

## ИМИТАЦИОННАЯ МОДЕЛЬ БАЛАНСИРОВКИ НАГРУЗКИ В РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Фролов С.Г., Попов В.Н., Ботыгин И.А.

Ботыгин И.А.

Томский политехнический университет

sgf2@tpu.ru

### Введение

Одна из ключевых проблем в распределенных вычислительных системах – это проблема балансировки нагрузки вычислительной сети, возникающая по нескольким достаточно очевидным причинам. Здесь и различная степень сложности, поступающих на обработку заданий, и неоднородность самого вычислительного кластера, и неоднородность межузловых взаимодействия в кластере и т.п. Основными задачами системы балансировки являются контроль за нагрузкой и состоянием вычислительных узлов, а также выбор, в соответствии с заданным алгоритмом, узла для передачи на него запроса пользователя. И именно балансировка нагрузки является на сегодняшний день одним из наиболее используемых приемов повышения производительности распределенных вычислительных систем за счет оптимального распределения заданий между узлами вычислительной системы. Поэтому разработка методов и алгоритмов оптимального диспетчирования вычислений в распределенных системах, динамично изменяющих в процессе работы свою инфраструктуру – важная задача.

На сегодняшний день существует множество методов балансировки нагрузки: Round Robin, Weighted Round Robin, Least connections, Weighted least connections и др. Балансировка нагрузки также разделяется на статическую, полудинамическую и динамическую. Если говорить о том, как взаимодействует балансировка нагрузки в вычислительной системе, то балансировка подразделяется на централизованную и распределенную.

В статье описывается метод балансировки вычислительной нагрузки в распределенных автоматизированных системах, ориентированных на мультиагентность и многопоточность обработки данных. Предложена схема управления обработки запросов от терминальных устройств, обеспечивающая эффективное динамическое горизонтальное масштабирование вычислительной мощности при пиковых нагрузках. Представлены GPSS-модель и результаты модельных экспериментов исследования разработанного алгоритма диспетчеризации нагрузки, показывающие его эффективность даже при значительном увеличении количества подключаемых узлов и увеличении масштаба архитектуры распределенной вычислительной системы.

### Архитектура распределенной вычислительной системы

Моделирование функционирования алгоритма балансировки нагрузки исследовалось на функциональной структуре распределенной вычислительной системы. Особенностью данной системы является возможность быстрого наращивания производительности с помощью горизонтального масштабирования вычислительных узлов и ресурсов. Архитектура системы базируется на компонентах, используемых при построении современных инструментальных средств распределенных вычислений (grid-систем). Но внутренняя структура комплекса, содержание функциональных компонентов (middleware) и протоколы их взаимодействия являются оригинальными. Архитектура разработанного комплекса виртуализирует три основных технических ресурса, на которых строится высокопроизводительный центр обработки данных (вычислительные системы, системы хранения данных и глобальные коммуникации), а затем собирает их в единый виртуальный компьютер, чтобы предоставлять его ресурсы в виде сервисов.

В структуре распределенной вычислительной системы обработки данных выделены вычислительные серверы, коммуникационный сервер и сервер базы данных. Планирование и диспетчирование процессами обработки данных возложено на специальный коммуникационный сервер. Взаимодействие всех пользователей с комплексом обработки осуществляется только через коммуникационный сервер. Основной задачей коммуникационного сервера является обеспечение оптимальной загрузки имеющихся в его распоряжении вычислительных серверов и обеспечение режима работы в реальном времени (online) пользователей. В такой ситуации пользователю уже не важно, на каком конкретном узле сети исполняется его задача – он просто потребляет определенное количество виртуальной процессорной мощности, имеющейся в сети. На вычислительных серверах комплекса производится математическая обработка данных[1].

Алгоритм использует информацию, которая хранится в профилях (метаописаниях) вычислительных серверов (таблица процессоров, таблица соответствия задач и таблица задач).

Основные шаги алгоритма:

Шаг 1. Формирование списка вычислительных серверов для решения задачи.

Шаг 2. Исключение из списка выключенных серверов.

Шаг 3. Исключение из списка занятых серверов.

Шаг 4. Исключение из списка серверов, с которыми связь по техническим причинам невозможна.

Шаг 5. Определение времени установления связи и скорость передачи данных с серверами.

Шаг 6. Определение сервера с максимальной свободной вычислительной мощностью.

