

## АНАЛИЗ СИСТЕМ И МЕТОДОВ БАЛАНСИРОВКИ НАГРУЗКИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В данной статье проведен анализ систем и методов балансировки нагрузки вычислительных систем с учетом современной методической и программно-аппаратной базы. Также показано нынешнее состояние данной проблемы и сформулированы новые и нерешенные задачи в области эффективного распределения заданий в вычислительной системе.

This paper analyzes systems and methods for load balancing of computer systems with modern methodology and software and hardware solutions. The current status of the problem is considered and arising and unresolved problems in the field of efficient tasks scheduling in the computing system are outlined.

### Введение

Для планирования использования ресурсов компьютерных систем с целью решения вычислительных задач в настоящее время используется комплекс различных методов. От адекватности выбранных методов зависит эффективность функционирования вычислительной системы. Для достижения нового вклада в ту или иную отрасль обязательным является изучение существующих наработок в данной области знаний. Поэтому в данной статье поставлена цель провести анализ подходов построения систем и методов балансировки нагрузки вычислительных систем с учетом современной методической и программно-аппаратной базы, что даст возможность не только показать нынешнее состояние данной проблемы, но и сформулировать новые и нерешенные задачи в области эффективного распределения заданий в вычислительной системе.

Для достижения поставленной цели был проведен предварительный поиск информации по РЖ ВИНТИ за последние 20 лет. Анализ публикаций показывает, что разработками и исследованиями проблемы балансировки нагрузки в вычислительных системах занимаются во всех ведущих странах. Поэтому было пересмотрено ряд патентных серверов США, Великобритании, Японии, Германии, а также России и Украины. Пересмотрено по сети Интернет более 1000 содержаний документов, около 120 отобрано для дальнейшего анализа. Документы, заслуживающие, на мой взгляд, наибольше-

го внимания, основываясь на полноте охвата решения проблемы и практической значимости, кратко охарактеризованы в данной статье. С полной версией отчета о проведенных патентных исследованиях можно ознакомиться на сайте [http://yurich.org.ua/patent\\_report.html](http://yurich.org.ua/patent_report.html).

### 1. Устройство и метод для балансировки нагрузки между различными путями в компьютерной системе

В [1] система и метод оценивают приспособленность различных путей в компьютерной системе, определяя приспособленность или состояние на не менее, чем двух уровнях каждого пути. Приспособленность или состояние путей определяется на основании информации о приспособленности ветки на каждом уровне пути. Устройство балансировки нагрузки направляет информацию для обработки различными путями, основываясь на определенной приспособленности этих путей. Пути с наиболее высокой приспособленностью отмечаются специальным алгоритмом путем «взвешивания», давая больший вес уровням большей важности. Система и метод могут также добавлять или удалять дополнительные пути, если общая определенная приспособленность существующих путей выше или ниже значения предопределенной приспособленности. Кроме того, взвешивание некоторых уровней может быть запрещено, чтобы обеспечить обновление определенных уровней. Топология компьютерной системы, используемая для определения приспособленности или

состояния путей, может быть введена программистом или независимо определена системой балансировки нагрузки.

Предыдущие устройства балансировки нагрузки связаны только с первым уровнем пути, и поэтому могут получить информацию только относительно состояния или приспособленности первого уровня путей, но не могут получить информацию относительно других уровней. Соответственно и маршрутизация происходит, учитывая только информацию, полученную с первого, физического, уровня. Устранение этих недостатков способом, описанным в данном патенте, обеспечивает более эффективное использование ресурсов каждого из путей, а поэтому более эффективную обработку информации в целом.

