МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп’ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Ваврейчук Сергій Володимирович

Система реплікації Master-Slave серверів MySQL на основі платформи Percona / MySQL Master-Slave servers replication system based on Percona

напрям підготовки: 6.050102 - Комп’ютерна інженерія
фахове спрямування - Комп’ютерні системи та мережі
Бакалаврська робота

Виконав студент групи КСМ 41/1
Сергій Володимирович Ваврейчук

Науковий керівник:
Возняк С. І.

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 48 сторінок пояснюючої записки, 3 рисунки, 7 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломного проекту є розроблення системи взаємодії Master-Slave серверів MySQL на основі платформи Percona.

Проведено аналітичний огляд сучасних SQL СУБД: MySQL, PostgreSQL, Oracle, Microsoft SQL , DB2, Centaur, Ingres, Teradata Database. Перевагами MySQL є: пропонується багато функцій, навіть у безкоштовній версії; пакет MySQL включений в стандартні репозиторії ОС Linux, що дозволяє встановлювати її елементарно.

Проведено огляд основ реплікації Master-Slave серверів.

Здійснено перевід MySQL серверів на систему Percona MySQL.

Здійснено запуск серверів баз даних у виробничому режимі.

Здійснено конфігурацію і налаштування Percona MySQL серверів. Введено в експлуатацію додаткові серверні процедури Master-Slave.

Проведено моніторинг та діагностику серверів Percona MySQL.

Ключові слова: РЕПЛІКАЦІЯ, MYSQL, PERCONA, СЕРВЕР БАЗ ДАНИХ.

RESUME

The diploma project contains 48 pages of explanatory note, 3 figures, 7 tables, 2 appendices. Volume of graphic material 2 sheets of A3 format.

The aim of the diploma project is to develop a system of interaction of Master-Slave MySQL servers based on the Percona platform.

An analytical review of modern SQL databases: MySQL, PostgreSQL, Oracle, Microsoft SQL, DB2, Centaur, Ingres, Teradata Database. The advantages of MySQL are: it offers many features, even in the free version; MySQL package is included in standard Linux repositories, which allows you to install it easily.

The basics of replication of Master-Slave servers are reviewed.

MySQL servers were transferred to the Percona MySQL system.

Database servers started in production mode.

Percona MySQL servers are configured and configured. Additional server procedures Master-Slave were put into operation.

Percona MySQL servers were monitored and diagnosed.

Keywords: REPLICATION, MYSQL, PERCONA, DATABASE SERVER.

ЗМІСТ

Вступ 3
1 СУБД на основі MySQL 4
1.1 Сучасні SQL системи управління базами даних 4
1.2 Основи реплікації Master- Slave серверів баз даних 15
1.3 Постановка задачі 18
2 Проектування системи master-slave на основі Percona MySQL 19
2.1 СУБД MySQL для логістичної фірми 19
2.2 Перевід MySQL Серверів на систему Percona MySQL Master-Slave 20
2.3 Оптимізація взаємодії транзакцій між серверами баз даних 23
3 Запуск серверів баз даних у виробничому режимі 29
3.1 Конфігурація і налаштування Percona MySQL серверів 29
3.2 Ввід в експлуатацію додаткових серверних процедур Master-Slave 35
3.3 Моніторинг та діагностика серверів Percona MySQL 38
Висновки 41
Список використаних джерел 42

					ДП.КСМ. 07100/14.00.00.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ваврейчук С.В			СИСТЕМА РЕПЛІКАЦІЇ MASTER-SLAVE СЕРВЕРІВ MYSQL НА ОСНОВІ ПЛАТФОРМИ PERCONA	Літ.	Арк. 8 / Акрушів 48
Перевір.		Возняк С.В.					
Консультан		Паздрій І.Р.				ТНЕУ. ФКІТ. КСМ – 41/1	
Н. Контр.		Гураль І.В.					
Затверд.		Березький					

ВСТУП

База даних являє собою певний набір даних, які, як правило, пов'язані об'єднуючим ознакою або властивістю (або декількома). Ці дані впорядковані, наприклад, за алфавітом. Велика кількість різних даних, які можуть бути поміщені в єдину базу, веде до безлічі варіацій того, що може бути записано: особисті дані користувачів, записи, дати, замовлення і так далі. Наприклад, якщо у вас інтернет-магазин, то база даних вашого сайту може містити прайс-листи, каталог товарів або послуг, звіти, статистику та інформацію про клієнтів.

В першу чергу це зручно тим, що інформацію можна швидко заносити в базу даних і так само швидко її витягати при необхідності. Якщо на зорі розвитку web-розробки всі необхідні дані потрібно було прописувати в коді сторінки, то тепер така необхідність відсутня - потрібна інформація може бути запрошена з бази даних за допомогою скриптів. Спеціальні алгоритми зберігання і пошуку інформації, які використовуються в базах даних, дозволяють знаходити потрібні відомості буквально за частки секунд - а при роботі в віртуальному просторі швидкість роботи ресурсу важлива як ніщо інше.

Рано чи пізно перед будь-яким адміністратором виникає проблема, коли потужності його серверів не вистачає для виконання поставлених завдань, а придбання нової техніки практично неможливо з причини фінансових труднощів, або просто така покупка викликає сумнів з приводу її окупності в майбутньому. У загальних рисах реплікація бази даних - це об'єднання потужностей набору комп'ютерів для підвищення коефіцієнта продуктивності системи та її відмовостійкості.

Метою дипломного проекту є розроблення системи взаємодії Master-Slave серверів MySQL на основі платформи Percona.

1 СУБД НА ОСНОВІ MYSQL

1.1 Сучасні SQL системи управління базами даних

База даних являє собою певний набір даних, які, як правило, пов'язані об'єднуючим ознакою або властивістю (або декількома). Ці дані впорядковані, наприклад, за алфавітом. Велика кількість різних даних, які можуть бути поміщені в єдину базу, веде до безлічі варіацій того, що може бути записано: особисті дані користувачів, записи, дати, замовлення і так далі. Наприклад, якщо у вас інтернет-магазин, то база даних вашого сайту може містити прайс-листи, каталог товарів або послуг, звіти, статистику та інформацію про клієнтів.

В першу чергу це зручно тим, що інформацію можна швидко заносити в базу даних і так само швидко її витягати при необхідності. Якщо на зорі розвитку web-розробки всі необхідні дані потрібно було прописувати в коді сторінки, то тепер така необхідність відсутня - потрібна інформація може бути запрошена з бази даних за допомогою скриптів. Спеціальні алгоритми зберігання і пошуку інформації, які використовуються в базах даних, дозволяють знаходити потрібні відомості буквально за частки секунд - а при роботі в віртуальному просторі швидкість роботи ресурсу важлива як ніщо інше.

Важливою є і взаємозв'язок інформації в базі даних: зміна одного рядка може привести до значних змін інших рядків. Працювати з даними таким чином набагато простіше і швидше, ніж якби зміни стосувалися лише одного місця в базі даних.

Для забезпечення доступу до інформації, що зберігається в базі даних, а також для управління нею, застосовують систему управління базами даних (СУБД). СУБД - це комплекс мовних і програмних засобів, призначений для створення, ведення і сумісного використання БД багатьма користувачами.

Зазвичай СУБД розрізняють по використовуваної моделі даних. Так, СУБД, що базуються на використанні реляційної моделі даних, називають реляційними СУБД. Системи управління базами даних допомагають впорядкувати інформацію, а також зв'язати бази даних між собою, при цьому надавши звіт про зміни і зареєстрованих події. СУБД відповідає за копіювання та відновлення даних після будь-яких збоїв.

Що стосується класифікації баз даних, то тут можливі різні варіанти. Якщо розділити бази по моделі даних: ієрархічні (мають деревоподібну структуру), мережеві (за своєю структурою схожі на ієрархічні), реляційні (використовуються для керування базами даних), об'єктно-орієнтовані (використовуються для об'єктної моделі даних) і об'єктно реляційні (злиття реляційного і об'єктно-орієнтованого виду баз даних).Або, якщо розподіл йде по тому, де розміщується СУБД, їх можна розділити на локальні - вся СУБД розміщується на одному комп'ютері, і розподілені - частини системи управління базами даних знаходяться на декількох комп'ютерах.

