

ОЦЕНКА ЭФФЕКТИВНОСТИ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗАЯВКАМИ НА ОСНОВЕ ПРОГРАММНОГО ЭКСПЕРИМЕНТА

Нагиев А.Е.

Научный руководитель: Ботыгин И.А.
Томский политехнический университет
andrew_nagiev09@mail.ru

Введение

Последние годы все более активно развиваются распределенные вычислительные системы (РАС). Такие системы представляют собой объединение территориально разнесенных компьютеров в единую вычислительную систему с целью реализации параллельных вычислений [1-4]. Таким образом, РАС неразрывно связаны с такими понятиями, как многозадачность, параллельность, синхронизация выполняющихся процессов, балансировка нагрузки и т.п.

Главным преимуществом распределенных вычислительных систем является то, что в них имеется возможность неограниченного наращивания вычислительной мощности, как минимум, за счет горизонтального масштабирования ресурсов.

Однако, разработка таких систем влечет за собой ряд проблем системного характера – необходимо обеспечить настройку инфраструктуры системы для выполнения определенной задачи, обеспечить эффективное распределение информационно-вычислительных ресурсов, управление распараллеливанием вычислений и многое другое, что определяется не только конкретными характеристиками (быстродействие, емкость и т.д.) устройств различных уровней иерархии в системе, но и особенностями реализуемой вычислительной задачи.

Целью настоящей работы является разработка системы управления потоком, поступающих на обработку заявок, обеспечивающей эффективное использование информационно-вычислительных ресурсов.

Функциональная структура системы управления потоками заявок

В функциональной структуре системы управления выделено несколько уровней иерархии:

Первый уровень:

- имитатор терминалов запросов, который генерирует запросы на обработку в отдельных потоках операционной системы. Необходим для рабочего тестирования разработанной системы управления заявками на вычисления (СУЗ).

Второй уровень:

- основной центр управления (ОЦУ), задача которого заключается в обработке полученных запросов;

- буфер запросов, задача которого заключается в накоплении необработанных в ОЦУ запросов;

- диспетчер распределения нагрузки, который предназначен для перераспределения выполнения накопившихся запросов из буфера между дополнительными центрами управления (ДЦУ);

- дополнительный центр управления, который аналогично ОЦУ, обрабатывает запросы.

Взаимодействие между уровнями иерархии СУЗ происходит с использованием интерфейса сокетных соединений.

При небольшом количестве поступающих запросов обработку производит только ОЦУ, т.к. его мощностей достаточно. В отдельном потоке ОЦУ «слушает» поступающие запросы. Адрес соединения для каждого терминала запросов записывается в буфер основного центра управления, который удаляет данное подключение после обработки запроса и решения поставленной задачи. Далее в ОЦУ запускается отдельный поток на обработку поступившего запроса (рис. 1). Запросы обрабатываются в порядке очереди.

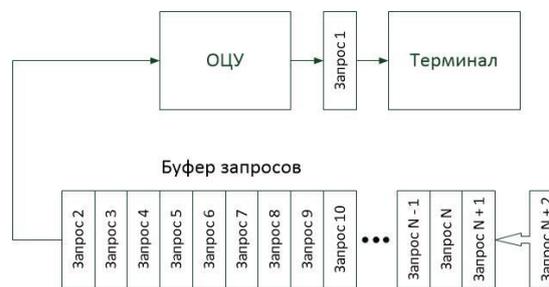


Рисунок 1 - Обработка ОЦУ поступающих запросов

С увеличением количества подключаемых терминалов и запросов на обработку, ОЦУ начинает не справляться с обработкой и тормозит всю систему. При достижении определенного количества запросов, указываемого в настройке, которые ожидают обработку в ОЦУ, но еще не начали выполняться, начинает работу диспетчер распределения нагрузки.

Диспетчеры распределения нагрузки и ДЦУ являются дополнительными вычислительными мощностями, которые подключаются к ОЦУ.

Диспетчер динамически подключается к ОЦУ и запускает столько дополнительных центров управления, сколько необходимо, чтобы увеличить быстродействие распределенной вычислительной системы до требуемого уровня.