## **2. Иерархический диспетчер приоритетности для управления задачами в многопроцессорной компьютерной системе**

В [2] рассмотрена следующая проблематика. Функциями операционных систем (ОС) является отображение приоритетов процессоров для каждой новой задачи и если бы не некоторые условия, когда другим процессорам разрешают захватить задачи, этот приоритет остается неизменным. Иерархическая балансировка нагрузки отображена через матрицу приоритетности (которая может быть выражена как таблица), к которой обращается исполняемый код, доступный через диспетчера разнообразным процессорам команд (ПК) в многопроцессорной компьютерной системе. Так как компьютерная система имеет множественные уровни кэш-памятей, связанные шинами, и координирует оперативную память, иерархия, отображающая пары кэш-памяти, распределяет задачи сначала ПК, наибольшей вероятностью совместно использующим один и тот же остаток кэш-памяти приоритетными задачами, или с меньшей вероятностью, но с большей стоимостью времени доступа. Каждый ПК имеет свою собственную очередь переключения (ОП) для первично распределенной задачи, по которой ОС определяет начальную степень привилегии. Когда ОП ПК становится свободной от задач, код диспетчера ищет свободный ПК по ОП других ПК в соответствии с отображенной иерархией, если порог бездействия достигнут.

Здесь описаны три особенности управления диспетчеризацией или распределением задач на

процессоры команд в пределах многопроцессорной компьютерной системы. Назначение приоритетов отдельному ПК (или отдельному кластеру ПК) достигается использованием очереди переключения для каждого ПК (или кластером) компонентами операционной системы. Чтобы уменьшить перегрузку, которая может привести к несбалансированному распределению задач для обработки одиночными процессорами (или отдельными кластерами), используются переключения очередями других ПК, которые могут быть захвачены каждым ПК (или кластером) иерархическим способом, разработанным, чтобы обеспечить приоритет и времена доступа сквозь архитектуру многопроцессорной системы. Наглядно описано, какой уровень занятости опрошен сначала для каждого процессора, и затем для всей системы, перед формированием порогового значения, определяющего возможность перехода к операции захвата.

Таким образом, данная разработка позволяет повысить эффективность работы многопроцессорной системы за счет приоритетного иерархического (в плане использования памяти) распределения задач операционной системы.

## **3. Метод и система для балансировки и управления нагрузкой**

В [3] рассмотрена компьютерная система для web-узла, которая использует балансировку нагрузки новых сессий, основанных на измерениях и оценках прошлой, настоящей, и будущей производительности, используя среднее время, необходимое каждому серверу приложений, чтобы ответить на запрос (время ожидания) и число активных сессий на каждом сервере приложений. В большинстве случаев, какая-либо сессия назначается определенному серверу приложений, и в будущем на нем обрабатываются все запросы, касающиеся этой сессии. Однако, если нагрузка или производительность сервера приложений значительно ухудшается, failover-механизм последующие запросы передает для обработки другим серверам приложений, пока производительность не улучшится. При критических ситуациях, сервер приложений будет перегружен. Если web- или HTTP-сервер неспособны обслуживать запросы, каким-либо сервером приложений, последующие запросы могут быть переадресованы на другой web-сервер.

В разработках, описанных в данном патенте, используется набор вероятностей, определяющих выбор более или менее загруженного сервера приложений. Каждый сервер приложений содержит модуль управления нагрузкой.

Таким образом, можно сделать вывод, что в основе работы метода, предложенного в данном патенте, лежит система массового обслуживания (СМО).

#### **4. Планирование распределения подзадач процессора**

В [4] раскрыт метод для обработки процессором выполняемых задач и среда для выполнения в ней исполняемых команд процессора для осуществления этого метода, а также система для того, чтобы распределять работу обработки среди множества распределенных процессоров. Основная задача состоит в том, чтобы определить число узлов для оптимального решения той или иной задачи. При этом особое внимание уделяется определению характеристик распределяемых задач и ресурсов, на которые они будут распределены.

Расчет оптимального числа узлов основан на минимальном эффективном времени выполнения задачи.

