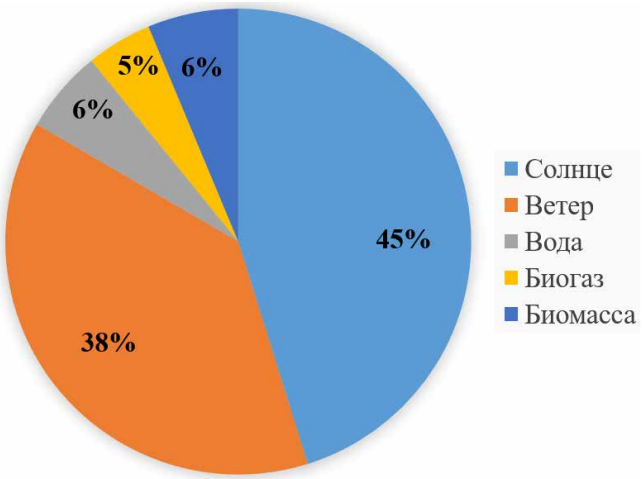

Рис. 1. Процентное соотношение различных технологий в производстве электроэнергии возобновляемыми источниками в Республике Беларусь

Гидроэнергетика республики имеет ограниченный потенциал, но есть возможность ее развития за счет малой энергетики в центральных и северных регионах страны. По оценкам экспертов гидропотенциал республики оценивается в 850 МВт.

Ветропотенциал республики оценивается величиной до 1 600 МВт, при этом можно построить 1840 ветряных электростанций. Существует 108 установок по получению энергии ветра, расположенных в Минской, Могилевской и Гродненской областях. Для Беларуси характерна низкая скорость ветра, однако с учетом создания турбин, рассчитанных на эту скорость, возможен пересмотр указанных показателей.

Солнечная энергия является наиболее перспективных направлением в развитии ВИЭ, поскольку потенциал этой энергии достаточно высок, особенно на юге и юго-востоке республики. Солнечная энергия может быть получена как за счет фотоэлектрических систем, так и путем использования гелиоколлекторных установок. Солнечный потенциал оценивается в 49,7 млн т н.э. / год.

Оценивая сложившиеся энергетические и экономические особенности страны и имеющиеся в Беларуси энергетический потенциал ВИЭ, можно прийти к заключению, что в ближайшей перспективе возобновляемая энергетика не может рассматриваться в качестве реальной альтернативы углеводородному топливу. В то же время очевидна необходимость развития ВИЭ в качестве элемента стратегии энергетической независимости республики. Однако с учетом развития атомной энергетики и соответствующим существенным сокращением потребления газа целесообразно постепенное замещение его возобновляемыми источниками энергии. Деятельность по развитию ВИЭ, которые будут постепенно заменять углеводородные виды топлива, надо рассматривать как необходимую долгосрочную перспективу увеличения энергетического потенциала республики,.

ЛИТЕРАТУРА

1. Цилибина В. М. Возобновляемая энергетика становится самым быстро развивающимся видом генерации [Электронный ресурс]. – Режим доступа: https://belchemoil.by/news/analitika/vozobnovlyaemaya-energetika-stanovitsya-samym-bystro-razvivayushhimsya-vidom-generacii. – Дата доступа: 24.03.2021.

2. Renewable Energy Sources and Climate Change Mitigation. Special Report of the Intergovernmental Panel on Climate Change. – Intergovernmental Panel on Climate Change, 2012. -1088 p.

3. Ископаемое топливо: преимущества и недостатки [Электронный ресурс]. – Режим доступа: http://electrowelder.ru/index.php/news/55-energetics/1480-fossil-fuel.html. – Дата доступа: 24.03.2021.

4. The European Power Sector in 2020. An analysis by Agora Energiewende and Ember. – 2021 – 18 p.

5. Belarus energy profile [Электронный ресурс]. – Режим доступа: https://www.iea.org/reports/belarus-energy-profile. – Дата доступа: 24.03.2021.

ИССЛЕДОВАНИЕ И СРАВНЕНИЕ ОПТИМИЗАЦИИ КОМПЬЮТЕРНЫХ ПРОГРАММ
RESEARCH AND COMPARISON OF OPTIMIZATION OF COMPUTER PROGRAMS

Л. А. Липницкий, П. К. Шалькевич, М. А. Трейвас, Е. П. Черевань
L. A. Lipnitski, P. K. Shalkevich, M. A. Treyvas, E. P. Cherevan

Белорусский государственный университет, МГЭИ им. А. Д. Сахарова
г. Минск, Республика Беларусь gloomyhit@gmail.com
The Belarusian State University ISEI BSU, Minsk, the Republic of Belarus

Улучшение быстродействия создаваемого программного продукта является одним из важнейших моментов в информационной индустрии. Оптимальные значения скорости работы программы достигаются при помощи различных методов оптимизации кода. В этой работе представлены основные пути решения данной проблемы.

