

На правах рукопису

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія харчових технологій
Навчально-науковий інститут холоду,
кріотехнологій та екоенергетики
Факультет інформаційних технологій та кібербезпеки

**XVI Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів**

**“СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ”**

Матеріали конференції



Одеса
25–26 квітня 2016 р.

Стан, досягнення і перспективи інформаційних систем і технологій / Матеріали XVI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 25–26 квітня 2016 р. - Одеса, Видавництво ОНАХТ, 2016 р. - 176 с.

Збірник включає матеріали доповідей її учасників, які об'єднані по секціях кафедр: комп'ютерної інженерії (КІ), інформаційних технологій та кібербезпеки (ІТтаКБ).

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова – д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

Співголови :

Капрельянець Л.В. – д.т.н., проф., проректор з наукової роботи та міжнародних зв'язків,

Косой Б.В. – д.т.н., проф., в.о. директора ННІХКтаЕ ОНАХТ,

Котлик С.В. – к.т.н., доц., декан ФІТта КБ ОНАХТ,

Волков В.Е. – д.т.н., доц., директор ННІМАтаКС ОНАХТ,

Хобін В.А. – д.т.н., проф., завідувач кафедри автоматизації виробничих процесів ОНАХТ,

Невлюдов І.Ш. – д.т.н., проф., завідувач кафедри технології і автоматизації виробництва радіоелектронних і електронно-обчислювальних засобів ХНУРЕ,

Мельник А.О. – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,

Тарасенко В. П. – д.т.н., проф., завідувач кафедри СПіСКС НТУУ «Київський політехнічний інститут»,

Жуков І. А. – д.т.н., проф., директор інституту комп'ютерних технологій Національного авіаційного університету.

Члени оргкомітету:

Плотніков В. М. – д.т.н., проф., завідувач кафедри інформаційних технологій та кібербезпеки ОНАХТ.

Артеменко С.В. – д.т.н., проф., в.о. завідувача кафедри комп'ютерної інженерії ОНАХТ.

Князєва Н.О. – д.т.н., проф. кафедри комп'ютерної інженерії ОНАХТ.

Грищенко І.В. – к.т.н., заступник декана ФІТта КБ ОНАХТ.

Шамрай О.А. – к.т.н., доц. кафедри ТДтаВЕ ОНАХТ.

Матеріали подано українською, російською та англійською мовами.
Редактор збірника Шамрай О.А.

Щоб позбутися від мінуса, але не змінювати напрямки і знак, можна брати абсолютне значення.

Список літератури

1. Математика Unity3D. 2013 р. - <http://poqxert.ru/page/matematika-unity3d>

АНАЛИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ

Коломиец А. І.

Вопрос о планировании нагрузки следует решать ещё на ранней стадии развития любого веб-проекта. Первоначально проблемы недостаточной производительности сервера в связи ростом нагрузок можно решать путем наращивания мощности сервера, или же оптимизацией используемых алгоритмов, программных кодов и так далее. Но рано или поздно наступает момент, когда и эти меры оказываются недостаточными.

Приходится прибегать к кластеризации: несколько серверов объединяются в кластер; нагрузка между ними распределяется при помощи комплекса специальных методов, называемых балансировкой. Помимо решения проблемы высоких нагрузок кластеризация помогает также обеспечить резервирование серверов друг на друга [1].

Эффективность кластеризации напрямую зависит от того, как распределяется (балансируется) нагрузка между элементами кластера. Балансировка нагрузки может осуществляться при помощи как аппаратных, так и программных инструментов.

Очень желательно также, чтобы алгоритм балансировки обладал следующими свойствами [2]: **предсказуемость** (эффективность алгоритма для решения поставленных задач); **равномерная загрузка ресурсов системы**; **масштабируемость** (сохранение работоспособности при увеличении нагрузки).

Целью данной работы является анализ основных алгоритмов балансировки нагрузки с указанием их достоинств и недостатков.

Эффективность алгоритмов балансировки нагрузки определяется несколькими показателями, которые представлены ниже [1-2]:

Пропускная способность для оценки общего количества задач, которые успешно завершены.

Затраты указывают на стоимость процессов, участвующих в решении задачи, перераспределении процессов.

Отказоустойчивость измеряет способность алгоритма равномерно выполнять балансировку нагрузки в случае любого сбоя.

Время миграции - общее время перехода задачи от одного узла или ресурса к другому.

Время отклика. - интервал времени между отправкой запроса и получением ответа.

Использование ресурсов надлежащее использование всех ресурсов системы.

Масштабируемость - равномерная балансировка нагрузки в системе в соот-

ветствии с требованиями при увеличении числа узлов.

Производительность может быть определена как эффективность системы.

Балансировщики нагрузки используют различные алгоритмы для распределения нагрузки и/или максимального использования всех серверов в кластере. Благодаря алгоритмам достигается более высокая пропускная способность и улучшается время отклика в распределенных системах. Каждый из алгоритмов имеет как преимущества, так и недостатки [3].

