

**УЛУЧШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ «УЗКИХ» МЕСТ
КОМПЬЮТЕРНОГО КОДА НА ОСНОВЕ ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНОГО
ПРОГРАММИРОВАНИЯ**

Рудзейт О.Ю. (Дальневосточный федеральный университет), **Зайнетдинов А.Р**
(Дальневосточный федеральный университет), **Недяк А.В.** (Дальневосточный федеральный
университет)

Научный руководитель – ассистент кафедры ИСУ Пашин С.С. (Дальневосточный
федеральный университет)

На данный момент эффективность исполнения некоторых программ по времени играет очень большую роль, например, при программировании контроллеров для роботов или выполнении каких-либо операций в режиме реального времени. Если исполнение программы будет очень долгим, то это может привести к непредсказуемым результатам вплоть до полного краха системы. В данной статье описываются технологии многопоточности для улучшения эффективности по времени частей программного кода, которые исполняются достаточно долго для того, чтобы задержать процессы обработки информации, что может привести к отрицательным последствиям, а также приводится статистка по улучшению производительности в зависимости от количества параллельных потоков функции и сложности исполнения компьютерного кода.

Введение. Компьютерные технологии в совокупности с мощным аппаратным обеспечением способны достаточно быстро исполнять программные инструкции в доли секунды, но с увеличением сложности программ время исполнения их увеличивается до секунд или десятков секунд, что может привести к задержке работы системы или привести к ее отказу. Например, робот не успеет повернуть в нужный момент, исходя из поставленной траектории, перевернуться и откажет в работе. Существуют различного рода компиляторы, оптимизирующие код программ или другие языки программирования, которые работают быстрее по сравнению с другими языками, например С, С++, но и этого может быть недостаточно. За рубежом из-за развития аппаратной части компьютеров, появления многоядерных процессоров и технологий многопоточности в 2007 году было принято решение о том, что производительность можно улучшить, используя сразу несколько логических и физических ядер процессора, распределив выполнение вычислений по нескольким параллельным потокам.

Основная часть. Для уменьшения времени выполнения программы используют технологию распараллеливания выполнения отдельных блоков программы на несколько потоков. В данном случае за основу берется библиотека OpenMP, которая хорошо работает с языками С и С++ и встроена в среду разработки Microsoft Visual Studio. Она представляет описание совокупности директив компилятора и библиотечных процедур, предназначенных для воссоздания многопоточных программ на процессорах с большим количеством ядер и с общей доступной памятью.

Данная библиотека использует модель параллельного выполнения «ветвление-слияние». Код программы, используя библиотеку OpenMP, начинается как единственный поток выполнения, называемый «главным» потоком. Когда начальный поток встречает параллельную конструкцию, он создает новую группу потоков, состоящую из себя и некоторого числа дополнительных потоков, и становится главным в новой группе. Все члены новой группы выполняют код внутри параллельной конструкции. В конце параллельной конструкции имеется неявный барьер. После параллельной конструкции выполнение пользовательского кода продолжает только главный поток. Предполагается, что потоки выполняются параллельно на машине с несколькими процессорами или ядрами одного и того же процессора. При подключении данной технологии программист не обременяется

дополнительными сложностями, связанными с созданием, синхронизацией, балансировкой нагрузки и уничтожением потоков.

На сегодня активно ведутся разработки в области многопоточных приложений и на базе другого языка такого, как Golang, где уже встроена возможность распараллеливания вычислений без каких-либо дополнительных библиотек на основе создания горутин, которые не требовательны к ресурсам компьютера. Также разработки ведутся для языков C# и Java с их библиотеками Thread, которые работают приблизительно одинаково, но требовательнее к ресурсам компьютера.

Выводы. Библиотека OpenMP является достаточно простым инструментом для распараллеливания вычислений и уменьшения времени исполнения программы, которая снимает нагрузку с программиста, беря на себя функции создания, синхронизации и уничтожения потоков. Проведенные эксперименты с использованием данной технологии на вычислениях, производимых с помощью массива с несколькими десятками тысяч элементов позволяют уменьшить время выполнения приблизительно на 20%, распределяя нагрузку на несколько потоков.

Данная технология из-за своей простоты может быть встроена в исполняемый код роботов, бортовые системы самолетов в системы онлайн – заказов и т.д.