Improving the performance of the created software product is one of the most important moments in the information industry. Optimal values of the speed of the program are achieved using various methods of code optimization. This paper presents the main ways to solve this problem.

Ключевые слова: оптимизация, быстродействие, создание программ, информационные технологии.

Keywords: optimization, performance, program creation, information technologies.

https://doi.org/10.46646/SAKH-2021-2-302-305

При создании программы часто возникает проблема в выборе референтного метода оптимизации. Потоки выполнения – наименьшая единица обработки информации компьютером. Поток выполнения находится внутри выполняемого процесса. Несколько потоков выполнения могут быть реализованы в пределах одного и того же процесса и вместе использовать оперативную память компьютера. Такое явление называется многозадачностью. С помощью данной функции достигается увеличение производительности компьютера.

Конкурентность– это свойство программы или компьютера, заключающееся в одновременном выполнении нескольких процессов. Конкурентность в программирования реализована с помощью установления определенных управляющих потоков. С их помощью остальные процессы избегают ожидания завершения всех остальных вычислений – как это реализуется в последовательном программировании.

Параллельное исполнение является подмножеством конкурентного исполнения и предполагает наличие нескольких вычислительных устройств, которые будут одновременно выполнять множество задач.

Параллельные вычисления задействуют несколько вычислительных ядер процессора. Из-за одновременной работы всех управляющих потоков и задействования всего рабочего цикла ядра за время исполнения параллельные вычисления невозможно реализовать с помощью одноядерного процессора. В этом и состоит основное отличие от конкурентных вычислений, которые сконцентрированы на пересечениях основных циклов вычислений. Этапы выполнения самого процесса могут быть разделены на временные промежутки - если процесс не прерывается до конца временного промежутка, то он предоставляет другому процессу возможность начать вычисление. Принцип конкурентного вычисления обычно реализован в качестве многопоточности, которая позволяет использовать преимущества параллелизации, т.е. быстро и эффективно использовать ресурсы компьютера.

Параллельные вычисления – метод формирования вычислений компьютера, позволяющий программам разрабатываться в качестве комплекта взаимодействующих вычислений, функционирующих одновременно. Данный термин не только включает в себя комплекс задач для разработки программ, но и создает эффективно манипулирующие аппаратные исполнения. Данная теория включена в отдельную область утилитарной теории программных алгоритмов.

Существует много способов осуществления параллельных расчетов. Это может быть любая вычислительная операция, которая реализована в качестве процесса операционной системы. Вычисления могут реализовываться в качестве набора потоков в одном процессе операционной системы. Параллельные утилиты могут иметь вещественное исполнение или выполняться последовательно на одном вычислительном блоке, реализуя поэтапно шаги дня осуществления каждого операции, или параллельно предоставляя всем процессам отдельный вычислительный блок, подключенный в одну сеть.

Главной проблемой при разработке программ, основанных на параллельных вычислениях, является обеспечение последовательности взаимосвязей между разными процессами, а также регулированием ресурсов, выделяемых для каждого из них.

В определенных параллельных системах передача информацией между параметрами не видна разработчику, в иных должна показываться отчетливо. Явные взаимодействия между компонентами можно разделить на три типа:

Явное взаимодействие с помощью разделяемой памяти. Процессор запускает каскад исполнения, принадлежащий процессу. Каждый поток передает данные другим потокам с помощью общего участка оперативной памяти. Потоки могут создаваться посредством языков программирования, библиотек, декларативно или автоматически, используя встроенные возможности компилятора. Данный вид одновременного программирования нуждается в форме передачи управления для согласования общей работы потоков.

Явное взаимодействие через обмен сообщениями. Для каждого процессора включается один процесс, делящийся информацией с остальными процессами, выполняющимися на остальных процессорах. Передача информации осуществляется библиотекой или языком программирования. Обмен сообщениями между процессами может осуществляться асинхронно, или используя метод «рандеву», временно блокирующий отправителя до того момента, когда его сообщение не придет получателю.

Системы с разделяемой оперативной памятью считаются более сложными для понимания, нежели параллельные системы, которые базируются на передаче сообщений, и обычно считаются более плохим методом для параллельного программирования. Обмен сообщениями между несколькими системами может быть выполнен на процессорах даже без разделяемой памяти.