Шаг 7. При наличии нескольких серверов с одинаковой вычислительной мощностью выбирается тот сервер, с которым скорость обмена данными выше.

Шаг 8. Увеличение на единицу значения поля Количество пользователей.

Шаг 9. При отсутствии серверов со свободными вычислительными мощностями заявка на выполнение задачи помещается в буфер коммуникационного сервера.

Шаг 10. Передача управления на выбранный вычислительный сервер.

Критерий выбора заключается в поиске такого сервера, нагрузка на который минимальна, а скорость обмена данными – максимальна.

#### **Gpss-модель для оценки эффективности алгоритма балансировки нагрузки**

Для оценки эффективности функционирования распределенной вычислительной системы, представленной выше, была разработана её имитационная модель на языке GPSS World [2].

Основными объектами моделируемой системы являются: потоки входных заданий, вычислительные сегменты, вычислительные серверы (ВС) сегментов, коммуникационный сервер (КС), серверы баз данных (БД), менеджер БД. Вычисление моментов появления заданий (случайной величины, связанной с промежутком времени между появлениями двух соседних заданий) осуществляется по нормальному закону распределения. Интервалы обслуживания также являются случайной величины, распределённой по экспоненциальному закону.

Изменяемыми параметрами модели являются: количество ВС, максимальное время обработки задания, количество типов заданий, средний интервал между заданиями, доля параллельных заданий, доля отклонённых заданий, среднее время обработки заданий КС, отклонение от среднего времени обработки заданий КС, вероятность сбоя ВС, вероятность восстановления ВС, среднее время восстановления ВС, отклонение от среднего времени восстановления ВС, количество КС, максимальное время обработки задания администратора, количество типов заданий, средний интервал между заданиями, среднее время восстановления ВС,

отклонение от среднего времени восстановления ВС. Типовыми заданиями в настоящей модели являлись задачи сбора, хранения и обработки метеорологических данных для анализа и прогнозирования климатических процессов [2].

В работе проведено исследование зависимостей таких критериев как: количество заданий общее, отклонённых, без параллелизма, с параллелизмом, аварийных ситуаций с восстановлением, без восстановления, с восстановлением (для заданий с параллелизмом), без восстановления (для заданий с параллелизмом), заданий в очереди, от следующих параметров: количество ВС, максимальное время обработки задания, количество типов заданий, интервал между заданиями, количество КС. Во всех экспериментах моделирование осуществлялось в течении 24 часов (модельное время).

При выбранных в модели параметрах быстродействия ВС: их количество (свыше 100), максимального времени обработки задания (до 100 единиц), количество КС (до 10) практически не влияет на исследуемые критерии. Количество типов выполняемых заданий значительно влияет на число выполненных в МВК заданий, а также на очередь к КС. Изменение интервала поступления заданий значительно влияет на исследуемые критерии, особенно на начальных этапах его увеличения. Эксперименты имитационного моделирования разработанной функциональной структуры распределенной вычислительной системы показали эффективность ее работы даже при значительном увеличении количества подключаемых узлов и увеличении масштаба архитектуры. Предложенные параметры модели и статические критерии оценки ее функционирования позволили комплексно и оценить качество разработанной модели.

#### **Заключение**

Вопрос о планировании нагрузки в распределенной вычислительной системе следует решать еще на ранней стадии развития любого сервиса. Первоначально проблемы недостаточной производительности распределенной вычислительной системы в связи ростом нагрузок можно решать путем наращивания мощности её серверов, но эффективнее – оптимизация используемых алгоритмов и программных кодов.

#### **Список использованных источников**

1. Ботыгин И.А., Попов В.Н. Архитектура распределенной файловой системы // Интернет-журнал «НАУКОВЕДЕНИЕ» 2014. № 6 <http://naukovedenie.ru/PDF/137TVN614.pdf> (доступ свободный). Загл. с экрана. Яз. рус. англ. DOI: 10.15862/137TVN614

2. I. A. Botygin, V. N. Popov, V. A. Tartakovsky, V. S. Sherstnev Architecture of scalability file system for meteorological observation data storing // Proc. of SPIE,

21st International Symposium Atmospheric and Ocean  
Optics: Atmospheric Physics. – 2015. – vol. 9680. – pp.  
96800J-1– 96800J-4. – doi: 10.1117/12.2205749.