

А.А. Нечай, А.И. Копьев

МЕТОД УПРАВЛЯЕМОГО РАСПРЕДЕЛЕНИЯ РЕСУРСОВ МЕЖДУ ЯДРАМИ ПРОЦЕССОРА

Статья посвящена проблемам распределения ресурсов и оптимальному распределению вычислительных возможностей процессора. Анализируются методы распределения нагрузки центрального процессора на примере операционной системы Windows. Представлен пример управляемого распределения процессов между ядрами под управлением операционной системы Windows.

Ключевые слова: нагрузка ядра, операционная система, Windows.

А.А. Nechay, A.I. Kop'yev

METHOD OF OPERATED RESOURCE ALLOCATION BETWEEN PROCESSOR'S CORES

The article is devoted to the problems of resource allocation and optimal allocation of processor's computing capacity. The analysis of load distribution methods for the CPU on an example of the Windows operating system is presented. The example of operated distribution processes between cores with the Windows operating system is presented, too.

Keywords: core load, operational system, Windows.

1. Актуальность

В настоящее время программное обеспечение (ПО) отстаёт от аппаратного. Существующие компоненты ПО и утилиты не способны реализовать весь потенциал представленного на рынке аппаратной составляющей электронных вычислительных машин (ЭВМ). Это приводит к большому времени отклика системы на действие пользователя, если рассматривать персональные машины (ПЭВМ), или долгому ответу серверов на запрос от клиента, если рассматривать web-технологии. Для повышения производительности было принято разделять центральное процессорное устройство (CPU) на ядра и тем самым распределять задачи ЭВМ между ядрами CPU.

2. Постановка задачи

Целью исследования являются алгоритмы распределения нагрузки процессора, методы и способы оптимизации и равномерного распределения задач. В данной статье рассмотрены существующие методы распределения нагрузки между ядрами процессора. За основу расчётов взяты алгоритмы оптимизации пользовательских ЭВМ. Так как современное системное ПО по тем или иным причинам не использует все возможные ресурсы, перед разработчиками стоит задача уменьшить разрыв в отставании ПО от аппаратной составляющей ЭВМ, увеличить производительность программного средства за счёт передачи ему большего числа ресурсов, с которыми программа быстрее справится с задачей. Из этого следует цель данной работы – увеличение производительности вычислительных систем посредством перераспределения нагрузки между ядрами.

© Нечай А.А., Копьев А.И., 2018.

3. Результат работы

Результатом оценки существующих методов является экспериментальный образец программного комплекса ПКРН. Ключевое различие между ПК и встроенным механизмом – распределением нагрузки – занимается оператор ЭВМ. Программа может стать альтернативой механизму распределения нагрузки, встроенного в ОС. Характеристика исследуемого процессора представлена на рис. 1.

```
Ваш CPU:  
Название CPU - Name  
AMD Turion(tm) X2 Dual-Core Mobile RM-74  
Версия процессора - Version  
Модель 3, выпуск 1  
Частота процессора <МГц> - CurrentClockSpeed  
2200  
Максимальная частота процессора <МГц> - MaxClockSpeed  
2200  
Количество ядер - NumberOfCores  
2  
Количество логических ядер - NumberOfLogicalProcessors  
2
```

Рис. 1. Характеристика исследуемого процессора

Программа представляет собой интерпретатор консоли с расширенными возможностями. Алгоритм работы ПК представлен на рис. 2. ПКРН обладает достаточно простым алгоритмом и высокой эффективностью. На рис. 4 представлен результат времени после запуска 10 приложений «Калькулятор» с помощью встроенного механизма ОС и с помощью ПКРН.

