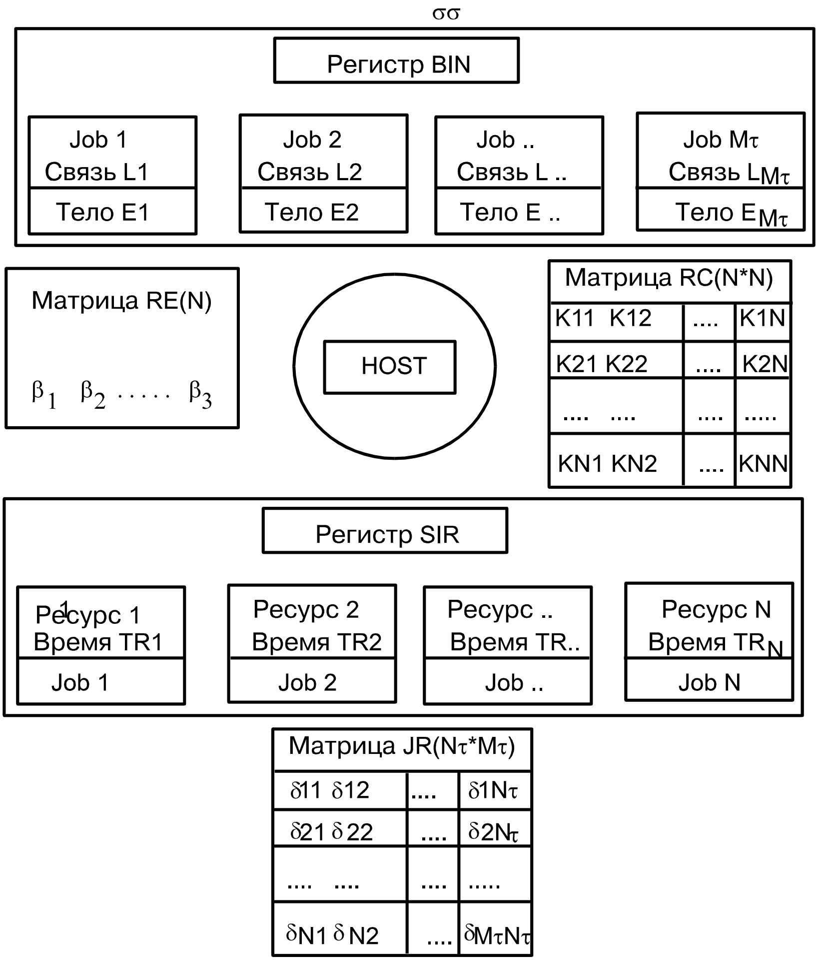


Рис. 5.6. Структура данных ресурс—задание

- **<u>Ресурсы Системы</u>**: состояния всех N ресурсов системы характеризуются содержимым регистров SIR.TR[1..N] и SIR.J[1..N] (рис. 5.6), где:
    - SIR.TR[i] — время (измеряется в К), которое данный ресурс i будет занят. Таким образом, если ресурс свободен, то его SIR.TR[i]=0.
    - SIR.J[i] содержит номер задания, которое выполняется на данном ресурсе (если он занят).

Ресурсы системы, кроме динамических характеристик, значения которых определяются в начале каждого кванта планирования, имеют еще статические характеристики, значения которых не изменяются в течение всего времени работы системы до следующей перезагрузки или инсталяции системы:

- RC[1..N, 1..N]: матрица затрат на передачи информации между ресурсами, где $RC[i,j]=\beta_{ij}$ есть временные затраты передачи определенного объема информации из ресурса i в ресурс j за время К.
- RE[1..N]: матрица выполнения, которая характеризует коэффициенты относительной взвешенной производительности для ресурса i $RE[i]=K_i$. Этот коэффициент вместе с объемом работы ε_j (измеряемый в К единицах) задания j определяет действительное время выполнения задания j на ресурсе i.

(В случае реконфигурируемых систем, выше перечисленные статические данные могут стать динамическими)

ОПИСАНИЕ ДЕЙСТВИЙ:

Процессы оптимизации и распределения необходимо рассматривать вместе с процессами загрузки и разгрузки ресурсов системы, так как эти процессы тесно связаны друг с другом. Действия планировщика можно представить в виде следующей последовательности действий:

Проверка и подготовка (ПП): монитор ОС в ведущей HOST машине проверяет и собирает информацию о состоянии системы, т.е. составляет "образ" системы состоящей из следующей информации: состоянии ресурсов (рабочих узлов), фиксация поступающих новых заданий, готовность к выполнению уже находящихся в системе заданий, следующим образом:

- В случае выхода из строя какого-то ресурса, фиксирует в **BUFFER-IN** задания, выполнявшиеся на этом ресурсе, как невыполненные и удаляет отказавший ресурс из системы планирования.
- Определяет число свободных ресурсов N_τ на момент времени τ.
- Определяет готовые задания и их описание — формирует содержимое регистров BIN.L, BIN.E, BIN.M, и BIN.P в **BUFFER-IN**. Определяет число заданий M_τ **,**подлежащих распределению из общего числа заданий, находящихся в **BUFFER-IN** на момент τ**,**.

