

Генератор понять та тез

Введіть у це поле текст який потрібно розбити на поняття і тези.

Ефективність/Продуктивність - кількість ресурсів системи, що споживає програма (час процесора, розмір пам'яті, зовнішня пам'ять, ширина каналу мережі, і навіть взаємодія з користувачем). Чим менше ресурсів споживається, тим краще.

Надійність - ймовірність того, що результат роботи програми правильний. Це залежить від коректності алгоритмів та правильності кодування.

Стійкість це як програма розв'язує проблеми в нестандартних ситуаціях, як наприклад неправильні дані, недоступність необхідних ресурсів як наприклад пам'ять, локальна мережа, та неправильні дії користувача.

Зручність: ергономічність програми. Легкість, з якою особа може використовувати програму для своїх цілей.

Переносимість це діапазон апаратного забезпечення та операційних систем на яких можна компілювати чи інтерпретувати код програми, виконуючи її. Це залежить від

Для кращої генерації, у тексті кожне речення повинне закінчуватися на крапку. Кожне поняття повинне бути розділеним симаолом ' '.

Або введіть сюди власні поняття і тези.

ЗГЕНЕРУВАТИ

поняття1

теза1

Рисунок 1. Поля для вводу на сторінці заповнення бази знань

Висновок

Розроблено web-орієнтовану систему, яка забезпечує можливість автоматичного наповнення бази знань із навчального фрагменту, автоматичного генерування тестових завдань з навчального фрагменту та збереження цих тестових завдань. Система надає можливість учням проходити тести, після чого результати зберігаються у системі, що надає змогу переглянути ці результати в подальшому. Усе це забезпечує більш якісний, надійніший та оптимальніший спосіб контролю знань учнів

Список використаних джерел

1. Титенко, С. В. Генерація тестових завдань у системі дистанційного навчання на основі моделі формалізації дидактичного тексту / С. В. Титенко // Наукові вісті НТУУ "КПІ". – 2009. – № 1(63). – С. 47–57.
2. Gagarin A., Tyutenko S. Complex model of educational hypermedia environment for ongoing learning // Образование и виртуальность-2007: Сб. науч. тр. 11-й Междунар. конф. Укр. ассоциации дистанционного образования / Под общ. ред. В.А. Гребенюка, Др. Киншука и В.В. Семенца. — Харьков—Ялта: УАДО,2007. — С. 140—145.
3. Гагарін О.О., Титенко С.В. Проблеми створення гіпертекстового навчаючого середовища // Вісн. Східноукр. нац. ун-ту ім. Володимира Дала. — 2007. — Ч. 2, № 4(110). — С. 6—15.

УДК 004.94

ПОБУДОВА МОДЕЛЕЙ ВИСОКОШВИДКІСНИХ ПОТОКІВ ДЛЯ ХМАРКОВИХ СХОВИЩ

Струбицький Р.П.

Національний університет "Львівська політехніка", аспірант

Нові послуги реального часу для реалізації хмаркових обчислень з'явилися на ринку послуг мережі Інтернет порівняно недавно, але ці послуги поки-що не в достатній мірі підтримуються в мережі "адекватними" засобами, такими як, наприклад, нові високошвидкісні протоколи, що орієнтовані на ці послуги і алгоритми управління високозавантаженим трафіком. Саме тому якість, з якою надаються послуги хмаркових обчислень, в основному напряму залежать від кількості вільних ресурсів мережі.

Очевидно, що забезпечення якості обслуговування потребує запровадження нових механізмів і протоколів як в мережі, так і в обладнанні користувачів. Транзитні вузли (маршрутизатори) є власністю провайдерів і є достатньо інертними з точки зору впровадження нових механізмів і протоколів. Саме тому простіше і дешевше змінювати процеси передачі такого трафіку при проектуванні нової мережі або внесення кардинальних змін у вже існуючу.

Протоколи транспортного рівня UDP і TCP надають розробнику додатків свободу вибору транспортної послуги з точки зору доставки інформації: UDP є простим протоколом, але не дає гарантій на якість обслуговування, а протокол TCP, у свою чергу, забезпечує гарантовану доставку даних, але дуже часто з великою затримкою.