Round Robin, или алгоритм кругового обслуживания, представляет собой перебор по круговому циклу: первый запрос передается одному серверу, затем следующий запрос передается другому и так до достижения последнего сервера, а затем всё начинается сначала. В этом алгоритме все задачи поровну делятся между всеми процессорами, однако различные задачи имеют различное время выполнения, то есть совершенно не учитывается загруженность того или иного узла в составе кластера.

Weighted Round Robin – это усовершенствованная версия алгоритма Round Robin. Суть усовершенствований заключается в следующем: каждому серверу присваивается весовой коэффициент в соответствии с его производительностью и мощностью. Это помогает распределять нагрузку более гибко: серверы с большим весом обрабатывают больше запросов.

Least Connections учитывает количество подключений, поддерживаемых серверами в текущий момент времени. Каждый следующий вопрос передается серверу с наименьшим количеством активных подключений. Однако не учитывается количество доступных ресурсов серверов.

В алгоритме **Destination Hash Scheduling** сервер, обрабатывающий запрос, выбирается из статической таблицы по IP-адресу получателя. Однако не учитывается количество доступных ресурсов серверов.

Алгоритм **Source Hash Scheduling** основывается на тех же самых принципах, что и предыдущий, только сервер, который будет обрабатывать запрос, выбирается из таблицы по IP-адресу отправителя.

Sticky Sessions это алгоритм распределения входящих запросов, при котором соединения передаются на один и тот же сервер группы. С помощью этого метода запросы распределяются по серверам на основе IP-адреса клиента. Метод гарантирует, что запросы одного и того же клиента будут передаваться на один и тот же сервер. Если закреплённый за конкретным адресом сервер недоступен, запрос будет перенаправлен на другой сервер.

Таким образом, в работе проведен анализ основных динамических алгоритмов балансировки нагрузки по основным показателям производительности, указаны достоинства и недостатки каждого алгоритма.

Список литературы

1. Thomas Erl, Robert Cope, Amin Naserpour. Cloud Computing Design Patterns. Prentice Hall. Ed.1st. – 2015. – P.592.
2. Rajwinder Kaur, Pawan Luthra. Load Balancing in Cloud Computing. DOI: 02.ИТС.2014.5.92 Association of Computer Electronics and Electrical Engineers. – 2014. – P.374-381.

3. Kalyani Ghuge, Minaxi Doorwar. A Survey of Various Load Balancing Techniques and Enhanced Load Balancing Approach in Cloud Computing. International Journal of Emerging Technology and Advanced Engineering. - Volume 4(10). 2014. – P.410-414.

АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ СИСТЕМ КОНТРОЛЮ ЗНАНЬ

Корчмар О. І., студент ОКР „бакалавр” факультету ІТ та КБ ОНАХТ

Керівник – ст. викл. каф. КІ Бондаренко В.Г.

У навчанні існують кілька способів використання інформаційно-комунікаційних технологій, широко застосовуються в таких процесах, як: отримання знань, закріплення отриманих знань, контроль знань. Контроль знань учня можна проводити за допомогою тесту або на прикладі практичних завдань. На сьогоднішній день у всіх навчальних закладах в якості технологічного засобу контролю знань широко застосовуються інформаційні технології. Для ведення таких процесів можна навести спеціальні допоміжні програми easyQuizzу, TestMaker, SunravTestOffice, iSpringQuizMaker [1]. Кожна вищевказана програма має свої можливості, способи застосування. Наприклад, якщо програмою easyQuizzу можна користуватися для проведення індивідуального тесту, то програми TestMaker, SunravTestOffice можна використовувати для поширення тестів через мережні технології і збору зберігання результатів тесту. Розглянемо iSpringQuizMaker [2, 3] як інтерактивний прийом тестування знань учнів. Програма iSpringQuizMaker є частиною програмного пакету iSpringSuite, відрізняється від інших програм проведення тесту різноманітністю видів тестів, пропонує проведення процесу тїста у вигляді анімації, дає можливість застосування підготовленого збірки тестів в різних технологіях, можливість підготовки програми підготовлених тестів на довільній мові. У програмі QuizMaker існують кілька видів проведення тестів і вони такі:

- швидка питання-відповідь у вигляді «так», «ні»;
- варіанти відповідей одного вибору;
- варіанти відповідей з кількома виборами;
- відкритий тест (викладач показує кілька правильних відповідей, і якщо тестований вводить текст, рівний якомусь варіанту відповіді, то він вважається правильним);
- установка відповідності (тут частини правого боку відповідно об'єднати з частинами лівого боку, відповідь вважається правильною);
- установка правильної послідовності (тут варіанти відповідей скласти так, як зазначено в питанні, відповідь вважається правильною);
- чисельний відкритий тест (тут відповідь питання полягає тільки з чисел і відповідь можна перевірити кількома умовами);
- заповнення тексту (тут потрібно доповнення пропущених частин мови);