

ДИСПЕТЧЕРИЗАЦИЯ НА ОСНОВЕ ДИНАМИЧЕСКОГО ПОДКЛЮЧЕНИЯ РЕСУРСОВ

Нагиев А.Е., Ботыгин И.А., Шерстнёва А.И.
Томский политехнический университет
andrew_nagiev09@mail.ru

Введение

В настоящее время наметилась четкая тенденция к проектированию распределенных гетерогенных систем управления. И это понятно – очень привлекательна идея увеличить вычислительную мощность систем за счет простой интеграции компьютерной техники в единый виртуальный кластер. Разработка распределенных вычислительных систем тесно связана с балансировкой нагрузки (БН), которая позволяет максимально эффективно использовать ресурсы виртуального кластера, а также повышает его отказоустойчивость и обеспечивает дополнительное подключение вычислителей при высоких нагрузках на основные узлы.

Метод динамического подключения ресурсов

В настоящей работе развивается диспетчеризация нагрузки на основе метода динамического подключения ресурсов [1-3].

Функциональная структура, на основе которой проводились программные эксперименты по исследованию предложенного алгоритма диспетчеризации (ДПР) представлена на рис. 1.

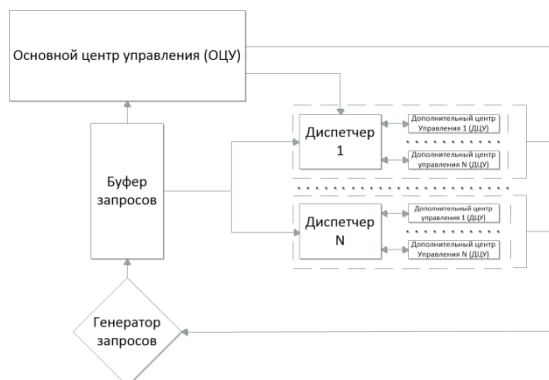


Рис. 1. Структурная схема распределенной вычислительной системы с ДПР

На схеме изображен основной центр управления (ОЦУ), который играет роль сервера управления. Также были добавлены специальные диспетчеры нагрузки. Они позволяют производить дополнительное подключение ресурсов (центров управления). Буфер запросов ОЦУ предназначен для накопления поступающих запросов на обработку от генератора запросов. После обработки запроса на вычисления происходит его удаление из буфера.

С увеличением количества поступающих запросов от генератора ОЦУ не успевает обрабатывать все запросы. Однако, в отличие от

классического серверного решения, где происходит отказ системы, в распределенной вычислительной системе с использованием ДПР происходит накопление поступающих запросов в буфере.

При достижении определенного количества запросов на обработку в буфере, происходит подключение диспетчеров нагрузки. Диспетчеры нагрузки в отдельных потоках начинают запускать дополнительные центры управления (ДЦУ). Количество подключенных ДЦУ ограничивается только скоростью поступления запросов и физическими возможностями системы, т.е. наличием дополнительных вычислительных мощностей.

Каждый запускаемый диспетчер и ДЦУ работают с буфером в отдельном потоке. После вычисления всех, переданных с помощью диспетчера нагрузки запросов, ДЦУ прекращает работу.

Программное моделирование

На рис. 2 изображена кривая, отображающая количество потоков, работающих в данный момент в генераторе запросов.

При определенной невысокой частоте поступающих запросов ОЦУ успевает их принимать, обрабатывать и отправлять результаты из различных потоков обратно в генератор запросов для дальнейшей работы. Однако, при повышении скорости поступающих запросов происходит увеличение количества обрабатываемых потоков и основной центр управления уже не успевает обработать все поступающие запросы, и они начинают накапливаться в буфере запросов. В этот момент происходит отказ классического серверного решения, т.к. ввиду отсутствия буфера потоки начинают конкурировать за ресурсы сервера. При достижении определенного количества потоков, задаваемого программно, включается диспетчер нагрузки, который подключает ДЦУ и увеличивает вычислительную мощность системы. Это приводит к тому, что рост количества запросов в буфере прекращается и происходит медленное снижение количества запросов путем распределения их между дополнительными центрами управления.

На рис. 3 можно заметить, разницу между загрузкой ЦП до начала отправки запросов от генератора и при обработке запросов. Увеличение частоты поступающих запросов увеличивает использование ЦП основным центром управления,

который все быстрее должен обрабатывать поступающие запросы.