На основе этой информации, монитор ОС (находится в HOST) вычисляет и подготавливает необходимые данные для планировщика (тоже находится в HOST).

Начало оптимизации и распределения (НОР): после подготовки данных для момента времени начала планирования τ**,** в ведущем узле HOST имеются следующие данные о:

- M_τ заданиях, находящихся в **BUFFER-IN**: регистры BIN.L, BIN.E, BIN.M, и BIN.P;

- состоянии **N ресурсов** системы: регистры SIR.TR и SIR.J, матрица RC и вектор RE.

НОР выполняется в несколько этапов:

Этап 1: формирование матрицы ресурс—задание JR.

Из данных о заданиях в **BUFFER-IN** и из статических и динамических характеристик ресурсов определяются элементы матрицы взаимоотношения между ресурсами и заданиями $JR[1..N_\tau,1..M_\tau]$, где, $JR[i,j]=[Te_{i,j}+Tc_{i,j}]/\rho_i$, ρ_i - приоритет задания J_i (1)

Для определения $Te_{i,j}$ используем BIN.E[j] и RE[i], где SIR.TR[i]=0.

Для определения $Tc_{i,j}$ используем BIN.L[j] и RC[i,l], где $l \in$ BIN.L[j]

Этап 2: формирование расписания из матрицы ресурс—задание $JR[1..N_\tau, 1..M_\tau]$, где N_τ есть число свободных ресурсов на данный момент (рис. 5.7).

Расписание SCH представляет собой матрицу—запись включающую:

- SCH.NJ[1..N*]=J*, номер задания выбранное планировщиком из **BUFFER-IN**.
- SCH.NRE[1..N*]=R*, номер ресурса из числа свободных, на котором указанное задание будет выполняться.

Расписание SCH для N_τ из M_τ заданий на N_τ имеющихся ресурсах должно удовлетворять следующим условиям оптимизации и распределения:

- набор назначений в расписании должен быть бесконфликтным.
- число назначения максимально.
- сумма затрат выполненных назначений минимальна.

Конец ОР: передача расписания SCH[1..N*] загрузчику ОС ВС.

Далее загрузчик выполняет следующие действия:

Разгрузка: для занятых на момент времени τ ресурсов i вычисляются новые значения $SIR.TR[i]^{\tau+1}=SIR.TR[i]^{\tau}-1$ для нового $\tau+1$ момента планирования. Если значение $SIR.TR[i]^{\tau+1}=0$, то задание, номер которого хранится в SIR.J[i]=j, считается выполненным, задание удаляется и ресурс становится доступным для системы планирования. При этом, номер данного ресурса записывается в регистр BIN.L[j*] для задания j* (из кластера B1), являющимся первым в очереди готовых заданий.

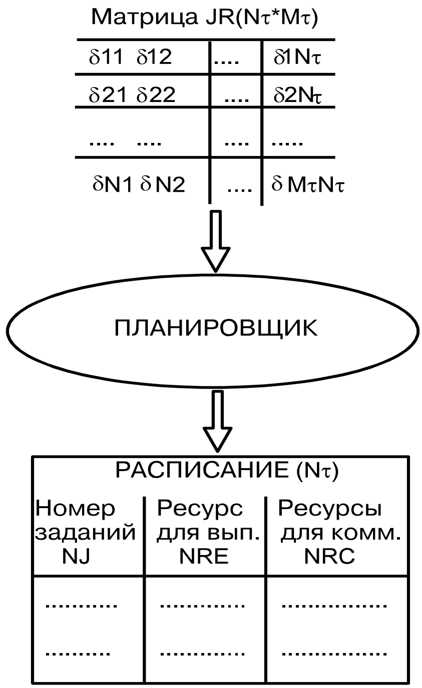


Рис. 5.7. Ввод и вывод оптимизации и распределения

Загрузка: по содержанию расписания SCH загрузчик загружает N* заданий на N* соответствующих ресурсов и записывает новые значения в регистры SIR.TR[1..N] и SIR.J[1..N]:

- Для N* свободных ресурсов в расписании: SIR.TR[R*]=$Te_{R*,J*}$ и SIR.J[R*]=J*.
- Для N-N* остальных ресурсов в регистрах ничего не изменяется.

На этом система планирования заканчивает свои действия, и система управления выполняет операции, связанные со слежением за активизированными процессами.

Выполнение: задания выполняются.

В момент времени **τ+1** выполняются следующие действия:

ПП: Монитор ОС проверяет состояние ресурсов системы:

- в случае выхода из строя какого-то ресурса i#, который выполняет задание j#, это задание записывается обратно в **BUFFER-IN,** но с значением $\varepsilon_{j\#}=(Te_{i\#,j\#}/k_{i\#})$, а поля SIR.TR[i#] и SIR.J[i#] обнуляются и эти ресурсы не учитываются на новом цикле планирования.