Інформація, яка зберігається в базах даних, не обмежується тільки текстовими або графічними файлами - сучасні версії СУБД підтримують також формати аудіо і відеофайлів.

Крім основної функції БД - зберігання і систематизації величезної кількості інформації - вони дозволяють швидко обробляти клієнтські запити і видавати свіжу й актуальну інформацію.

Це стосується і змін, які вносить користувач - замість того, щоб міняти інформацію в кожному файлі сайту, можна поміняти її в базі даних, і тоді на кожній сторінці відразу ж буде відображена коректна інформація.

Схема роботи з базою даних зображе на рисунку 2.1.

Рисунок 1.1 – Робота з базою даних

Якщо мова йде про вибір СУБД для підприємства, то слід взяти до уваги можливість СУБД «рости» разом з розвитком організації. Малому бізнесу можуть знадобитися тільки базові функції і можливості, а також невелика кількість інформації, що розміщується в БД. Але вимоги можуть істотно зростати з плином часу, а перехід на іншу СУБД може стати проблемою.

Існує кілька популярних СУБД, як платних, так і безкоштовних, які можна рекомендувати для застосування в організації.

MySQL є однією з найпопулярніших і поширених СУБД, яка використовується в багатьох компаніях (наприклад, Facebook, Wikipedia, Twitter, LinkedIn, Alibaba і інших). MySQL є реляційною СУБД, яка відноситься до вільного програмного забезпечення: вона поширюється на умовах GNU Public License. Як правило, цю систему управління базами даних визначають як хорошу, швидку і гнучку систему, рекомендовану до застосування на невеликих або середніх проектах. У MySQL є безліч різних переваг. Наприклад, вона підтримує різні типи таблиць: як відомі MyISAM і InnoDB, так і менш відомі HEAP і MERGE; крім того, кількість підтримуваних типів стає дедалі більше. MySQL виконує всі команди швидко - можливо, зараз це найшвидша СУБД з усіх існуючих. З цією системою управління базами даних може одночасно працювати необмежена кількість користувачів, а число рядків в таблицях може дорівнювати 50 мільйонам.

Так як в порівнянні з деякими іншими СУБД MySQL підтримує меншу кількість можливостей, то і працювати з нею значно простіше, ніж, наприклад, з PostgreSQL.

Для роботи з MySQL використовується не тільки текстовий, а й графічний режим. Це можливо завдяки додатку phpMyAdmin: для роботи в додатку навіть не потрібно буде знати SQL-команди, а адмініструвати свою базу даних можна прямо через браузер.

В цілому можна відзначити, що MySQL - це вибір тих, кому необхідна СУБД для проекту невеликого або середнього розміру, швидка і зручна в роботі і без складнощів з адмініструванням.

Основними перевагами MySQL є:

- розповсюджується безкоштовно;
- зрозуміла документація;
- пропонується багато функцій, навіть у безкоштовній версії;
- пакет MySQL включений в стандартні репозиторії найбільш поширених дистрибутивів операційної системи Linux, що дозволяє встановлювати її елементарно;
- може працювати з іншими базами даних, включаючи DB2 і Oracle.

Також є і недоліки:

- відсутня вбудована підтримка XML або OLAP;
- для безкоштовної версії доступна тільки платна підтримка.

Ідеально підходить для: організацій, яким потрібен надійний інструмент управління базами даних, але безкоштовний.

PostgreSQL – це вільно поширювана система управління базами даних відноситься до об'єктно-реляційному типу СУБД. Як і у випадку з MySQL, робота з PostgreSQL грунтується на мові SQL, однак, на відміну від MySQL, PostgreSQL підтримує стандарт SQL-2011. Ця СУБД не має обмежень ні щодо максимального розміру бази даних, ні по максимуму записів або індексів в таблиці.

Якщо говорити про переваги PostgreSQL, то, безумовно, це надійність транзакцій і реплікації, можливість наслідування і легка розширюваність. PostgreSQL підтримує різні розширення і варіанти мов програмування, такі як PL / Perl, PL / Python і PL / Java. Також є можливість завантажувати C-сумісні модулі.

Багато хто відзначає, що на відміну від MySQL дана СУБД має хорошу і детальну документацію, яка дає відповіді практично на всі питання.Про те, що це більш масштабна, ніж MySQL, СУБД, говорить і той факт, що PostgreSQL періодично порівнюють з такою потужною системою управління даних, як Oracle.

Все це дозволяє говорити про PostgreSQL як про одну з найбільш просунутих СУБД на даний момент.

Oracle відноситься до об'єктно-реляційному типу. Назва походить від назви розробила цю систему фірми Oracle. Нарівні з SQL СУБД використовує процедурне розширення під назвою PL / SQL, а також мову Java.

Oracle - це система, що відрізняється стабільністю вже не один десяток років, тому її вибирають великі корпорації, для яких важлива надійність відновлення після збоїв, налагоджена процедура бекапа, можливість масштабування і інші цінні можливості. До того ж ця СУБД забезпечує відмінну безпеку і ефектну захист даних.

Актуальна версія Oracle призначена для хмарних середовищ і може бути розміщена на одному або декількох серверах, це дозволяє управляти базами даних, які містять мільярди записів. Деякі з функцій новітньої версії Oracle включають в себе grid framework і використання як фізичних, так і логічних структур.

Це означає, що фізичне управління даними не впливає на доступ до логічних структур. Крім того, безпеку в цій версії доведена до найвищого рівня, тому що кожна транзакція ізольована від інших.

До переваг Oracle можна віднести:

– найсвіжіші інновації та вражаючий функціонал вже впроваджені в цьому продукті, оскільки компанія Oracle прагне тримати планку навіть на тлі інших розробників СУБД;

– СУБД від Оракул є вкрай надійною, фактично це еталон надійності серед подібних систем.

Крім переваг,також є і декілька недоліків:

– вартість Oracle може виявитися непомірно високою, особливо для невеликих організацій.

– система може потребувати значних ресурсів вже відразу після установки, тому можливо буде потрібно модернізувати обладнання для впровадження Oracle.

Ідеально підходить для великих організацій, які працюють з величезними базами даних і різноманітними функціями.

Ще однією з популярних СУБД є програмний продукт Microsoft SQL-сервер. Це система управління базами даних, движок якої працює на хмарних серверах, а також локальних серверах, причому можна комбінувати типи застосовуваних серверів одночасно. Незабаром після випуску Microsoft SQL сервер 2016, Microsoft адаптувала продукт для операційної системи Linux, а на Windows-платформі він працював спочатку.

Однією з унікальних особливостей Microsoft SQL є temporal data support (тимчасова підтримка даних), яка дозволяє відстежувати зміни даних з плином часу. Остання версія Microsoft SQL-сервер підтримує dynamic data masking (динамічну маскування даних), яка гарантує, що тільки авторизовані користувачі будуть бачити конфіденційні дані.

Перевагами Microsoft SQL :

– продукт дуже простий у використанні;

– поточна версія працює швидко і стабільно;

– платформа надає можливість регулювати і відслідковувати рівні

продуктивності, які допомагають знизити використання ресурсів;

- можна отримати доступ до візуалізації на мобільних пристроях;
- дуже добре взаємодіє з іншими продуктами Microsoft;

Недоліками Microsoft SQL є:

- ціна для юридичних осіб виявляється неприйнятною для більшої частини організацій;
- навіть при ретельній налаштування продуктивності корпорація SQL Server здатний зайняти всі доступні ресурси;
- повідомляється про проблеми з використанням служби інтеграції для імпорту файлів.

Ідеально підходить для: великих організацій, які вже використовують ряд продуктів Microsoft.

Створена компанією IBM, DB2 є СУБД, яка має можливості NoSQL, і може читати JSON і XML-файли. З огляду на те, що система розроблялася для серверів компанії IBM модельного ряду iSeries, логічно, що система працює на Windows, Linux і Unix.