#### **5. Система и метод для оптимизированного планирования задачи в гетерогенной системе обработки данных**

Рассмотрен метод, программный компьютерный продукт, и система обработки данных для оптимизации производительности решения задач в многопроцессорной системе [5]. Показатель производительности рассчитан на основании счетчика производительности, измеряющего характеристики задачи, выполненной на одном из множества процессоров определенной частоты, доступных в пределах многопроцессорной системы. Характеристики, снятые счетчиками производительности указывают, как на активность процессора, так и на активность памяти. Показатель производительности дает возможность, используя снятые данные на одной доступной частоте, предсказать производительность на другой частоте процессора, доступной в многопроцессорной системе. Потеря производительности минимизируется за счет назначения определенной задачи для выполнения на определенной частоте. Кроме того, в

данном изобретении предусмотрен механизм балансирования загрузки задач на основе приоритета, который минимизирует совокупную потерю производительности, принесенную выполнением всех задач в системе.

Разработанный планировщик является заменой для планировщика операционной системы, но также может использоваться, чтобы дополнить существующий планировщик.

Задачи, требующие значительных затрат ресурсов центрального процессора назначаются на процессор с более высокой частотой, в то время как задачи, требующие значительных ресурсов памяти назначаются на процессор, работающий на более медленной частоте.

Отношение требуемых ресурсов центрального процессора к ресурсам памяти задачи определяет точку насыщенности.

Показатель производительности используется для того, чтобы оценить время выполнения задачи  $t$  на определенной частоте  $f$  и отображает числовое значение производительности в командах в секунду, хотя его можно заменить и другой метрикой.

Подпрограмма балансировки, которая гарантирует минимизацию потери производительности, осуществлена как набор команд, исполняемых процессором, типичных для процессоров.

Таким образом, данное изобретение представляет собой метод, аппарат, и компьютерные команды для определения идеальных частот процессора для этапа выполнения приложений в многопроцессорной системе.

#### **6. Метод и система для балансировки нагрузки и вычислительных ресурсов между компьютерами**

Описывается метод и система балансировки нагрузки вычислительных ресурсов среди множества компьютеров [6], имеющих потребителей, использующих вычислительные ресурсы. Метод основан на том, что после определения более низкого порога уровня потребления ресурсов и получения уровня потребления ресурсов для каждого из потребителей и для каждого из указанных компьютеров в течение некоторого периода времени, он сравнивается с определенным для каждого из компьютеров более низким порогом. Всякий раз, когда компьютер, имеющий уровень потребления ресурсов выше, чем более определенный низкий порог, для ка-

ждого из потребителей определяется новое размещение вычислительных ресурсов. Вычислительные ресурсы необходимые потребителю тогда сдвигаются от их текущего местоположения на одном компьютере к соответствующему местоположению на другом компьютере, согласно планированию, таким образом, чтобы уровень потребления ресурса или ресурсов для компьютера уменьшился.

Работа балансировщика нагрузки, основана на сравнении. При этом не применяется оптимизация при выборе другого компьютера (перемещении ресурсов), основываясь на том, что менее загруженный компьютер, это уже лучше, чем более загруженный. Соответственно и не рассмотрен метод для решения задачи оптимизации.

#### **7. Динамическая балансировка нагрузки среди процессоров в параллельной компьютерной системе**

Описана параллельная система [7] осуществляет динамическую балансировку нагрузки для распределения рабочей нагрузки имеющимся процессорам в параллельной компьютерной системе. Препроцессор в системе конвертирует вложенную параллельную программу в последовательный программный код, выполняемый на процессорах параллельного компьютера, и вызывает MPI (Message Passing Interface) для межпроцессорной связи между процессорами. При обработке вложенной параллельной программы, препроцессор вставляет функцию тестирования, чтобы посчитать вычислительную стоимость функции вызова. Во время запуска процессоры оценивают при помощи функции тестирования, стоит ли пересылать функцию вызова на другой процессор. Эта технология дает возможность процессорам выгрузить функцию вызова на другие имеющиеся процессоры в случаях, когда это является более эффективным с учетом пересылки функции вызова, и получив результаты, потом передать функцию вызова первоначальному процессору.