- определяется число свободных ресурсов $\mathbf{N}_{\tau+1}$.
- с помощью FILTER определяются новые готовые задания из кластера B1 и их данные: регистры BIN.L, BIN.E, а также возможно, BIN.M, и BIN.P и эта информация посылается в **BUFFER-IN**. Определяется число заданий $\mathbf{M}_{\tau+1}$, находящихся в **BUFFER-IN** на момент $\boldsymbol{\tau+1}$.

Таким образом, завершается цикл распределения заданий и загрузки ресурсов системы.

5.4. ОБЩАЯ СХЕМА РАБОТЫ СИСТЕМЫ

Выше описанные действия планировщика могут выполнятся по двум вариантам, в зависимости от соотношения между динамичностью системы и заданным временем планирования:

1.Режим ON-LINE: этот режим может быть применен, когда время планирования (для разработки одного расписания) меньше, чем период времени (Квант) за которое HOST узел получает информацию о системе. В этом случае описанные действия выполняются в следующем порядке:

1. Проверка и подготовка.
2. Оптимизация и распределение.
3. Загрузка.
4. Выполнение.

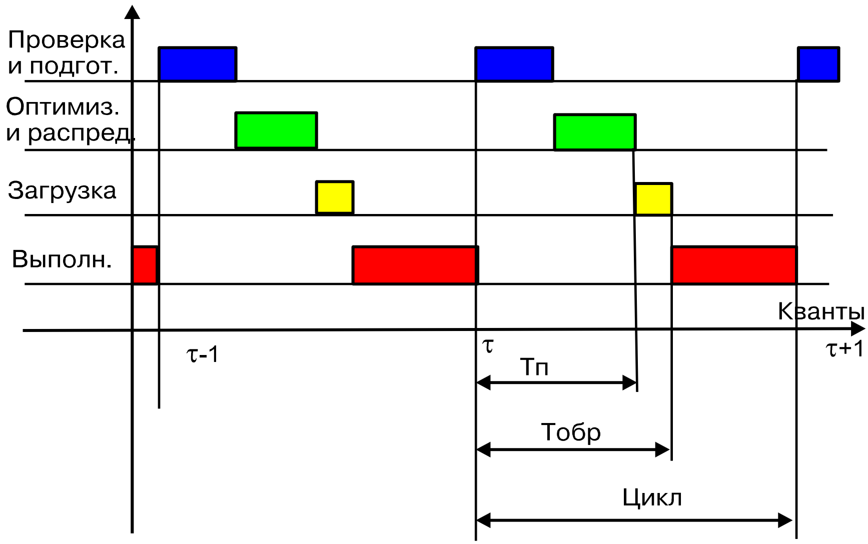
Таким образом, данные, полученные в момент времени τ (в начале сеанса) процедурой ПП, используются для планирования и изменения состояния системы во время решения задачи планирования не учитываются.

2.Режим OFF-LINE: этот режим используется, когда время планирования (для разработки одного расписания) больше, чем период времени, выделенный на планирование (Квант) за которое HOST общается с рабочими узлами для получения информации о их состоянии. В этом случае, описанные действия могут выполняться параллельно по следующей схеме:

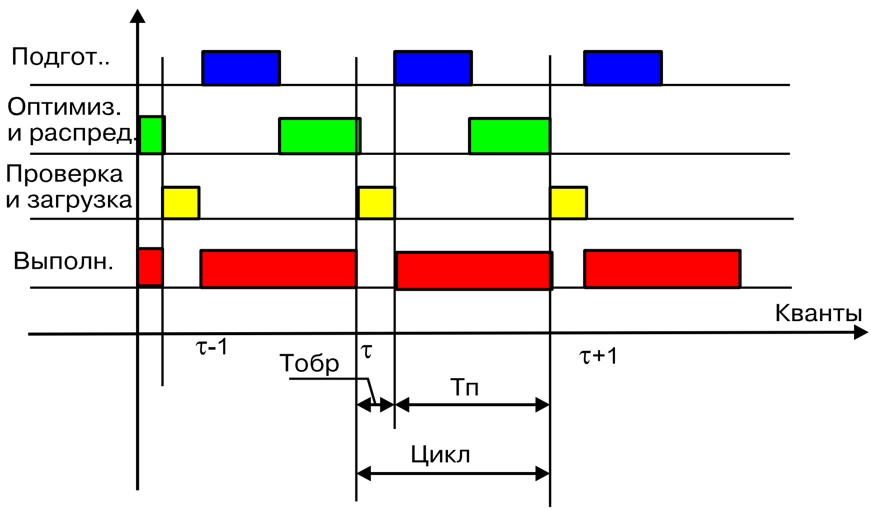
1. Загрузка. 1. Проверка и подготовка.
2. Выполнение. 2. Оптимизация и распределение.