Діалект мови SQL, що використовується в DB2 за рідкісними винятками строго декларативний, система забезпечена багатофазовими оптимізаторами, які будують за цими декларативним конструкціям план виконання запиту. У діалекті SQL DB2 відсутні підказки оптимізатора, мало розвинена (а довгий час взагалі була відсутня) мова збережених процедур, і, таким чином, все направлено на підтримку декларативного стилю написання запитів. Мова SQL DB2 при цьому є обчислювально повною, тобто потенційно дозволяє в декларативній формі визначати будь-які обчислювані відповідності між вихідними даними і результатом. Це досягається в тому числі за рахунок використання табличних виразів, рекурсії та інших розвинених механізмів маніпулювання даними.

Оптимізатор DB2 широко використовує статистику розподілу даних в таблицях (якщо процес її збору був виконаний адміністратором бази даних),

тому один і той же запит на мові SQL може бути направлений в абсолютно різні плани виконання в залежності від статистичних характеристик даних, які він обробляє.

В рамках концепції підвищення рівня інтеграції засобів безпеки в комп'ютерній системі, DB2 не має власних коштів аутентифікації користувачів, інтегруючись із засобами операційної системи або спеціалізованими серверами безпеки. В рамках DB2 здійснюється тільки авторизація користувачів, аутентифікованих системою.

Сучасні версії DB2 забезпечують розширену підтримку використання даних в форматі XML, в тому числі операції з окремими елементами документів XML.

Поточна версія DB2 пропонує різноманітні поліпшення і доробки. Одне з них, прискорення Blu, яке призначене, для того щоб зробити цю базу даних швидше. Пропуск даних призначений для підвищення швидкодії системи з великою кількістю даних, ніж може вона може вмістити в себе. Остання версія DB2 також забезпечує вдосконалені функції аварійного відновлення, сумісності та аналітики.

Переваги DB2:

– Blu Acceleration дозволяє грамотно задіяти ресурси для об'ємних баз даних;

– може бути розміщена в хмарному сховищі, на фізичному сервері, або ж і там, і там одночасно;

– кілька завдань можуть виконуватися одночасно за допомогою планувальника завдань;

– коди помилок і коди завершення дозволяють легко відстежити, які завдання виконуються або виконалися за допомогою планувальника завдань.

Недоліки DB2:

– ціна за межами бюджету багатьох фізичних осіб і невеликих організацій;

– сторонні додатки або додаткове програмне забезпечення потрібно, для того щоб змусити функціонувати кластери або кілька вторинних вузлів;

– базова підтримка доступна тільки протягом трьох років.

Підходить для: великих організацій, які планують вичавлювати максимум з наявних ресурсів і обробляють великі БД.

Останнім часом намітився перехід від реляційних СУБД до об'єктно-орієнтованим (що явно простежується на прикладі Oracle). Informix також дотримуючись цієї концепції анонсувала нове рішення СУБД Centaur що базується на реляційній БД Informix Dynamic Server 7.3 і об'єктно-реляційної БД Informix Universal Data Option і поєднує в собі високу швидкодію Dynamic Server при роботі з даними з універсальністю і мультимедіа функціями Universal Data Option. Дана реалізація призначена для розробки інтернет систем. Реалізовані в новій системі засоби Java дозволять розробникам створювати на цій мові збережені процедури, призначені для користувача програми і компоненти DataBlades, які в Informix називають замовними розширеннями бази даних.

З точки зору клієнтів Informix, це стане великим кроком вперед, оскільки до теперішнього часу при роботі з DataBlades вони могли користуватися тільки мовою Сі і SPL, внутрішнім мовою фірми Informix для написання збережених процедур. Крім того, пакет Centaur буде оснащений вбудованими засобами обробки об'єктів ActiveX. Це дасть можливість, наприклад, створювати збережені процедури БД на мові Visual Basic; правда, для цього потрібно, щоб пакет Centaur виконувався в середовищі Windows NT.

Centaur буде являти собою надбудову Informix Dynamic Server і працювати з традиційним для цього пакета форматом БД, так що в розпорядженні користувачів залишаться всі колишні функції, а модернізація системи до рівня нової версії не буде пов'язана з великими труднощами. Крім того, в пакеті Centaur будуть збережені всі можливості конструювання і

програмування, завдяки яким система Informix Universal Server визнана видатним технічним досягненням. Нова система буде оснащена засобами об'єктно-орієнтованого конструювання баз даних, створення спеціалізованих таблиць і програм індексування; що дозволить користувачам вбудовувати в запити власні функції і не покладатися виключно на стандартні засоби SQL.

СУБД Ingres - інтелектуальна розподілена реляційна система управління БД з відкритим кодом, об'єктно-орієнтоване середовище розробки додатків в архітектурі клієнт / сервер компанії Computer Associates International, Inc. (CA).

Можливості СУБД включають:

– кластери високої доступності для забезпечення безперебійної продуктивності в разі відмови одного вузла сервера або СУБД в рамках кластерної конфігурації;

– масштабовані кластери СУБД, що дозволяють замовникам використовувати Linux і комерційне апаратне забезпечення для досягнення цілодобової і щоденної доступності;

– секціонування таблиць та механізм індексації, спеціально розроблений для потреб великих БД;

– обробка паралельних запитів, що забезпечує масштабованість шляхом ділення індивідуальних запитів на компоненти, які можуть бути виконані в паралельному режимі на всіх доступних обчислювальних ресурсах;

– оперативна реорганізація таблиць і індексів, що дозволяє забезпечити цілодобову доступність, допускаючи проведення обслуговування таблиць СУБД при збереженні БД в стані готовності до експлуатації;

– можливість використання повної функціональності 64-бітних ОС, включаючи поліпшене виконання запитів, адресацію, підтримку 64-бітних файлових систем і 64-бітного кешу БД;

– підтримка глобальних джерел бізнес-даних в кодуванні Unicode, універсального набору символів фіксованої довжини.

СУБД Ingres може інтегруватися з іншими програмами та даними в гетерогенному середовищі. Ingres добре підходить для вбудованих додатків, що досягається завдяки використанню промислових стандартів підключення до джерел даних, тому розробники можуть працювати як з платформою J2EE, так і з .NET, або використовувати обидві платформи одночасно. СУБД Ingres доступна у версіях ОС Windows, UNIX і OpenVMS.

СУБД компанії Teradata Database призначена для створення сховищ даних, доступна для ОС Windows 2000 Server, Windows 2000 Advanced Server, Windows .NET Server і ряду версій UNIX.

Відмінністю цієї СУБД від інших продуктів є спеціальний механізм паралельної обробки запитів, який використовує розбиття завдання на ряд окремих підзадач і їх паралельне виконання. При цьому навіть такі операції, як сортування або угруповання даних, також можуть розділятися на кілька завдань і виконуватися паралельно - це досягається за рахунок спеціального способу зберігання даних, при якому частини БД також діляться між підзадачами. Самі підзадачі можуть розподілятися між наявними процесорами, яких може бути 1, 2 або більше. Завдяки механізму паралельного вирішення завдань Teradata Database відрізняється високою масштабованість.

Для створення клієнтських додатків, що використовують Teradata Database, існують ODBC-драйвер, документований клієнтський API, сумісний з IBM DB2 (це означає, що клієнтські програми, які використовують DB2, можуть виконуватися і с Teradata), і ряд інших інтерфейсів. Засоби адміністрування цієї СУБД включають засоби резервного копіювання і відновлення, перегляду і редагування даних, моніторингу продуктивності і навантаження, а також утиліту

адміністрування для управління всіма перерахованими завданнями, доступну для ОС Windows.

1.2 Основи реплікації Master- Slave серверів баз даних

Рано чи пізно перед будь-яким адміністратором виникає проблема, коли потужності його серверів не вистачає для виконання поставлених завдань, а придбання нової техніки практично неможливо з причини фінансових труднощів, або просто така покупка викликає сумнів з приводу її окупності в майбутньому. У загальних рисах реплікація бази даних - це об'єднання потужностей набору комп'ютерів для підвищення коефіцієнта продуктивності системи та її відмовостійкості.