Описано использование MPI как эффективного средства далеко не во всех случаях. Поэтому предлагается использовать язык Макиавелли, который поддерживает как последовательную компиляцию параллельных функций (удаляя конструкции MPI), так и отмену заданных по умолчанию функций с определяемыми

пользователем последовательными функциями, которые могут сделать алгоритм более эффективным. Язык Макиавелли разработан для облегчения переносимости программ на все архитектуры.

Для пользователя язык Макиавелли представлен как три главных дополнения к языку Си: векторная структура данных и связанные функции; параллельная конструкция данных, позволяющая напрямую создавать векторы; и параллельная конструкция управления, выражающая рекурсию в алгоритме «разделяй и властвуй».

#### **8. Метод и система для балансировки нагрузки вычислительного ресурса между компьютерами**

Данный патент [8] является продолжением патента [6], в котором производится так называемое «деление» всей поступающей работы в пределах домена. При этом существует отдельный элемент, балансировщик нагрузки, которому направляются все запросы на обслуживание. Он в определенном порядке циклически отсылает их каждому из компьютеров. Так как балансирование нагрузки охватывает множество серверов, то существует необходимость в их резервировании. Тогда запросы направляются группе серверов, выполняющих одинаковые функции, т.е. происходит дублирование. Всякий раз, когда на основном компьютере происходит сбой, резервный компьютер берет управление по IP, и начинает служить web-сервером. Различие в том, что резервный компьютер не обслуживает никаких запросов в то время, как он находится в резервном режиме. Кроме того, такая методика основана на дублировании поставщика услуг.

#### **9. Математические модели оптимального распределения информационных ресурсов среди узлов вычислительных сетей и методы их реализации**

В диссертации [9] построены математические модели оптимального распределения файлов среди узлов вычислительных сетей, которые в отличие от существующих не зависят от схемы обработки запросов в узлах. Доказана теорема для априорной оценки числа копий файлов распределенной базы данных, что уменьшает размерность задач математического

программирования, полученных вследствие построения моделей. Развита теоретическая база применения генетических алгоритмов в задачах математического программирования. Создана система поддержки принятия решений для расположения файлов на серверах локальных вычислительных сетей. Разработаны алгоритмы маршрутизации, в процессе использования ко-

торых не возникает перегрузки каналов связи в сетях передачи данных.

С целью предоставления большей наглядности проанализированных методов и сравнения их с предлагаемым решением построим таблицу, содержащую сравнительную характеристику объектов-аналогов (табл.1).

Таблица 1. Сравнительная характеристика объектов-аналогов

Характеристики Объект-аналог	Алгоритм, лежащий в основе	Объект оптимизации	Инструментарий
Стандартные средства ОС	СМО	Загрузка процессора	C, C++, Assembler
[1]	Сравнение с пороговым значением	Приспособленность пути	?
[2]	СМО	Используемая память	Компоненты ОС
[3]	СМО	Производительность	?
[4]	?	Число узлов	Команды процессора
[5]	СМО	Частота процессора	Компьютерные команды
[6]	Сравнение с пороговым значением	Уровень потребления ресурсов	?
[7]	Модифицированный алгоритм «разделяй и властвуй»	Вычислительная стоимость функции вызова	Язык Макиавелли
[8]	СМО	Число запросов на обслуживание	?
[9]	Задача о назначении в расширенной постановке	Маршрутизация пакетов	?
Разработанное решение	Модифицированная транспортная задача	Время	C++, Java

### Выводы

Проведенный анализ современных публикаций по проблемным вопросам управления вычислительными системами показал, что в основе большинства методов балансировки нагрузки лежит математический аппарат системы массового обслуживания. В качестве объекта оптимизации выбраны загрузка процессора, приспособленность пути, используемая память, производительность, число узлов, частота процессора, уровень потребления ресурсов, вычислительная стоимость функции вызова, число запросов на обслуживание, маршрутизация пакетов. Это еще раз подтверждает, что в целом проблема балансировки вычислительной системы является достаточно сложной, особенно с точки зрения представления математической модели. Отсюда возникает ряд частных новых задач в рамках проблемы:

- разработка математической модели задачи распределения заданий в вычислительной системе;
- поиск решения в рамках представленной модели;
- поиск более эффективных методов и алгоритмов распределения нагрузки в вычислительных системах;
- создание методик распределения нагрузки, математически обоснованных, и способов модернизации существующих систем управления ресурсами вычислительных систем.