Таким образом, данные, полученные в момент времени τ процедурой ПП во время активизации планировщика, используются для планирования в следующий момент времени $\tau+1$. Планирование (оптимизация и распределение) и выполнение производятся параллельно.

Согласование процессов по времени для этих двух вариантов показаны во временных диаграммах (рис. 5.8).



а). Для ON_LINE режима



б). Для OFF_LINE режима

Рис. 5.8. Временные диаграммы процессов планирования

5.5. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ И РАСПРЕДЕЛЕНИЯ

В неоднородной распределенной системе обработки данных (НРСОД), состоящей из N ресурсов, на данный момент времени τ имеются N_τ свободных ресурсов и M независимых, готовых к выполнению заданий.

- *Система ресурсов* задана графом системы $G_R=(V_R,E_R,W_{VR},W_{ER})$, где:
  - Множество вершин $V_R=\{R_1, R_2,..., R_N\}$, каждый элемент которого представляет один из N ресурсов системы и $R_i \in N|$ (множество натуральных чисел), i=1..N.
  - Множество дуг $E_R=\{E_1, E_2,.., E_d\}$, каждый элемент которого представляет связь между двумя ресурсами $E_i=\{R_i,R_j\}$, где $R_i,R_j \in V_R$ и $0 \le d \le N^2$.
  - Множество весов вершин $W_{VR}=\{WVR_1, WVR_2,..., WVR_N\}$, где $WVR_i=\{RE_i, RT_i\}$. Для $\forall i=1..N$, $RE_i \in \Re^+$ (множество положительных реальных чисел) есть характеристика ресурса R_i, $RT_i \in \{0 \text{ и } \Re^+\}$ — состояние ресурсов.
  - Множество весов дуг $W_{ER}=\{WER_1, WER_2,.., WER_p\}$. Это множество можно представить в виде некоторой матрицы $RC=RC[i,j] \in \Re^+$, где i=1..N и j=1..N.
- *Поток M заданий*, задан множеством $V_J=\{Job_1, Job_2,..., Job_M\}$, каждый элемент которого представляет одно из M заданий и $Job_i=\{JN_i, JE_i, JL_i, JM_i, JP_i\}$, где для $\forall i=1..N$:
  - $JN_i \in N|$ есть номер задания,
  - $JE_i \in \Re^+$ есть объем работы задания,
  - $JL_i=\{(R^1, \varphi_1),.., (R^q, \varphi_q)\}$, где $R^l \in V_R$ есть ресурс, с которым данное задание требует обмена данными, $\varphi_l \in \Re^+$ — объем передачи, l=1..q, $q \in N|$,
  - $JM_i=\{0 \text{ или } R^i\}$ есть маска задания, где $R^i \in V_R$ — номер ресурса, на котором желательно выполнять данное задание,
  - $JP_i \in \Re^+$ есть приоритет данного задания.

<u>*Определение 1*</u>: Г есть отображение множества заданий $V_J=\{Job_1, Job_2,.., Job_M\}$ на множество ресурсов $V_R=\{R_1, R_2,.., R_N\}$ графа системы $G_R=(V_R,E_R,W_{VR},W_{ER})$, если результат отображения $\Gamma(V_J,V_R)$ есть некоторое множество A: $A=\{a_1, a_2,.., a_n\}$, где $a_i=(R^i, J^i)$, $R^i \in V_R$, $J^i \in V_J$, i=1..n, $n \in N|$.
Обозначим $AR=\{R^1, R^2,.., R^n\}$, $AJ=\{J^1, J^2,.., J^n\}$. Таким образом, $|A|=|AR|\cap|AJ|$, $AR \subseteq V_R$, $AJ \subseteq V_J$.

<u>*Определение 2:*</u> отображение Г есть ***распределение*** заданий V_J на ресурсы V_R, если его результат $\Gamma(V_J,V_R)=A$, где $A=\{(R^1, J^1), (R^2, J^2),.., (R^n,J^n)\}$ удовлетворяет следующему условию: для $\forall i=1..n$, $R^i \notin AR \backslash R^i$, $J^i \in AJ \backslash J^i$, где $AR=\{R^1, R^2,.., R^n\}$, $AJ=\{J^1, J^2,.., J^n\}$. ***Размером*** данного распределения $\Gamma(V_J,V_R)$ является число n. Тогда $\Gamma(V_J,V_R) \to A$, $n=|A|$.

<u>*Определение 3:*</u> результат распределения заданий на ресурсах $A=\Gamma(V_J,V_R)$ называем ***расписанием*** для данного распределения Г. Пара $a_i=(R^i, J^i)$, i=1..n называется ***назначением*** для ресурса $R^i \in V_R$, и для задания $J^i \in V_J$.