Першого ефекту можна добитися шляхом перекидання частини запитів на резервний сервер, а другий - шляхом створення локальної копії бази даних, яка в разі необхідності замінить головну базу. Найбільш простий і розповсюджений спосіб організації реплікації - це реплікація MySQL. Причому MySQL підтримує цей функціонал з 3.23.15-версії і тільки односторонню реплікацію. В даному випадку створюється спеціальна конфігурація «master-slave», причому головний сервер (master) може також одночасно бути і підлеглим сервером (slave).

Реплікація - прийом, який застосовується в архітектурі систем що працюють під навантаженням, результатом якого є розподіл навантаження при роботі з однією базою даних на кілька серверів. MySQL MASTER SLAVE реплікація використовується частіше, але застосовується і другий тип реплікації - Master-Master.

Реплікація (дублювання) даних в MySQL - простий механізм, який дозволяє тримати в декількох місцях однакові, актуальні таблиці, причому

робота даного механізму не впливає на продуктивність MySQL-сервера. Особливо корисною можливістю є те, що MySQL-сервери можуть знаходиться в різних локаціях - наприклад, в різних країнах.

За допомогою додаткових інструментів (mysql-proxy, наприклад) можна використовувати slave-сервери для читання даних, а запис робити на master. Іншими словами, можна легко створити потужний масштабований і розподілений кластер.Зазвичай немає необхідності slave робити настільки ж продуктивними, як і master.

Щоб уникнути можливої, але вкрай малоймовірною несумісності, рекомендується використовувати одну і ту ж версію MySQL-сервера на всіх вузлах.

Реплікація в першу чергу потрібна як підстраховка на випадок, якщо основний mysql-сервер вийде з ладу, тоді можна переключитися на slave-сервер і продовжити роботу. По-друге, це можливість зменшити навантаження на основний сервер Mysql, використовуючи master-сервер тільки для запису, а операції на читання виконувати на slave-сервері. Як відбувається реплікація? Master-сервер пише binlog-і, в яких вказує операції, які виконуються над базою даних (базами даних) і запам'ятовує зміщення в журналі від його початку до поточного запису (позицію). Slave-сервер підключається до master-а, порівнює значення позицій і зчитує зміни в журналі починаючи зі значення власної позиції і закінчуючи значенням позициии master-а. Зміни (команди) він застосовує до баз даних на slave-сервері.

Реплікація MySQL дозволяє підтримувати синхронізацію даних між декількома БД, використовуючи їх одночасно для підвищення ефективності, безпеки та швидкості роботи баз даних MySQL. Хоча система управління базами даних (СКБД) MySQL підтримує реплікацію "з коробки", перед її використанням щоразу потрібно спеціально конфігурувати її параметри.

Є безліч областей застосування техніки реплікації MySQL:

– створення резервних копій основного сервера (Master Server) СУБД MySQL, до того як зробити реплікацію, ви можете перемістити бази даних основного сервера на фізичний або віртуальний сервер, що підтримує роботу в підлеглому (Slave) режимі з копією БД;

– конфігурація підлеглого (Slave) сервера, для збільшення продуктивності веб-сайту або програми рекомендується встановити кілька підлеглих (Slave) баз даних: це збільшує доступність і швидкість доступу до даних за рахунок використання іншого механізму зберігання даних;

– резервне копіювання даних на підлеглих (Slave) серверах, важливо розуміти, що реплікація MySQL не може замінити регулярного резервного копіювання: ефективним шляхом тут є комбінування обох підходів - резервне копіювання даних з slave-серверів MySQL.

При конфігуруванні реплікації MySQL в режимі mater-slave важливо розуміти принципи роботи відповідного механізму в СУБД MySQL. В MySQL існують два основних механізми реплікації даних:

– операційна реплікація (Statement - based replication) записує всі запити SQL, зроблені до головної (Master) БД, і відтворює їх на підлеглих (Slave) серверах. Цей метод швидкий, але не дуже точний: наприклад, якщо ви вставляєте в поле таблиці випадкове значення, отримане за допомогою функції MySQL RAND (), результат, що зберігається в різних копіях БД, буде відрізнятися;

– построчная реплікація (Row-based replication) підраховує всі зміни головній (Master) БД і автоматично змінює записи в таблицях підпорядкованої (Slave) БД щоразу при виявленні нових даних. Цей підхід значно повільніше, ніж операційна реплікація MySQL, але надає високу мобільність даних .

1.3 Постановка задачі

Метою дипломного проекту є розроблення системи взаємодії Master-Slave серверів MySQL на основі платформи Percona. Для досягнення мети виконуються наступні задачі:

- огляд сучасних SQL СУБД;
- огляд основ реплікації Master-Slave серверів;
- перевід MySQL серверів на систему Percona MySQL;
- запуск серверів баз даних у виробничому режимі;
- конфігурація і налаштування Percona MySQL серверів;
- ввід в експлуатацію додаткових серверних процедур Master-Slave;
- моніторинг та діагностика серверів Percona MySQL.

Розгляд пставлених задач містяться в наступних розділах проекту.

2 ПРОЕКТУВАННЯ СИСТЕМИ MASTER-SLAVE НА ОСНОВІ PERCONA MYSQL

2.1 СУБД MySQL для логістичної фірми

SQL (англ. Structured query language - мова структурованих запитів) - декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не бувши мовою програмування в тому розумінні, як C або Pascal, SQL може формувати інтерактивні запити або, бувши вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

Сучасна логістика немислима без активного використання інформаційних технологій. Важко уявити собі формування та організацію роботи ланцюгів доставки товарів без інтенсивного, постійного оперативного обміну інформацією, без швидкого реагування на потреби ринку. Сьогодні практично неможливо забезпечити необхідну споживачами якість товарів і послуг без застосування інформаційних систем і програмних комплексів для аналізу, планування та підтримки прийняття комерційних рішень в логістичній системі. Більше того, саме завдяки розвитку інформаційних систем і технологій логістика стала домінуючою формою організації товароруху на високо-конкурентних ринках економічно розвинених країн.

База даних транспортного підприємства забезпечує:

- дані про виконані замовлення за певний проміжок часу;
- дані про незадіяні транспортні засоби;

– звіт про виконану водіями роботу за певний проміжок часу.

Транспортне підприємство виконує замовлення на перевезення вантажу. В замовленні вказано прізвище замовника, дата, адреса, адреса вантажу, адреса доставки, відстань, характер вантажу, вартість. Для виконання замовлень призначається певний транспортний засіб і водій.

2.2 Перевід MySQL Серверів на систему Percona MySQL Master-Slave сервери

Компанія Percona заснована в 2008 році двома вітчизняними розробниками, вже колишніми членами MySQL dev.team, Петром Зайцевим і Вадимом Ткаченко. Їх БД-сервер грунтується на кодової базі MySQL, яка доповнена власними численними, і що хочеться окремо підкреслити, досить якісними патчами, спрямованими на додавання нових функцій, підвищення стабільності роботи і зручності адміністрування. Але головна фішка проекту - інтеграція власної платформи XtraDB, заснованого на коді InnoDB-plugin і повністю сумісного з ним, але може похвалитися істотно більш високою продуктивністю. За замовчуванням XtraDB не змінює дефолтний формат зберігання даних в InnoDB. Користувач може прозоро перемикатися між XtraDB і InnoDB по кілька разів на день.

Крім того, Percona поставляє разом з цією платформою дуже потужну бекапову систему XtraBackup, яка дозволяє вирішувати і автоматизувати навіть найважчі задачі з області збереження довірених на зберігання сервера БД даних.

Сервер Percona має ряд переваг над базовою MySQL:

- сервер Percona працює більш рівномірно під навантаженням;
- сервер Percona поставляється з рядом додаткових показників;

ефективності, що дозволяє точно визначити, які користувачі, таблиці, індекси чи запити сповільнюють користувача;

- оскільки сервер Percona є замінником для MySQL,Користувач також отримує всі звичайні переваги MySQL та великої спільноти користувачів. Це означає, що будь-яка програма, розроблена для MySQL, може безперешкодно використовувати Сервер Percona без будь-яких змін.