Гибридный способ взаимодействия между компонентами заключается в запуске на всех существующих узлах системы многопоточного процесса, распределяющего потоки этого узла между всеми процессорами. Через общую для всех процессов оперативную память производится обмен данными на узле в этом потоке, а передача сообщений осуществляется за счет обмена информацией между всеми узлами. Количество узлов определяет количество процессов, а общее количество процессоров определяет все количество потоков.

Многопоточность – это метод расчета, являющийся свойством программы, который заключается в параллельном выполнении нескольких составных задач, не мешая реализации друг другу и без какого-либо временного порядка. При расчете больших задач подобное разделение общего массива на составные части позволит использовать ресурсы компьютера наиболее эффективно. Главной задачей многопоточности является выполнение всех потоков в пространстве процесса, но реализующийся процесс имеет один или несколько главных потоков.

Многопоточная реализация какой-либо системы упрощает вычисления за счёт манипуляций общим адресным пространством, уменьшения времени на создание потока, вынесения чередования в выполнения взаимосвязанных задач, распараллеливания вычислений. Основная особенность данных процессов заключается в меньших временных затратах на начало операции, чем на основную работу. Это позволяет даже на устройствах с небольшими вычислительными мощностями выполнять множество асинхронных операций параллельно.

Just in time-компиляция расшифровывается как динамическая компиляция – набор методов для увеличения производительности программ за счет преобразования кода программы в машинный код прямо во время непосредственной работы самой программы. Данная операция позволяет повысить скорость выполнения за счёт уменьшения потребления оперативной памяти и как следствие затрат времени на выполнение расчетов. Данный алгоритм базируется на динамической компиляции байт-кода.

JIT-компиляция позволяет также применять динамическую рекомпиляцию и адаптивную оптимизацию, благодаря чему данный вид компиляции демонстрирует наиболее высокие результаты, нежели статическая компиляция.

JIT-компиляция также может применяться и к отдельной части выполняемой программы.

JIT-компиляция имеет определенную структуру, предоставленную на рис.1.

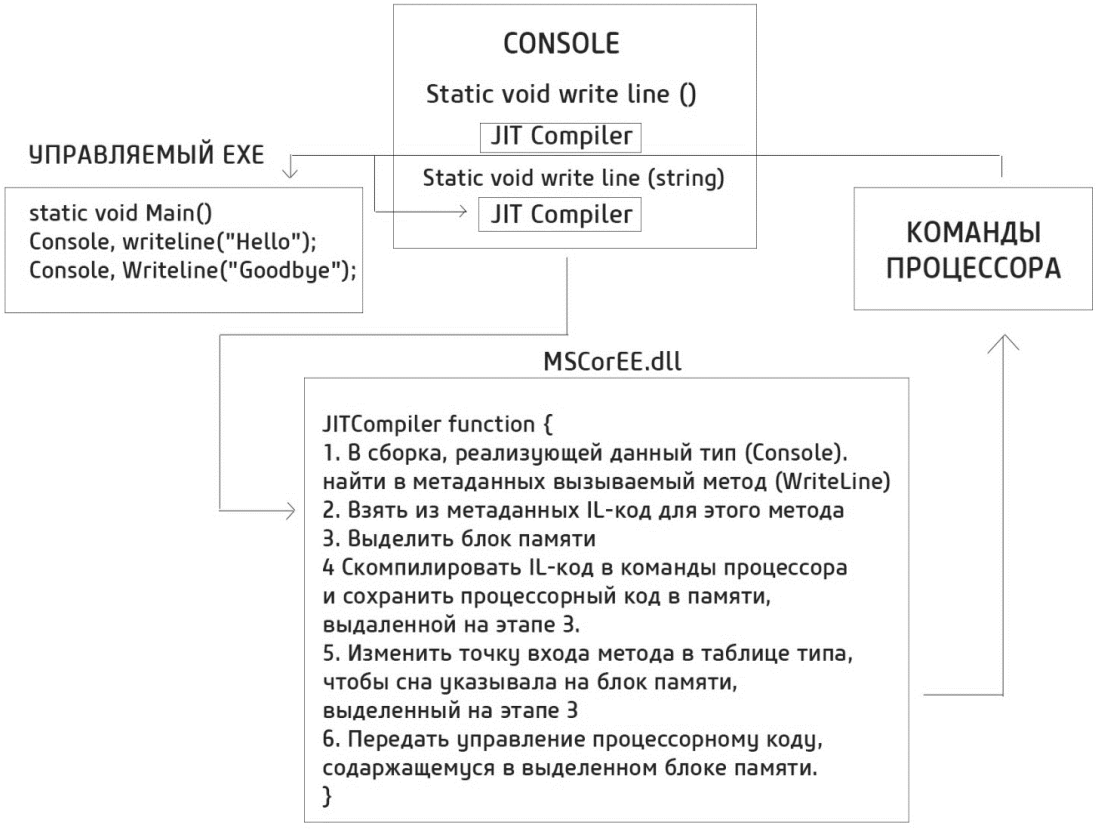

Рисунок 1 – Принцип работы Just-in-time compilation

Динамически компилируемая среда подразумевает собой среду, компилятор в которой вызван в момент непосредственного расчета. Данная функция может скомпилировать динамически сгенерированный код.