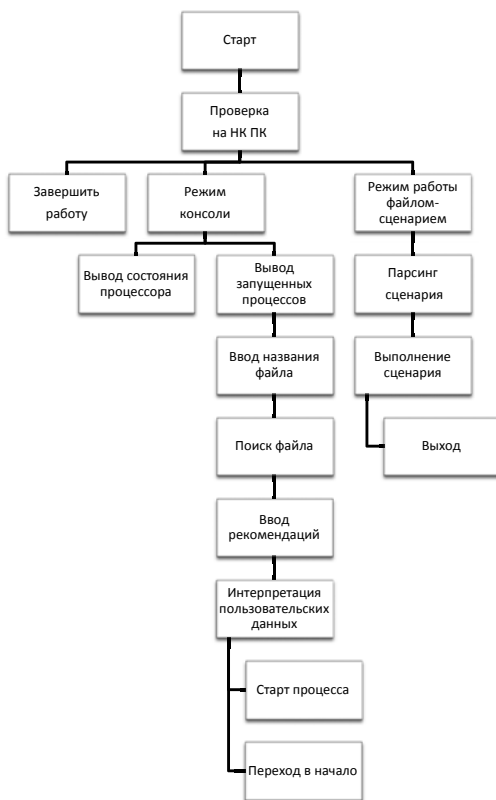


Рис. 2. Алгоритм работы ПК

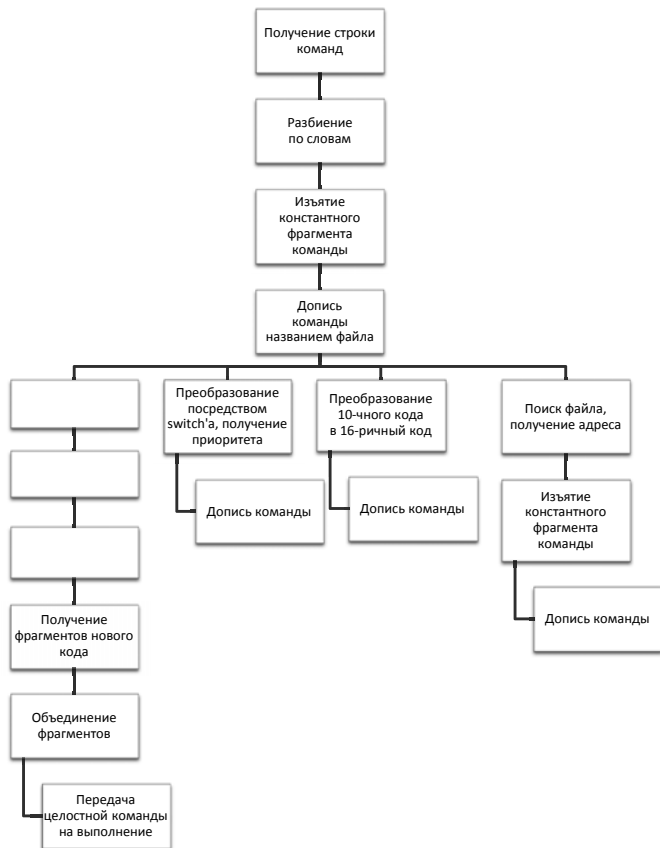


Рис. 3. Работа встроенного интерпретатора

Встроенный интерпретатор на вход получает пользовательскую строку, а на выход передаёт полноценную команду.

Опыты показали, что после запуска приложений через ПКРН производительность отдельного приложения увеличилась. В качестве теста была написана отдельная программа, выполняющая логические операции, выводит время, потраченное на их выполнение и формирует лог-файл для наглядности. На рис. 4 представлен итог теста.

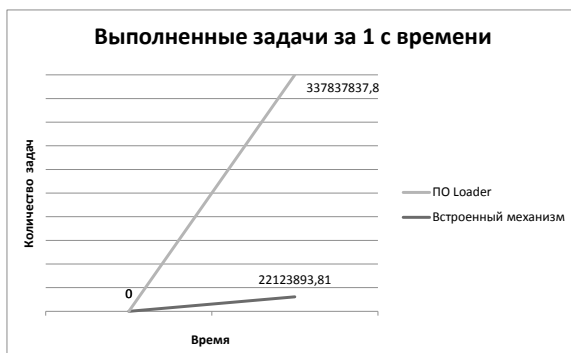


Рис. 4. Итог теста

Как показали тесты, производительность вычислений возросла в 1,5 раза. Программа позволяет запускать процессы со следующими приоритетами:

- 1) обычный;
- 2) ниже нормального.