Таким образом, проанализированные системы и методы дают основание считать, что представленная в патентах информация либо послужила уже на данный момент основой, либо еще станет базой для дальнейших разработок и усовершенствований в области управления ресурсами вычислительных систем.

## СПИСОК ЛИТЕРАТУРЫ

1. Патент №2281144 CA (A1), МПК7H04L12/56; H04L29/06. A device and method for balancing loads between different paths in a computer system [Электронный ресурс] /Ahmed Khalid [CA]; Xu Ming Q [CA]; Jones Rodney William [CA], заявл. 26.08.1999, опубл. 26.02.2001. – Режим доступа: <http://www.ep.espacenet.com>
2. Патент №6996822 US (B1), МПК8G06F 9/46. Hierarchical affinity dispatcher for task management in a multiprocessor computer system [Электронный ресурс] /Willen James W [US]; Merten James F [US], заявл. 01.08.2001, опубл. 07.02.2006. – Режим доступа: <http://www.ep.espacenet.com>
3. Патент №42908 WO, МПК7G06F9/50; G06F9/46; G06F9/00. Method and system for load balancing and management [Электронный ресурс] /Scott James Allan; Vroom Jeffrey J, заявл. 05.12.2000, опубл. 14.06.2001. – Режим доступа: <http://www.ep.espacenet.com>
4. Патент №2008021987 US (A1), МПК8G06F15/173; G06F15/16. Sub-task processor distribution scheduling [Электронный ресурс] /Bates John P [US]; White Payton R [US], заявл. 21.07.2006, опубл. 24.01.2008. – Режим доступа: <http://www.ep.espacenet.com>
5. Патент №2006168571 US (A1), МПК8G06F9/44. System and method for optimized task scheduling in a heterogeneous data processing system [Электронный ресурс] /Ghiasi Soraya [US]; Keller Thomas W JR [US]; Kotla Ramakrishna [US]; Rawson Freeman L III [US], заявл. 27.01.2005, опубл. 27.07.2006. – Режим доступа: <http://www.ep.espacenet.com>
6. Патент №20080320484 US, МПК7G06F9/50. Method and system for balancing the load and computer resources among computers [Электронный ресурс] / Yariv Rosenbach, Nir Baram, Gregory Bondar, Igal Harel; заявл. 27.08.2008 опубл. 25.12.2008. – Режим доступа: <http://www.ep.espacenet.com>
7. Патент №6292822 US, МПК7G06F9/00. Dynamic load balancing among processors in a parallel computer [Электронный ресурс] /Jonathan C. Hardwick, заявл. 30.06.1998 опубл. 18.09.2001. – Режим доступа: <http://www.ep.espacenet.com>
8. Патент №48748104 US, МПК7G06F 15/173. Method and system for balancing the load of a computer resource among computers [Электронный ресурс] /Rosenbach Yariv, Baram Nir, Bondar Gregory, Harel Igal, заявл. 26.07.2004, опубл. 02.12.2004. – Режим доступа: <http://www.depatistnet.dpma.de>
9. Математические модели оптимального распределения информационных ресурсов среди узлов вычислительных сетей и методы их реализации: Автореф. дис... канд. техн. наук: 01.05.02 /О.В. Демидович /Нац. ун-т "Львов. политехника". – Львов, 2001. – 20 с.

M.Yu. Yurich. **Analisis of load balancing systems and methods of computer systems**

М.Ю. Юрич. **Аналіз систем і методів балансування навантаження обчислювальних систем**

У даній статті проведено аналіз систем і методів балансування навантаження обчислювальних систем з урахуванням сучасної методичної та програмно-апаратної бази. Також показано сьогоденний стан даної проблеми і сформульовано нові та невирішені задачі у галузі ефективного розподілу завдань в обчислювальній системі.