При збільшенні пропускної здатності мережі і затримок TCP, що дуже характерне для міжконтинентальних мереж, протокол стає неефективним [2]. Ці проблеми виникають через повільне відновлення втрачених пакетів та збільшення RTT, що властиве алгоритмам управління протоколом. Додатки, які транспортують дані між хмарковими сховищами, повинні інтенсивно використовувати більш високу пропускну здатність мереж. Тобто для них потрібні нові транспортні протоколи.

Для більшості послуг сфери хмаркових обчислень, які реалізовані “з кінця в кінець” на прикладному рівні і, як правило, такі, що використовують на транспортному рівні протокол UDP, достатньо важливими параметрами є затримка пакету “з кінця в кінець” і дисперсія цієї затримки. В першу чергу, це пояснюється тим, що послуги “потокowego” типу орієнтовані на передачу інформації за запитом, в результаті чого кінцевий користувач і є системою, яка оцінює якість наданої послуги.

Протокол передачі даних UDP був визначений IETF в 1980 році в документі [1], він має ряд переваг, які роблять його доволі привабливим, в першу чергу для додатків, які не мають жорстких вимог за такими параметрами якості обслуговування, як ймовірність втрати пакету, затримка і дисперсія затримки. Серед очевидних переваг UDP необхідно виділити наступні: відсутність фази встановлення з'єднання, відсутність стану з'єднання, малі поточні витрати.

Для подолання недоліків протоколу UDP почали розроблятися протоколи прикладного рівня, які будуються над UDP. Один з них - UDP-базований протокол передачі даних (UDT). Цей протокол має в собі механізм управління перевантаженням, який підтримує ефективність та стабільність. Його розгортання є простим і відносно дешевим, без будь-яких змін в мережевій інфраструктурі чи операційній системі.

Схожі роботи з модифікацій протоколів для швидкісних мереж проводились для удосконалення TCP - Fast TCP, HighSpeed TCP і Scalable TCP [3].

Вивчення перевантажень, які виникають в комп'ютерних мережах, і методів їх запобігання є дуже актуальним питанням на сучасному етапі розвитку хмаркових обчислень. Даний підхід до обчислень передбачає передачу достатньо великої кількості даних через мережу, що й є передумовою виникнення перевантажень в комп'ютерних мережах.

Особливо актуальною стає дана задача при переході до багаточкової дистрибуції і розміщенні серверів даних на різних континентах (рис. 1). При такому підході доводиться розглядати комбінації протоколів передачі даних – окремо для сегменту роботи з абонентами і для передачі даних в середині хмаркового сховища між різними його серверами. Формально така модель передбачає наявність сателітів основних серверів у місцях найбільшої мережевої активності абонентів.

Використання такої моделі доставки даних знижує кількість хопів, що істотно збільшує швидкість скачування контенту з мережі Інтернет. Хоп (hop, стрибок) - назва процесу передачі мережевого пакету (або датаграми) між хостами (вузлами) мережі. Зазвичай використовується для визначення “відстані” між вузлами (чим більше хопів - тим складніший шлях маршрутизації і тим “далі” знаходяться вузли один від одного).

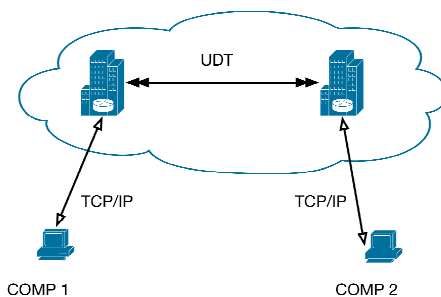


Рисунок 1 – Модель доступу до міжконтинентального сховища даних.

Тобто при зниженні кількості хопів, кінцеві користувачі відчувають меншу затримку при завантаженні контенту, відсутність різких змін швидкості завантаження та високу якість потоку даних. Така стабільність дозволяє операторам дата-центру доставляти відеоконтент у форматі HD, забезпечувати швидке завантаження файлів великих розмірів або організувати відеотрансляцію з високою якістю сервісу (QoS) і низькими витратами на мережу.