Определение 4: пусть $X=\{A^1, A^2,.., A^z\}$, $z \in N|$ есть множество результатов всех возможных распределений для множества заданий V_J и для множества ресурсов V_R. Тогда $\Gamma(V_J,V_R) \equiv X$. Распределение заданий на ресурсах $\Gamma(V_J,V_R) \to A^*$ есть ***максимальное распределение*** для данных множества заданий V_J и множества ресурсов V_R если:

1) $n^*=|A^*|$;
2) $n^*=\max\{|A^1|, |A^2|,.., |A^z|\}$.

Определение 5: пусть Δ есть некоторая функция от назначения $a_s=(R^s, J^s)$ (то есть назначения задания J^s на ресурс R^s, $R^s \in V_R$ и $J^s \in V_J$). Тогда $\Delta(a_s)=\Phi$ или $\Phi=\Delta(R^s, J^s)$ и $\Phi_i=\Delta(a_i)=\Delta(R^i, J^i)$, где $i=1..n$ назовем ***весом назначения*** $a_i=(R^i, J^i)$ по Δ.

Определение 6: сумму весов всех назначений $\{a_1, a_2,.., a_n\}$ назовем ***весом D(A) расписания*** A. То есть: $D(A) = \sum_{i=1}^{n} \Delta(a_i)$.

Определение 7: пусть $Xm=\{A_1, A_2,.., A_m\}$, $m \in N|$ есть множество всех максимальных распределений для множества заданий V_J и для множества ресурсов V_R. Тогда расписание $A^*=\Gamma(V_J,V_R)$ — ***оптимальное расписание*** распределения (заданий V_J на ресурсах V_R) Γ по измерению Δ, если A^* удовлетворяет следующим условиям:

1) $A^*=\{(R^1, J^1), (R^2, J^2),.., (R^n, J^n)\}$ является результатом максимального распределения для данных множества заданий V_J и множества ресурсов V_R, то есть $|A^*|=\max\{|A^1|, |A^2|,.., |A^z|\}$; (определение 5).
2) Вес расписания $A^*=\{a_1, a_2,.., a_n\}$ был максимальным из $Xm=\{A_1, A_2,.., A_m\}$, то есть:

$$D(A^*) = \sum_{i=1}^{n} \Delta(a_i^*) = \max\{D(A_1), D(A_2), \dots, D(A_m)\} = \max_{j=1}^{m}\{D(A_j)\}$$

Требование: нужно найти ***оптимальное*** (вес по заданной функции Δ) ***расписание A***$=\{(R^1, J^1), (R^2, J^2),.., (R^n, J^n)\}$, $n \in N|$ максимального распределения Γ (по определению 7) для N_τ свободных ресурсов (V_R) и M готовых к выполнению заданий (V_J).

5.5.1. Общая схема решения

Строим математическую *модель* решения для задачи оптимизации и распределения (математическая постановка которой была представлена в [39, 125, 126]) на основе модели оптимизации и распределения, которая была разработана в [38, 126].

Решение данной задачи для N_τ ресурсов $V_R=\{R_1, R_2,.., R_N\}$ и M заданий $V_J=\{J_1, J_2,.., J_M\}$ состоит из следующих этапов:

1 Определить функцию Δ для измерения весов назначения. Определить веса $\delta_{i,j}$ ($i=1..N_\tau$, $j=1..M$) всех возможных назначений по функции Δ.
2 Найти оптимальное расписание $A=\{a_1, a_2,.., a_n\}$, где $a_i=(R^i, J^i)$, $R^i\in V_R$, $J^i\in V_J$, $i=1..n$, $n\in N|$, которое удовлетворяет условиям определения 7 и весовыми значениями, определенными на первом этапе.

5.5.2. Определение функции измерения качества решения

При оптимизации и распределении, функцию Δ для измерения веса назначения задания J_j на ресурс R_i ($R_i\in V_R$ и $J_j\in V_J$), в общем случае, можно определить следующим образом:

$$\Delta(R_i, J_j)= \delta_{i,j} = \prod_{k=1}^{K} P_k^{i,j} \times \prod_{x=1}^{H} C_x^{i,j} \times \sum_{y=1}^{G} L_y O_y^{i,j} \qquad (2)$$

Где,

- $\prod_{k=1}^{K} P_k^{i,j}$ есть величина абсолютного приоритета назначения (R_i, J_j). Он вычисляется путем умножения величин всех K относительных приоритетов $P_k^{i,j} \in \Re^+$ не только заданий, но и ресурсов (которые связываются с разными факторами: время ожидания заданий, работоспособность ресурсов и т.д.). Для $\forall k=1..K$ $p^d \le P_k^{i,j} \le p^u$, поэтому $P^d \le \prod_{k=1}^{K} P_k^{i,j} \le P^u$.
- $\prod_{x=1}^{H} C_x^{i,j}$ есть результат анализа всех H обязательных требований, $C_x^{i,j}$ есть степень выполнения обязательного требования x для назначения задания J_j на ресурс R_i, $C_x^{i,j} \in \{0,1\}$. Например, требования наличия каналов передачи, объема требуемой памяти, наличия программ, данных и т.д. $C_x^{i,j}=1$, если ресурс R_i удовлетворяет требованиям задания J_j, $C_x^{i,j}=0$ в противном случае.
- $\sum_{y=1}^{G} L_y O_y^{i,j}$ есть результат анализа всех G оптимизирующих требований, где $O_y^{i,j} \in \Re^+$ и $O^d \le O_y^{i,j} \le O^u$ есть степень выполнения оптимизирующего требования **y** назначения задания J_j на ресурс R_i; $L_y\in\Re^+$ и $L^d \le L_y \le L^u$ есть весовой коэффициент оптимизирующего требования y.