Рис. 2. График количества действующих потоков в РАС

Затем рост нагрузки обеспечивается подключение ДЦУ. При этом происходит равномерное распределение вычислений между ядрами ЦП на основе датчика псевдослучайных чисел.

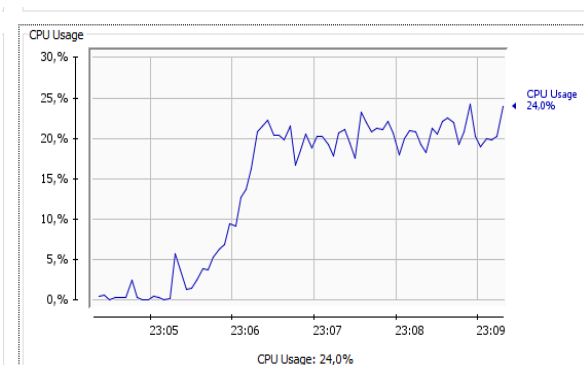


Рис. 3. Увеличение нагрузки системой ЦП

Далее, после окончания отправки запросов от генератора происходит обработка оставшихся запросов в буфере и выключение диспетчеров и ДЦУ. При этом происходит снижение использования центрального процессора до первоначального состояния (рис. 4, рис. 5).

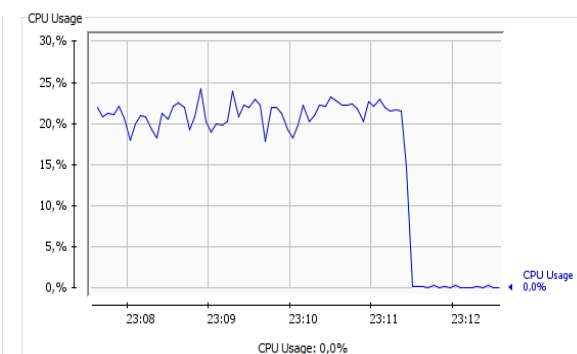


Рис. 4. Снижение использование ЦП при окончании поступления и обработки запросов

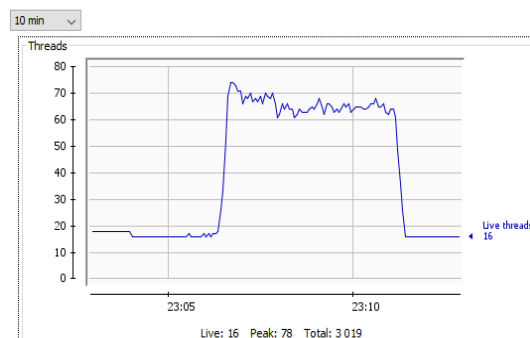


Рис. 5. Окончание отправки запросов от генератора характеризуется снижением потоков в системе

Результаты программного эксперимента

В результате проведения экспериментов можно говорить о том, что время подключения терминалов и обработки запросов к ОЦУ по сравнению с классическим вариантом, сократилось на 57%. За то же самое время к ОЦУ подключается и обрабатывается на 58,5% больше запросов от терминалов, чем при работе основного центра управления без подключения дополнительных.

При подключении второго ДЦУ скорость обработки выросла соответственно на 13% относительно системы с одним ДЦУ. При подключении третьего ДЦУ скорость обработки также возросла на 24% относительно системы, в которой работает два дополнительных центра управления.

Заключение

Эксперимент, описанный выше, показал эффективность разработанного метода динамического подключения ресурсов. Благодаря представленному модельному эксперименту можно сказать, что рассмотренный метод динамического подключения ресурсов будет эффективен также при значительном увеличении количества подключаемых узлов и увеличении масштаба архитектуры.

Список использованных источников

1. Botygin I.A., Nagiyev A.E. Model experiment for load balancing in a distributed computer system // 2015 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS). – Tomsk, Russia. 2015. – pp. 1-4.
2. Kshemkalyani A.D., Singhal M. Distributed Computing: Principles, Algorithms, and Systems. – Cambridge University Press, 2008. – 754 p.
3. Sherstnyov V.S., Sherstnyova A.I., Botygin I.A., Kustov D.A. Distributed Information System for Processing and Storage of Meteorological Data // High Technology: Research and Applications 2015. Key Engineering Materials. – Trans Tech

XIV Международная научно-практическая конференция студентов аспирантов и молодых учёных
«Молодёжь и современные информационные технологии»

Publications, Switzerland, 2016. – Vol. 685. – pp.
867-871.