Компанія «Перкона» (Percona) оголосила про вихід Percona Server для MySQL 5.7, розширеного варіанту популярної СУБД MySQL. Дана версія включає в себе всі покращення MySQL 5.7 Community Edition, випущеної компанією Oracle. Помимо цього, нова версія від Percona пропонує розширені можливості резервного копіювання, моніторингу, а також оптимізованого під операції запису платформи нового покоління TokuDB, що забезпечує більш високу ступінь стиснення даних та ефективності роботи в області. Percona Server для MySQL 5.7 також зберігає курс на підвищення продуктивності, поліпшення масштабованості та забезпечення високої доступності.

Статистика по таблиці, користувачам і індексам надається з мінімальним впливом на продуктивність і забезпечує більш повне розуміння активності на серверах, а також дозволяє визначити джерела навантаження.

Розподіл часу відповіді на запит дає детальну інформацію за запитами на читання і запис окремо для кожного виду запитів і протягом заданого проміжку часу.

Пул потоків (pool thread) забезпечує масштабування до більш ніж 10 000 з'єднань на сервер.

Плагін аутентифікації PAM (Pluggable Authentication Modules) виконує аутентифікацію користувачів MySQL.

Плагін аудита надає можливість моніторингу на основі політики та логінізації активності об'єднань і виконання запитів.

Блокування для резервного копіювання забезпечують створення резервних копій без істотного зниження продуктивності.

Percona Server для MySQL включає оптимізований під операції запис платформи нового покоління TokuDB - єдиний транзакційний запас для MySQL. Він забезпечує більш високий ступінь стиснення даних і знижує вартість роботи в області, а також підходить для застосування в додатках «Інтернету речей» (IoT). Стиснення даних скорочує використання дискового простору, що, в свою чергу, знижує сукупну вартість володіння.

Percona Server для MySQL пропонує розширені можливості повноцінної зовнішньої аутентифікації, логінування аудиту та масштабування на рівні пулу потоків. Аналогічні функціональні можливості доступні тільки в комерційній версії MySQL Enterprise Edition від Oracle.

Використання Percona Server для MySQL 5.7 з Percona XtraBackup дає можливості, не доступні для інших варіантів MySQL. Швидкі інкрементальні резервні копії створюються на основі бітової карти швидше, ніж за допомогою MySQL Enterprise Backup.

Percona Server для MySQL 5.7 інтегрується з цілим набором інших продуктів від Percona, включаючи Percona XtraDB Cluster, Percona XtraBackup, Percona Toolkit та ін.

Percona Server для MySQL 5.7 створено на основі MySQL 5.7 Community Edition і є її повністю сумісною альтернативою. Використовуючи відгуки та вимоги до продуктивності своїх клієнтів, Percona планує вдосконалити Percona Server для MySQL 5.7 та адаптувати його до зростаючого робочого навантаження. Для компаній це буде означати розширення можливостей MySQL як продукту Oracle, яке вони зможуть використовувати у корпоративних розробок без збільшення витрат.

2.3 Оптимізація взаємодії транзакцій між серверами баз даних

За останні 20 років бази даних отримали загальне визнання, а сфера їх застосування феноменально розширилася. Зниження вартості запам’ятовуючих пристроїв і супутніх технологій зумовило значний зростання обсягів збережених даних і поширення БД. Легко доступні технології зробили можливим впровадження БД в різні продукти і більш тривале зберігання даних на підприємствах. Як наслідок, підприємства прагнуть консолідувати свої сервери.

В процесі функціонування багатокористувацької БД необхідно забезпечити такі умови, щоб при паралельній обробці декількох завдань дії одних користувачів не чинили непередбаченого впливу на роботу інших. Іноді мета полягає в тому, щоб в умовах паралельної обробки користувач отримав той же результат, як у випадку, якби він був єдиним користувачем БД. В інших випадках мається на увазі, що дії різних користувачів будуть впливати один на одного, але очікуваним чином.

На жаль, не існує стратегії організації паралельної обробки даних, що ідеально підходить для всіх випадків, - всі способи припускають певний компроміс між зручністю роботи і забезпеченням несуперечності БД.

Важлива складова частина вирішення зазначеного завдання - використання механізму транзакцій. Транзакція - це послідовність дій з БД, в якій або всі дії виконуються успішно, або не виконується жодна з них. Транзакція є атомарною (англ. Atomic), вона виконується як єдине ціле.

Транзакцію можна розглядати як перетворення одного логічно узгодженого стану БД в інший, причому в проміжних точках (тобто під час виконання транзакції) БД може перебувати в неузгоджену стані.

Для вказівки рамок транзакції прикладна програма повинна дати команди на початок транзакції, збереження транзакції, відкат транзакції.

Транзакція починається з оператора BEGIN TRANSACTION (в MS SQL Server допустимо скорочення BEGIN TRAN, а деякі СУБД використовують інструкцію START TRANSACTION або SET TRANSACTION). Закінчується транзакція виконанням оператора COMMIT (в запису без скорочень для MS SQL Server - COMMIT TRANSACTION або COMMIT TRAN) або оператора ROLLBACK (ROLLBACK TRANSACTION, ROLLBACK TRAN). Оператор COMMIT встановлює точку фіксації, яка позначає закінчення логічної одиниці роботи. Виконання оператора ROLLBACK повертає БД до попередньої точки фіксації.

Транзакції володіють чотирма наступними властивостями (їх ще називають властивостями ACID - Atomicity, Consistency, Isolation, Durability):

– атомарність - транзакція є неподільною, виконуються або всі дії, або нічого;

– узгодженість - транзакція переводить один узгоджений стан БД в інший, без дотримання обов'язкової підтримки узгодженості в проміжних точках;

– ізоляція - навіть якщо запущено кілька конкуруючих між собою транзакцій, будь-яке оновлення, виконане однією з транзакцій, буде приховано від інших, поки зробилені зміни транзакцій не буде зафіксована;

– довговічність - коли транзакція виконана, її поновлення зберігаються, навіть якщо в наступний момент станеться збій системи.

Коли одночасно відбувається виконання декількох транзакцій, що використовують одні ті ж об'єкти БД, ці транзакції називаються паралельними або конкуруючими (англ. Concurrent transactions). Якщо не зробити додаткових дій, спрямованих на забезпечення перерахованих вище властивостей, поява конкуруючих транзакцій може призвести до таких проблем, як:

– проблема втраченого поновлення (англ. Lost update) – кілька

користувачів змінюють один і той ж рядок, грунтуючись на його початковому значенні, в результаті частина даних буде втрачено, так як кожна наступна транзакція перезапише зміни, зроблені попередньої;

– проблема "брудного читання" (англ. Dirty read) може виникнути при читанні транзакцією записів, які змінені, але ще не збережені в БД (дані змінені ще не завершилася транзакцією, яка після буде скасована);

– проблема неповторюваного читання (англ. Non-repeatable reads) – при повторному читанні даних, вже лічених раніше, транзакція виявляє модифікації або видалення, викликані іншою завершеною транзакцією; подібна зміна може порушити логіку роботи транзакції;

– проблема читання фантомів (англ. Phantom reads) з'являється, коли при повторному читанні даних транзакція виявляє нові рядки, вставлені або змінені іншою транзакцією, завершеною після попереднього читання цього набору даних.

Щоб уникнути подібних проблем, необхідно забезпечити ізоляцію транзакцій. Основний засіб тут - блокування ресурсів. СУБД не дає конкуруючим транзакціям отримувати копії одного і того ж запису, коли передбачається швидку зміну цього запису.

Блокування можуть накладатися або автоматично, з ініціативи СУБД, або командою, яка віддається СУБД прикладної програмою або запитом користувача. Накладні СУБД блокування називаються неявними блокуваннями (англ. Implicit locks), а блокування, що накладаються за командою користувача - явними блокуваннями (англ. Explicit locks) .

Блокування, в залежності від СУБД, можуть накладатися на окремі рядки, сторінки (області фіксованого розміру, наприклад 8 Кбайт, що включають кілька рядків), таблиці, всю БД цілком. Розмір блокованого ресурсу називається глибиною деталізації блокування (англ. Lock granularity). При більшій глибині деталізації, тобто при блокуванні більших об'єктів, СУБД краще справляється з адмініструванням блокування, але такі

блокування частіше викликають конфлікти. При зменшенні розміру блокується фрагмента ймовірність виникнення конфлікту знижується, але такими блокуваннями складніше управляти: СУБД необхідно відстежувати набагато більше деталей.