В нашей системе представления исходной информации имеем :

- $\prod_{k=1}^{K} P_k^{i,j}$ вычисляется с помощью приоритета JP_j данного задания J_j, где $JP_j=1/Tw_j=\rho_j$ (Tw_j есть время ожидания задания J_j в системе) и маски задания

$JM_j=\{0$ или $R^*\}$, $R^*\in V_R$ следующим образом:

$$\prod_{k=1}^{K} P_k^{i,j} = \mu_i \times \rho_j ,$$

где, $\mu_i = \begin{cases} M^o, åñëè R_i = JM_j = R^* \\ 1, åñëè R_i = JM_j \in \{0, R^*\} \end{cases}$

- $\prod_{x=1}^{H} C_x^{i,j}$ вычисляется с помощью сравнения характеристик по коммуникации задания $JL_j=\{(R^1, \varphi_1),..,(R^q,\varphi_q)\}$ с множеством дуг графа системы ресурсов $E_R=\{E_1, E_2,.., E_d\}$:

  для $\forall l=1..q$: $CC_l^{i,j} = 1$, если $(R_i, R^l)\in E_R$;

  $CC_l^{i,j} = 0$, если $(R_i,R^l)\notin E_R$;

  В конечном итоге $\prod_{x=1}^{H} C_x^{i,j} = C^{i,j} = \prod_{l=1}^{q} CC_l^{i,j}$.

$\sum_{y=1}^{G} L_y O_y^{i,j}$ вычисляется как сумма обратной величины времени выполнения $Te_{i,j}$ и обратной величины времени коммуникации $Tc_{i,j}$. Коэффициент выполнения ресурса $RE_i = k_i$ из его веса в WVR_i, объем работы задания $JE_j=\varepsilon_j$ — из матрицы весов дуг в графе системы ресурсов $RC[k,l]=\beta_{k,l}$, где $k=1..N$ и $l=1..N$, объемы требования задания — по коммутации $JL_i=\{(R^1, \varphi_1),..,(R^q, \varphi_q)$.

Таким образом $Te_{i,j}$ и $Tc_{i,j}$. вычисляются следующим образом:

$$Te_{i,j} = \varepsilon_j * k_i \; ; \quad Tc_{i,j} = \sum_{l=1}^{q} (\varphi_l * \beta_{i,l}) .$$

Таким образом, мы имеем:

$$\sum_{y=1}^{G} L_y O_y^{i,j} = 1/Te_{i,j} + 1/Tc_{i,j}) = 1/\ \varepsilon_j * k_i \ + 1 /\ \sum_{l=1}^{q} (\varphi_l * \beta_{i,l})$$

Из формулы (2) мы имеем:

$$\Delta(R_i, J_j) = \delta_{i,j} = (\mu_i \times \rho_j) \times C^{i,j} \times [1 \div (\varepsilon_j \times k_i) + 1 \div \sum_{l=1}^{q} (\varphi_l * \beta_{i,l})] \quad (3)$$

Очевидно, что $\delta_{i,j} \geq 0$ для $\forall i=1..N_\tau$, $j=1..M_\tau$. Поэтому:

$\inf(\Delta(R_i, J_j)) = 0$.

В случае отсутствия связи между ресурсами i и l, RC[i,l] получает такое значение $\beta_{i,l}$, что $Tc_{i,,j}= \sum_{l=p}^{q} (\phi_l * \beta_{i,l}) > \lambda_0$, где λ_0 некоторое заданное число. Число λ_0 есть порог для определения существования связи между двумя ресурсами. Время выполнения $Te_{i,j}$ имеет некоторую нижнюю границу $\mathbf{T^o}$.

Верхняя граница диапазона изменения $\boldsymbol{\delta_{i,j}}$ определяется следующим образом:

$$\sup(\Delta(R_i, J_j)) = (\mu_{Oj} \times \rho_{Oj}) \times [1/T^O + 1/\lambda_O] = \delta_{max}.$$

Определенные значения $\boldsymbol{\delta_{i,j}}$ для $i=1..N_\tau$, $j=1..M_\tau$ хранятся в матрице $JR[1..N_\tau, 1..M_\tau]$.