На рис. 5 представлен результат запуска 10^8 вычислений с плавающей точкой с разным приоритетом.

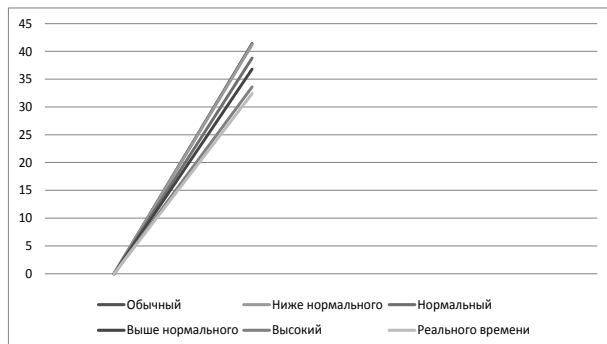


Рис. 5. Результат запуска 10^8 вычислений с плавающей точкой

На графике видно, что с ростом приоритета вычислений время вычислений уменьшается.

ПКРН обладает функцией управления тактовой частотой процессора. На рис. 6 представлен внешний вид модуля управления.

```
MaxClockSpeed
2200

CurrentClockSpeed
2200

Set clock speed
1- Low
2- High
3- Balance
Other- Refresh
!!- Exit
```

Рис. 6. Модуль предоставляет право выбора режима работы:

- Low – низкая частота;
- High – максимально допустимая частота;
- Balance – сбалансированный режим

Для удобства модуль отображает частоту, с которой ЭВМ функционирует в данный момент, и максимальную частоту, с которой он может функционировать. Low, или низкое, значение тактовой частоты процессора определяется как минимальная частота, с которой ЭВМ может функционировать. High, или высокое, значение тактовой частоты процессора определяется как максимально допустимая частота, с которой может работать ЭВМ. Balance, или сбалансированный, режим работы означает, что произойдет подсчет нагрузки ЦП и будет сделана выборка тактовой частоты процессора по принципу «минимально необходимая» для решения поставленных задач. На рис. 7 представлена зависимость производительности от тактовой частоты ЦПУ. Для этого были запущены 10^8 вычислений с плавающей точкой.

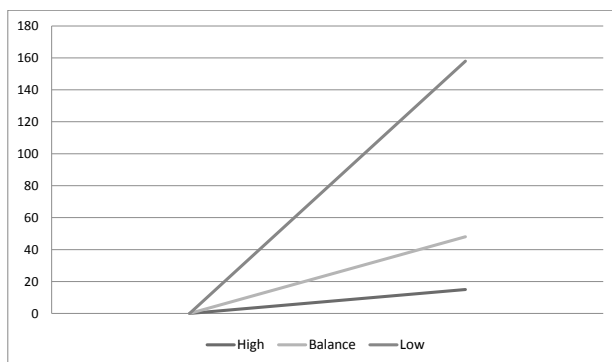


Рис. 7. Зависимость производительности от тактовой частоты ЦПУ

Из графика видно, что при снижении тактовой частоты ЦПУ происходит увеличение времени обработки вычислений и снижение производительности.

ПКРН обладает встроенным механизмом защиты от несанкционированного копирования. В момент установки программа анализирует среду, на которую происходит установка, и привязывается к ней. В последующем, при запуске, программа проверяет среду и выдает отказ в случае несовпадения с эталонным слепком. Работа данного модуля представлена на рис. 8.