Приоритетная задача использования JIT – достижение высокой производительности за наименьшее количество времени, оставляя достоинства динамической компиляции.

Компиляторы, занимающиеся превращением исходного кода в машинный код, намного проще в исполнении, так как большую часть методов по улучшению было реализовано самим компилятором.

JIT-компиляция эффективней, нежели обычная интерпретация исходного кода. JIT демонстрирует высокую производительность, нежели статическая компиляция, из-за оптимизаций, реализуемых исключительно во время работы самой программы.

Компиляция реализуется только для компьютера, на котором включено исполняемое приложение.

Среда собирает данные о выполняющейся программе, а затем производит ее оптимизацию. Почти все компиляторы используют информацию и данные о предшествующих активациях данного приложения.

Среда способна масштабно оптимизировать исходный код без явных потерь достоинств динамического компилирования. Наиболее легкая перестройка кода для наиболее эффективного использования кэша.

JIT содержит в себе компиляцию исходного кода в машинный код, а также реализацию этого кода. Результат поступает в оперативную память и исполняется в тот же момент, без переходного сохранения или запуска в качестве встроенной программы. Сейчас для сложных систем ради улучшения устойчивости и надежности не все участки памяти н используются в качестве машинного кода. Ради безопасного включения все части памяти должны быть заранее отмечены в качестве выполняемых, но для повышения уровня безопасности показатель выполнения осуществим только после удаления показателя допущения записи.

Представленные методы позволяют работать с большими файлами, приложениями, рассчитывающими большие объемы данных, или постоянно возникающими запросами с наименьшими затратами времени и наиболее эффективным использованием ресурсов вычислительной машины.

ЛИТЕРАТУРА

1. Основные принципы программирования: конкурентность [Электронный ресурс]. – Режим доступа: https://tproger.ru/translations/programming-concepts-concurrency. – Дата доступа: 21.03.2021.

2. Синхронность и асинхронность процессов [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/453192. – Дата доступа: 21.03.2021.

3. Многопоточность [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C. – Дата доступа: 07.03.2020.

4. Асинхронность (компьютерное программирование) [Электронный ресурс]. – Режим доступа: https://ru.qaz.wiki/wiki/Asynchrony_(computer_programming). – Дата доступа: 21.03.2021.

СНИЖЕНИЕ ЭНЕРГОЕМКОСТИ ПРОЦЕССА ПРОИЗВОДСТВА ПИВОВАРЕННОГО СОЛОДА

DECLINE OF POWER-HUNGRYNESS OF PROCESS OF PRODUCTION OF BREWING MALT

В. А. Пашинский[1], О. В. Бондарчук[2]

V. A. Pashynski[1], O. V. Bondarchuk[2]

[1]Белорусский государственный университет, МГЭИ им. А. Д. Сахарова БГУ, г. Минск, Республика Беларусь, pashynski@mail.ru
Belarusian State University, ISEI BSU, Minsk, Republic of Belarus

[2]Белорусский государственный аграрный технический университет, г. Минск, Республика Беларусь,
Belarusian State Agrarian Technical University, Minsk, Republic of Belarus

Приведены результаты исследования расхода энергии на сушку светлого солода при обработке пивоваренного ячменя неоднородным электрическим полем. По сравнению с солодом, зерно которого не подвергалось обработке, расход энергии меньше на 0,03 Гкал, что составляет 4,286 кг у.т. в пересчете на 1 т исходного зерна влажностью 14%.

Research results are resulted expense of energy on drying of light malt at treatment of brewing barley the heterogeneous electric field less than on 0,03 Gkal, that makes 4,286 kg of u.t. in a count on 1 t of initial grain by humidity 14% as compared to a malt grain of which was not exposed to treatment.

Ключевые слова: пивоваренный ячмень, напряженность электрического поля, солод, влагопоглощение, сушка, энергоемкость сушки, экстракт солода.

Keyword: brewing barley, tension of the electric field, malt, dehumidification, drying, power-hungryness of drying, extract of malt.

https://doi.org/10.46646/SAKH-2021-2-305-308