5.5.3. Определение оптимального распределения

Множества N_τ ресурсов $V_R=\{R_1, R_2,.., R_{N\tau}\}$ и M заданий $V_J=\{J_1, J_2,..,J_M\}$ можно представлять как множество вершин некоторого графа G . Тогда множество неориентированных дуг $E=\{E_1, E_2,.., E_d\}$ между вершинами графа G соответствует множеству назначений задания J* на ресурсе R* (пример графа для 6-и ресурсов и 6-и заданий на рис. 5.8). Дуга $E_k=\{R_i, J_j\}$, где $R_i \in V_R$ и $J_j \in V_J$, где $k=1..d$, $0 \le d \le N_\tau \times M$, между вершинами J_j и R_i существует только тогда, когда назначение задания J_j на ресурсе R_i является *"невозможным"*, то есть когда $\boldsymbol{\delta_{i,j} \le \delta_0}$, где $\boldsymbol{\delta_0}$ есть некоторое заданное число (в данном примере $\boldsymbol{\delta_0}=1$).

Тогда, выполнение второго этапа представляет собой *задачу назначения*. Существует несколько методов для решения задачи назначения для взвешенного двудольного графа $G=(V_R,V_J, E, WE)$:

где, $V_R=\{R_1, R_2,.., R_{N\tau}\}$ и $V_J=\{J_1, J_2,.., J_M\}$,

$E=\{E_1, E_2,.., E_d\}$, $E_k=\{R^*,J^*\}$, где $R^* \in V_R$ и $J^* \in V_J$,

где $k=1..d$, $0 \le d \le N_\tau \times M$.

$WE=\{WE_1, WE_2,.., WE_d\}$, $WE_k=\Delta(E_k)$, где $k=1..d$, $0 \le d \le N_\tau \times M$.

Решение задачи назначения для графа размером $N_\tau \times M$, где $N_\tau \ne M$ приводится к решению задачи назначения для графа размером $N \times N$, где $N=\max\{N_\tau; M\}$ [6, 14, 15, 19, 21, 23, 24, 29, 33].

Решение задачи назначения для взвешенного графа G можно привести к решению задачи назначения для невзвешенного графа G' полученного из графа G снятием весов всех дуг.

Задача назначения для данного случая приводится к решению задачи поиска максимального паросочетания для взвешенного или невзвешенного двудольного графа.

ЗАДАНИЯ

РЕСУРСЫ

№	1	2	3	4	5	6
1	3.5	0.2	7.1	0.6	6.1	0.1
2	0	4.1	2.1	5.3	8.7	3.3
3	2.1	5.9	0.3	0	7.2	0.7
4	4.5	3.6	5.1	0.8	3.1	0
5	8.1	7.1	0	0	4.9	0.4
6	0.9	0	2.1	6.8	0	8.1

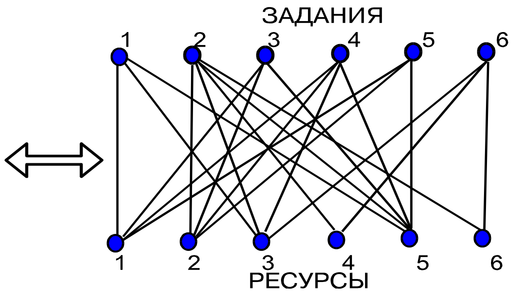


Рис. 5.8

Задача назначения в такой постановке решается во многих прикладных приложениях. На выбор метода и алгоритма решения влияет временная сложность, т.к. время решения задач планирования, особенно при динамическом планировании, является основным критерием, даже в ущерб качества получаемого решения. Как уже было сказано, т.к. большинство задач оперативного диспетчирования или динамического планирования можно свести к задаче поиска максимального паросочетания, то целесообразно выполнение сравнительного анализа известных алгоритмов на основе единого критерия. Наиболее приемлемым критерием является временная сложность алгоритма, позволяющая оценить изменение времени решения задачи от размерности, что существенно для параллельных систем. Наиболее часто используемые подходы к решению данной задачи.

- **поиск максимального потока в сети**
- **поиск максимального паросочетания.**

Задача о максимальном потоке является одной из самых фундаментальных задач в теории потоков в сетях. Впервые эта задача была сформулирована Фалкерсоном и Данцингом в 1955 году, и Данцингом и Фалкерсоном отдельно в 1956, а алгоритм ее точного решения описан Фордом и Фалкерсоном в 1956 году, с использованием их известного алгоритма увеличивающего пути. С тех пор многие исследователи пытались уменьшить временную сложность предложенного Фордом и Фалкерсоном алгоритма и появилось множество алгоритмов для решения этой задачи. В табл. 5.1 приведены сравнительные данные наиболее известных алгоритмов поиск максимального потока в сети.