Модель багаточкової дистрибуції здатна запобігти затримці при передачі даних, можливим переривання зв'язку і втратам на перевантажених каналах і стиках між ними. Управління навантаженням

при передачі мережевого трафіку дозволяє розвантажити магістраль і вузли мережі, розподілити навантаження між віддаленими серверами.

Розміщення серверів в безпосередній близькості від кінцевих користувачів може збільшити вихідну пропускну здатність всієї системи. Наприклад, наявність єдиного порту 100 Мбіт/с не означає дану швидкість на всіх ділянках мережі, так як вільна пропускна здатність магістрального каналу в момент передачі може бути всього 10 Мбіт/с. У випадку, коли використовуються 10 розподілених серверів, сумарна пропускна здатність може скласти 10×100 Мбіт/с.

При такому підході хмаркове сховище складатиметься з географічно розподілених багатофункціональних платформ, взаємодія яких дозволяє максимально ефективно обробляти і задовольняти запити користувачів для отримання контенту.

Проте, тут можливі дві моделі реалізації. У першому підході дані центрального сервера реплікуються на периферійні платформи. Кожна платформа підтримує в актуальному стані повну або часткову копію розповсюджуваних даних. В іншому випадку дані кешуються на сателітах і зберігаються там декілька днів.

Великі хмаркові сховища можуть складатися з величезної кількості розподілених вузлів і розміщувати свої сервери безпосередньо у мережі кожного локального інтернет-провайдера. Багато операторів роблять акцент на пропускній спроможності сполучних каналів і мінімальній кількості точок приєднання в регіоні присутності. Незалежно від використовуваної архітектури, головним призначенням подібних мереж є прискорення передачі як статичного контенту, так і безперервного потоку даних.

Для вивчення поведінки протоколів транспортного рівня широко використовується дискретно-часовий симулятор з відкритим вихідним кодом NS-3 (Network Simulator 3) [4]. Для дослідника він надає набір класів, які дозволяють моделювати протоколи і процеси, які відбуваються в комп'ютерних мережах. Також симулятор дозволяє моделювати процеси в реальному часі та інтегрувати його з дослідним стендом. Симулятор NS-3 має цілу множину готових тестів для всіх компонентів, що гарантує достовірність отриманих результатів.

Однак, нейромережевий симулятор містить тільки класи для базових протоколів передачі (TCP, UDP). При потребі досліднику потрібно на основі базових створювати власні класи протоколів, але відкритість архітектури симулятора дозволяє це зробити.

У процесі роботи над дослідженням характеристик протоколу UDT було розроблено класи для моделювання протоколу в середовищі NS-3 та проведено ряд модельних експериментів для дослідження характеристик протоколу при роботі з великими обсягами даних.

Список використаних джерел

1. User Datagram Protocol - <https://tools.ietf.org/html/rfc768>
2. W. Feng, P. Tinnakornsrisuphap, "The Failure of TCP in High-Performance Computational Grids". Supercomputing 2000.
3. E. He, R. Kettimut, S. Hegde, Y. Gu, M. Welzl, and W. E. Allcock, Survey of Transport Protocols Other Than Standard TCP, GGF White paper Draft, 2003.
4. The NS-3 Manual, The NS-3 Project, 2010. <http://www.nsnam.org/docs/release/3.10/manual/singlehtml/index.html>

УДК 004.045

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПОБУДОВИ ЗВІТІВ ДЛЯ СИСТЕМИ «HOSTEL MANAGEMENT SYSTEM»

Шпінгаль М.Я.¹⁾, Микитюк В.П.²⁾

Тернопільський національний економічний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Постановка проблеми

Для веб-орієнтованих інформаційних систем однією із функціональних особливостей є генерація різного роду звітів на основі інформації, яка зберігається в базі даних. При реалізації інтелектуальної підсистеми генерації звітної документації, як правило виникає проблема раціонального вибору