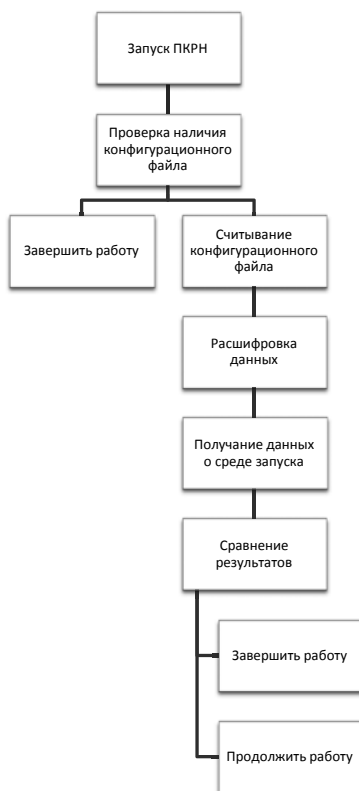


Рис. 8. Пример работы разработанного модуля

ПКРН использует метод необратимого шифрования для большей надёжности. Для каждого символа получается индивидуальный ключ, а процесс расшифровки происходит посредством шифрования алфавита.

Для автоматизации управления в программе предусмотрена работа в режиме Сценария. Отдельный файл с расширением .kai является Сценарием работы для ПКРН. В данный файл заранее записываются все рекомендации, которые в будущем следует выполнить. В момент запуска файл с расширением .kai регистрируется ПКРН, что вызывает произвольный запуск программы, которая, в свою очередь, считывает данный сценарий, интерпретирует его и выполняет его требования.

Модуль регистрации представляет собой небольшую программу, которая при запуске собирает информацию о среде запуска, зашифровывает полученные данные и записывает её в конфигурационный файл. Шифрование необходимо для предотвращения подделки документа, представляющего собой эталонный слепок, рис. 9.



Рис. 9. Пример работы модуля регистрации

Самоуничтожение модуля необходимо для предотвращения размножения и дальнейшего бесконтрольного распространения ПКРН. Иными словами, данная программа является одноразовой и распространяется по принципу 1 программа – 1 машина.

4. Вывод

В ходе проведенных исследований был разработан метод управляемого распределения ресурсов между ядрами процессора, реализованный в специализированном программном комплексе, обеспечивающем перераспределение нагрузки между ядрами процессора. При разработке были учтены основные недостатки существующих механизмов распределения нагрузки. Результатом является ПКРН, способный оптимизировать работу ОС. Работа ПК основана на перераспределении нагрузки и распараллеливании выполняемых задач между ядрами, за счёт чего увеличивается производительность вычислительного средства. Встроенная функция рекурсивного поиска обеспечивает простоту использования и быстроту интеграции ПК в ЭВМ. Тестирование ПК ПО Loader позволили убедиться в эффективности концептуальных положений, на которых основана данная работа.

Литература

1. *Новиков А.Н.* Математическая модель обоснования вариантов реконфигурации распределенной автоматизированной контрольно-измерительной системы / А.Н. Новиков, А.А. Нечай, А.В. Малахов // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». – 2016. – № 1–2. – С. 56–59.

2. *Нечай А.А.* Методика повышения надежности функционирования систем, организованных на перепрограммируемых элементах / А.А. Нечай, П.Е. Котиков // Вестник Российского нового университета. Серия «Сложные системы: модели, анализ и управление». – 2016. – № 1–2. – С. 87–89.

References

1. *Novikov, A.N.* Matematicheskaya model' obosnovaniya variantov rekonfiguratsii raspredelennoy avtomatizirovannoy kontrol'no-izmeritel'noy sistemy / A.N. Novikov, A.A. Nechay, A.V. Malakhov // Vestnik Rossiyskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". – 2016. – № 1–2. – S. 56–59.

2. *Nechay, A.A.* Metodika povysheniya nadezhnosti funktsionirovaniya sistem, organizovannykh na pereprogrammruemykh elementakh / A.A. Nechay, P.E. Kotikov // Vestnik Rossiyskogo novogo universiteta. Seriya "Slozhnye sistemy: modeli, analiz i upravlenie". – 2016. – № 1–2. – S. 87–89.