Два основних типи блокування:

– монопольна (англ. Exclusive lock, X-lock) - блокуються всі види доступу до елементу;

– колективна (англ. Shared lock, S-lock) - елемент стає недоступним для зміни, але його можна прочитати.

Якщо одна транзакція встановила блокування, то інша транзакція при спробі накласти несумісне блокування або виконати заборонену блокуванням дію буде переведена в режим очікування до тих пір, поки блокування не буде знято.

У деяких випадках може виникати ситуація взаємного блокування (англ. Deadlock), коли дві транзакції блокують роботу один одного і жодна з них не може продовжити виконання, щоб в результаті зняти блокування. Такі ситуації повинні відслідковуватися СУБД: зазвичай одна з транзакцій примусово "відкочується", а інша продовжує роботу.

Для вирішення перерахованих вище проблем в стандарті SQL-92 були визначені чотири рівні ізоляції транзакцій. Розробник може вказати бажаний рівень ізоляції (або використовувати рівень, прийнятий за замовчуванням), а СУБД здійснює управління блокуваннями відповідно до заданим рівнем. Кожен наступний рівень підтримує вимоги попереднього і накладає додаткові обмеження.

Рівень "незавершене читання" (англ. READ UNCOMMITTED) дозволяє уникнути тільки втрачене поновлення. Цей рівень вимагає, щоб змінювати дані могла тільки одна транзакція; якщо іншій транзакції необхідно змінити ті ж дані, вона повинна очікувати завершення першої транзакції.

Рівень "завершене читання" (англ. READ COMMITTED) додатково дозволяє уникнути "брудного читання". Якщо транзакція почала зміну даних, то ніяка інша транзакція не зможе прочитати їх до завершення першої.

Рівень "відтворюється читання" (англ. REPEATABLE READ) додаток забезпечує повторюваність читання: якщо транзакція зчитує дані, то ніяка інша транзакція не зможе їх змінити, і при повторному читанні вони будуть знаходитися в тому ж стані.

Рівень "серіалізація" (англ. SERIALIZABLE), найвищий рівень, що дозволяє впоратися з проблемою читання фантомів. Якщо транзакція звертається до даних, то ніяка інша транзакція не зможе додати нові або змінити існуючі рядки, які можуть бути лічені при виконанні транзакції. Подібне блокування накладається не на конкретні рядки таблиці, а на рядки, що задовольняють певну логічну умову.

Серіалізація означає, що дві транзакції обробляються паралельно таким чином, що їх результати узгоджуються з результатами, які можуть бути отримані при послідовній обробці цих транзакцій.

Треба розуміти, що більш високий рівень ізоляції транзакцій призводить до більшої кількості блокувань і знижує загальну продуктивність системи баз даних. Різні СУБД можуть реалізовувати відрізняються набори рівнів ізоляції

Розглянемо тепер механізм забезпечення довговічності транзакції. Потрібно розуміти, що дані з таблиць перед зміною зчитуються з диска в пам'ять, і якийсь час всі зміни БД існують тільки в оперативній пам'яті. При збої системи (наприклад в результаті відключення електроживлення) відбувається втрата вмісту оперативної пам'яті.

Паралельно зі змінами даних з БД СУБД здійснює журнал транзакцій - внесення записів про транзакції в спеціальний файл реєстрації. Для того щоб після перезавантаження система могла визначити, які транзакції необхідно відкинути, а які - ні, проводяться наступні дії.

Збій системи може призвести не тільки до втрати вмісту пам'яті, але і до пошкодження файлів БД, - це так званий жорсткий збій. В цьому випадку необхідно відновлення БД з резервної копії. Якщо з моменту створення резервної копії збереглися всі журнали транзакцій, з'являється можливість привести базу в той стан, який у неї було на момент збою. Конкретні дії в даному випадку визначаються тими стратегіями резервного копіювання та відновлення, які пропонує СУБД.

3 ЗАПУСК СЕРВЕРІВ БАЗ ДАНИХ У ВИРОБНИЧОМУ РЕЖИМІ

3.1 Конфігурація і налаштування Percona MySQL серверів

Percona server - це збірка MySQL з включеним за замовчуванням XtraDB storage engine. Відрізняється від MySQL + InnoDB plugin кращою продуктивністю / масштабованість, особливо на сучасних багатоядерних серверах. Також покращена функціональність - більше будь-якої корисної для оптимізації статистики та ін. Збирається в варіантах на MySQL 5.0 і 5.1. Повністю сумісний з таблицями innodb, тобто можна переходити від innodb до xtradb і назад без проблем (якщо не використовувати деякі специфічні для xtradb функції, типу меншого розміру сторінки).

Версії Percona Server є вставними, сумісними лише з їх еквівалентними версіями MySQL. тобто MySQL 5.6 слід замінити лише на Percona Server 5.6. Спроба використовувати невідповідні версії може призвести до пошкодження таблиці або запобігання запуску сервера.

Щоб перевірити, яку версію ви запускаєте, спершу потрібно підключитися до MySQL за допомогою поточного пароля root,а потім знайти поточну встановлену версію:

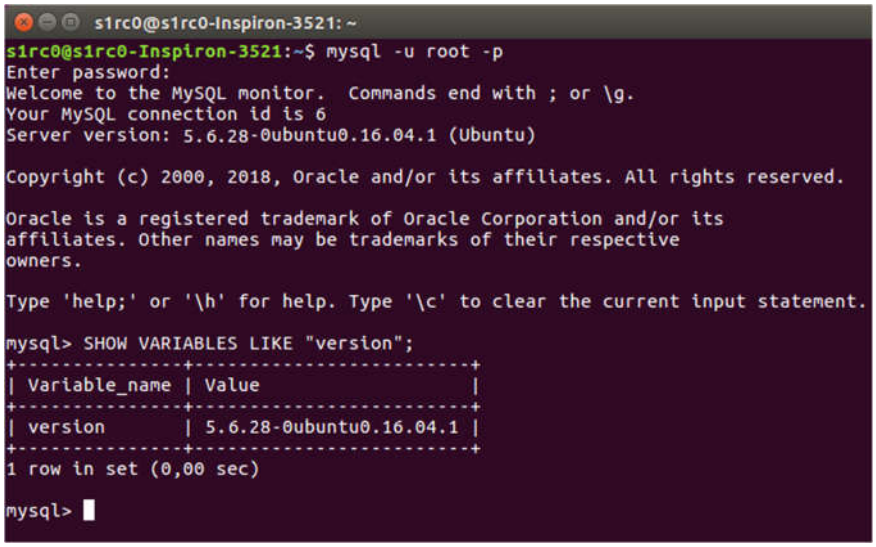


Рисунок 3.1 – Визначення версії MySQL

Це дасть нам змогу визначити, яку версію Percona Server 5.5 або 5.6 потрібно встановити. Якщо використовується MariaDB 10.0, то його слід замінити на Percona Server 5.6, а якщо використовується версія MySQL старше 5.5, то слід спочатку оновити MySQL до 5.5 або вище перед продовженням.

Сервер Percona, швидше за все, не в репозиторіях за замовчуванням у вибраному дистрибутиві Linux, оскільки Percona керує власними репозиторієм, щоб забезпечити якнайшвидше оновлення користувачів. Тому доведеться вручну додати сховища Percona APT або yum перед встановленням.