Задача о максимальном потоке формулируется следующим образом. Задан ориентированный граф $G=(V,E)$, каждой дуге $e=(u,v)\in E$, которого приписана пропускная способность $b(e)$, и две вершины - источник s и сток t. Говорят, что на сети задан поток величины v, если для каждой дуги $e\in E$ определены дуговые потоки $f(e)$ $(f\colon E\to R)$, удовлетворяющие соотношениям:

$0\le f(e)\le b(e)$, $\forall e=(x,y)$ или $e=(y,x)$

$$\sum_{v \in Î^{+}(u)} f(u,v) - \sum_{v \in O^{-}(u)} f(v,u) = \begin{cases} w, \text{åñëè } u = s, \\ -w, \text{ åñëè } u = t, \\ 0, \text{ åñëè } u \neq s,t. \end{cases}$$

Задача нахождения максимального потока в сетях состоит в определении дуговых потоков f(e), удовлетворяющих указанным условиям, для которых величина потока v достигает максимума.

Задача о максимальном паросочетании формулируется следующим образом. Для заданного графа G=(V,E) требуется найти множество ребер $M \subseteq E$ максимальной мощности, такое, что никакие два ребра из M не имеют общих концевых вершин. Такая формулировка задачи поиска максимального паросочетания справедлива для невзвешенного графа. В другом варианте даны также веса ребер и целью решения задачи является нахождение паросочетания, имеющего наибольший суммарный вес. Обе задачи вызвали большой интерес исследователей в последние три десятилетия. Они легко формулируются, апеллируют к интуиции и имеют много приложений. Кроме того, имеются интересные алгоритмы, эффективно решающие эти задачи. Ряд книг, содержащих наиболее полное изложение вопросов, связанных с задачей о паросочетании, написали Форд и Фалкерсон, Берж, Пападимитриу и Стайглиц, Кофман, Липский, Харари, Оре. Однако в случае задачи о паросочетании эффективное обнаружение и выполнение увеличений может оказаться непростым делом. Как в невзвешенном, так и во взвешенном вариантах задачи о паросочетании, все эти вопросы существенно упрощаются, если рассматриваемый граф является двудольным. В табл. 5.2. приведены сравнительные данные наиболее известных алгоритмов поиск максимального паросочетания.

Табл.5.1. Сравнительная оценка временной сложности различных алгоритмов решения задачи о максимальном потоке

№	Дата (год)	Авторы	Временная сложность
1	1956	Форд и Фалкерсон	$O(nmU)$
2	1969	Эдмотдс и Карп	$O(nm^{2)}$
3	1970	Диниц	$O(n^2m)$
4	1974	Карзанов	$O(n^3)$
5	1977	Черкасский	$O(n^2m^{1/2})$
6	1978	Малхотри, Кумара и Махешвари	$O(n^3)$
7	1978	Галил	$O(n^{5/3}m^{2/3})$
8	1978	Галил и Наамад	$O(nm\log^2 n)$
9	1978	Шилоач	$O(nm\log^2 n)$
10	1980	Слеатор, Тарьян	$O(nm\log n)$

11	1982	Шилоач, Вишкин	$O(n^3)$
12	1983	Габов	O(nmlogU)
13	1984	Тарьян	$O(n^3)$
14	1985	Гольдберг	$O(n^3)$
15	1986	Гольдберг, Тарьян	$O(nm\log(n^2/m))$
16	1986	Бертсекас	$O(n^3)$
17	1986	Ахуя, орлин	$O(nm+n^2\log U)$
18	1988	Чериян, Махешвари	$O(n^2m^{1/2})$
19	1988	Ахуя, Орлин, Тарьян	a). $O(nm + \frac{n^2 \log U}{\log\log U})$ b). $O(nm + n^2\sqrt{\log U}$ c). $O(nm\log(\frac{n\sqrt{\log u}}{m} + 2))$
20	1990	Чериян, Хагеруп,Мехлори	$O(n^3/\log n)$

Табл.5.2. Сравнительная оценка временной сложности решения задачи поиска максимального паросочетания

№	Дата (год)	Авторы	Временная сложность
1	1965	Эдмондс	$O(n^4)$
2	1973	Хопкрофт, Карп	$O(n^{1/2}m)$
3	1975	Эвен, Карив	$O(n^{1/2}m)$
4	1976	Габов	$O(n^3)$
5	1980	Мисали, Визирани	$O(n^{1/2}m)$
6	1980	Саваге	O(n)
7	1982	Пападимитриу	O(min(|V|,|V|,|E|)
8	1982	Галил, Микали	O(nmlog(n))
9	1984	Галло	O(nlogn)
10	1984	Филипс,ГарсиаДиас	$O(n^{1/2}m)$
11	1990	Блум	$O(\sqrt{n}m)$
12	1990	Шей, Куо, Чен	$O(n^{1/2}\log m)$
13	1992	Ксима, Ловасз	$O(n^2r\log r)$
14	1992	Чи, Шингтсан	$O(n^3)$
16	1992	Голдберг, Плоткин	$O(\sqrt{m}/\log^3 n)$
17	1993	Альт, Блум	$O(n^{1.5}\sqrt{m}/\log n)$