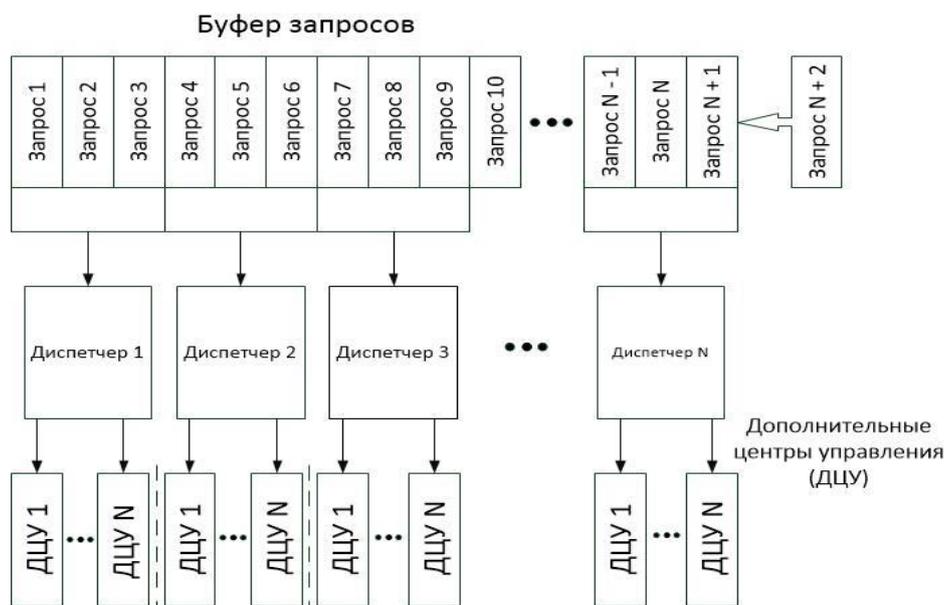


Рисунок 2 - Структурная схема выборки запросов диспетчерами обработки из буфера и отправка их на ДЦУ

Применение многопоточности

В СУЗ явно используется возможность многопоточных вычислений для дополнительного обеспечения эффективности управления потоком, поступающих на обработку заявок.

Диспетчеры распределения создаются в отдельных потоках и их количество определяется только вычислительной мощностью инфраструктуры распределенной системы (рис. 2).

Каждый диспетчер работает с буфером, в котором хранится очередь запросов на обработку. Каждый ДЦУ в диспетчере также запускается в отдельном потоке. Диспетчер отправляет определенное количество запросов из буфера в ДЦУ, которые обрабатываются последовательно. После обработки полученных от диспетчера запросов ДЦУ отключается.

Еще одним преимуществом предложенной СУЗ является то, что программно можно задать любое количество запросов, которое будет выбираться из буфера на обработку каждым из диспетчеров. Это важно для регулирования производительности системы.

Программный эксперимент

В качестве среды для программных экспериментов был выбран объектно-ориентированный язык программирования Java.

Эксперименты проводились при подключении имитатором различного числа терминалов запросов (от 10 до 10 000) с непрерывным равномерным распределением на интервале от 0 до 100 условных временных единиц. Основные программные компоненты СУЗ разработаны в соответствии со структурной схемой выборки запросов диспетчерами обработки из буфера и отправка их на ДЦУ (рис. 2).

Работа СУЗ моделировалась на компьютере с 4-х ядерным процессором. Результаты эксперимента фиксировались не только в .log-файлах системы, но и с помощью вкладки «Быстродействие» в диспетчере задач Windows.

Заключение

В ходе проведения экспериментов по обработке большого числа одновременно поступающих запросов из различных потоков и обрабатываемых на разных уровнях иерархической структуры СУЗ можно добиться высокой производительности, гибкости и быстродействия вычислений. Указанные критерии обеспечиваются горизонтальным масштабированием вычислительной мощности, ограниченным только физическими возможностями подключаемых дополнительных узлов.

Список использованных источников

1. Косяков М.С. Введение в распределенные вычисления. – Санкт-Петербург: НИУ ИТМО, 2014. – 155 с.
2. Введение в распределенные системы [Электронный ресурс]. – URL: <http://www.intuit.ru/studies/courses/1115/177/lecture/4778> Режим доступа: свободный (Дата обращения 10.09.2015)
3. Kshemkalyani A. D., Singhal M. Distributed Computing: Principles, Algorithms, and Systems. – Cambridge University Press, 2008. – 754 p.
4. Параллельный вычисления (базовый курс) [Электронный ресурс]. – URL: <http://bigor.bmstu.ru/?cnt/?doc=Parallel/base.cou> Режим доступа: свободный (Дата обращения 16.09.2015)