Пакунки Debian та Ubuntu, випущені Percona, підписані, що означає, що APT потрібно інформувати про новий ключ підпису:

Рисунок 3.2 – Створення новго ключа підпису

Далі потрібно додати нові сховища Percona, додавши наступні рядки до файлу /etc/apt/sources.list :

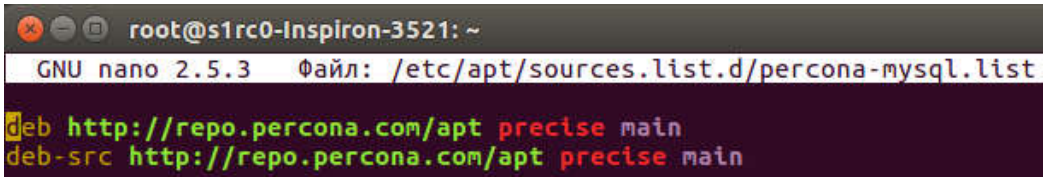


Рисунок 3.2 – Додавання нових сховищ Percona

Після збереження файлу джерел пакетів Percona їх слід закріпити, щоб переконатися, що пакети Percona завжди будуть пріоритетними над будь-якими пакетами з репозиторіїв за замовчуванням використовуваного дистрибутива. Для цього спочатку потрібно створимо новий файл для APT 00percona.pref.

Тепер потрібно відкрити цей файл обраним одним із текстових редакторів (Vim, nano та ін.) на /etc/apt/preferences.d/00percona.pref , додати наступні рядки та зберегти:

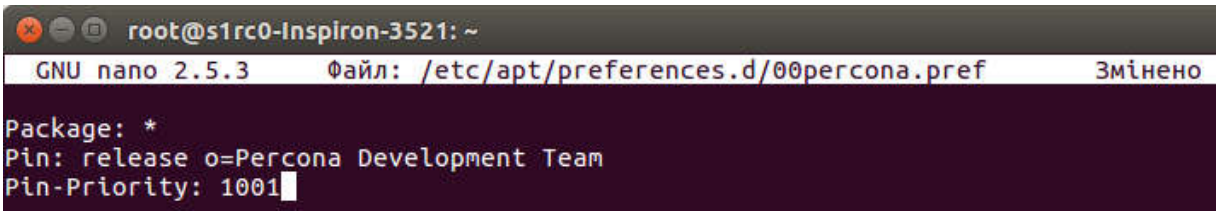


Рисунок 3.3 – Додавання нових рядків до файлу

Нарешті, коли джерела додані та закріплені, список пакетів може бути оновлений, і можна встановити пакет Percona Server.

Спочатку потрібно оновити локальний кеш. Після цього можна починати встановлювати серверний пакет:

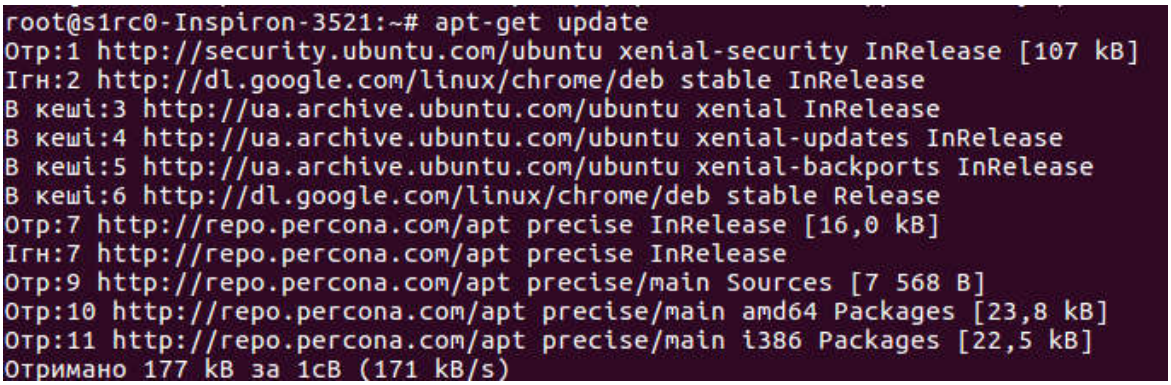


Рисунок 3.4 – Оновлення локального кешу

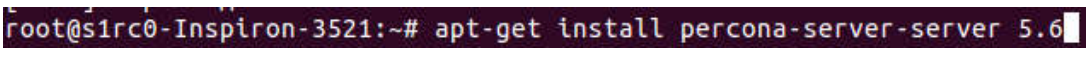


Рисунок 3.5 – Встановлення серверного пакету

При встановленні, у користувача запитають пароль root, який буде використовуватися для підключення до mysql сервера.

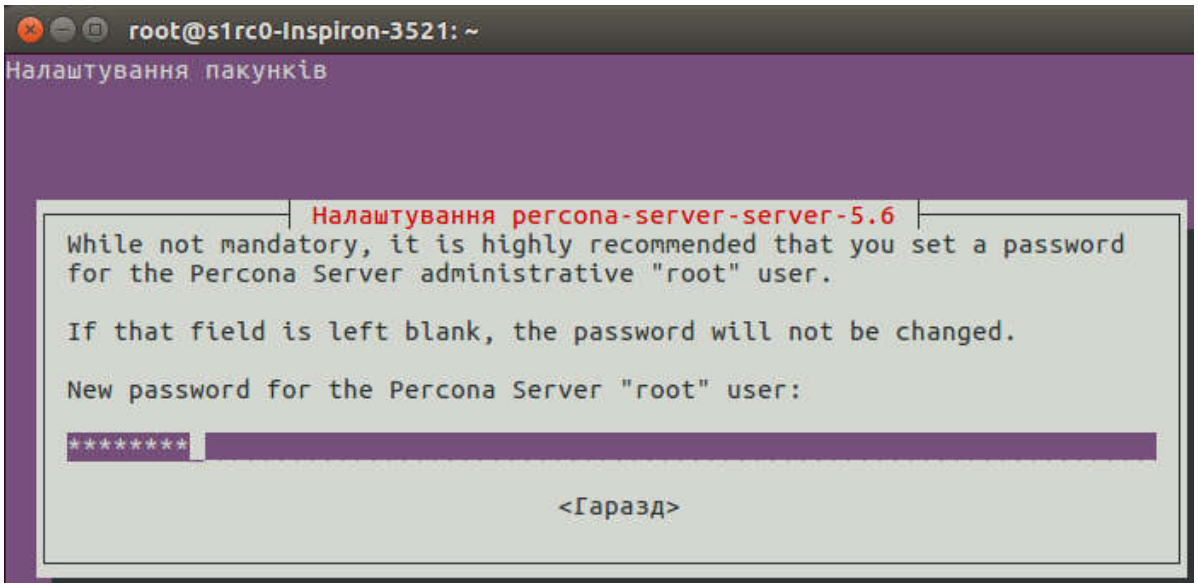


Рисунок 3.6 – Встановлення пароля root

Далі можна запустити mysql_secure_installation для поновлення пароля, видалення непотрібних БД та користувачів:

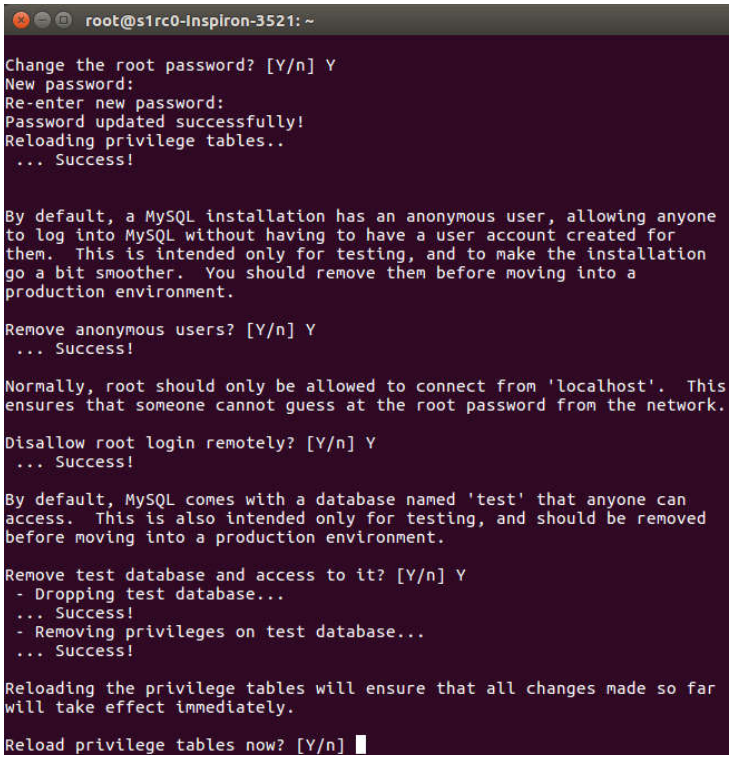


Рисунок 3.7 – Налаштування Percona-server

Тепер, коли сервер Percona був встановлений та запущений, можна переконатися, що встановлення пройшло успішно, виконавши кілька остаточних перевірок. По-перше, підключившись до бази даних за допомогою клієнта mysql , увійшовши за допомогою пароля кореневої облікового запису користувача бази даних:

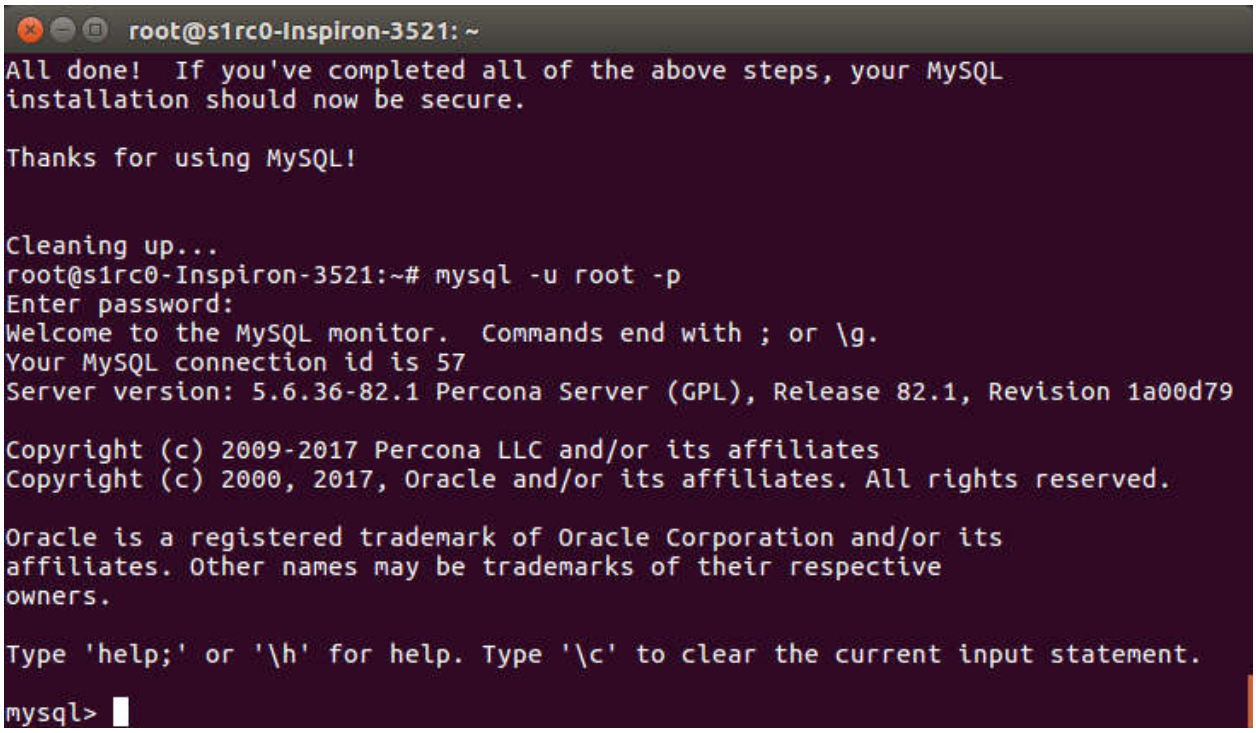


Рисунок 3.8 – Завершення встановлення

Відразу можна побачити, що серверна версія в тексті з'єднання тепер визначає сервер Percona. Використовуючи команду SHOW VARIABLES, можна розкопати подальші подробиці конкретної версії, яка була встановлена:

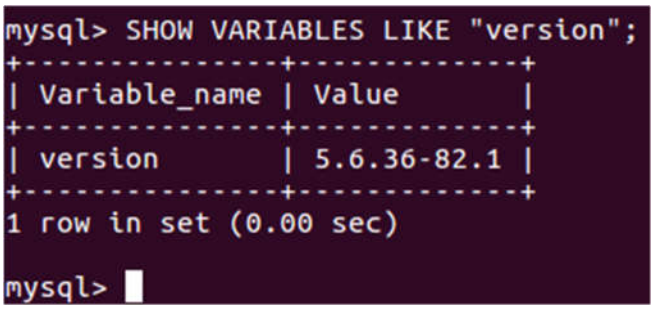


Рисунок 3.9 – Визначення версії сервера Percona

Оскільки Percona є сумісним з MySQL, команди MySQL працюватимуть так само, як і раніше.

Можна створити БД (testdb) на одному сервері і розгорнути на всіх інших. Підключившись на Master-сервері до mysql консолі, створюємо тестову БД.

Рисунок 3.10 – Створення тестової бази даних

Далі потрібно додати таблицю в нову базу даних ,і зразу ж додати один рядок в таблицю users.

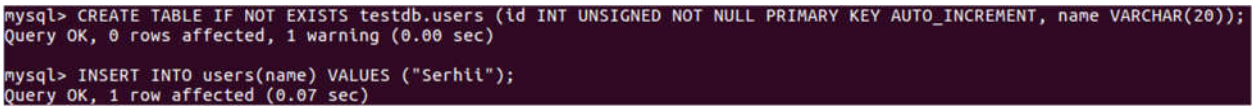


Рисунок 3.11 – Редагування бази даних

Далі можна перевірити,чи був доданий рядок у таблицю тестової бази даних.

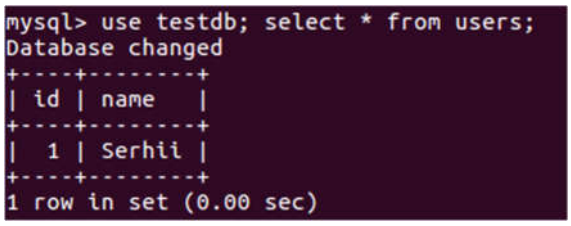


Рисунок 3.12 – Результа виконання команди select

Тепер потрібно створити користувача, який буде займатися репликацією. Для цього необхідно в mysql зайти в консоль і створити користувача replica з правами " replication slave ".

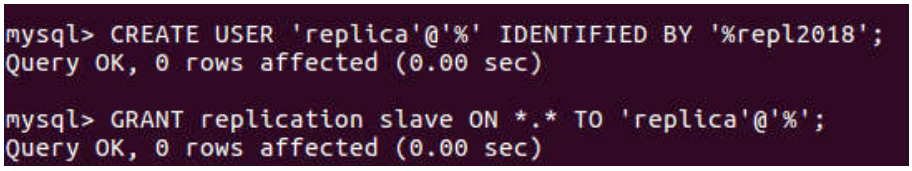


Рисунок 3.13 – Створення нового користувач

Ці дії потрібно виконати як і на Master-сервері,так і на Slave-сервері.

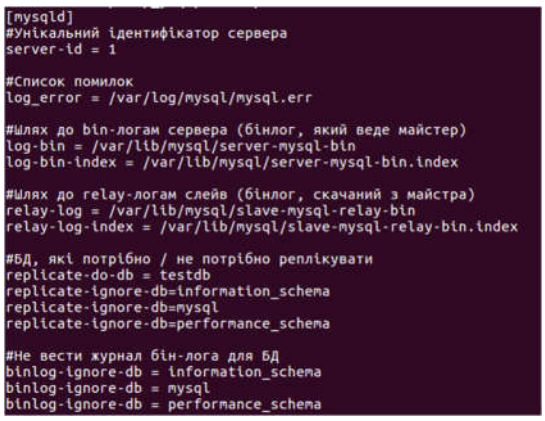
3.2 Ввід в експлуатацію додаткових серверних процедур Master-Slave

У MySQL існує два типи реплікації даних:

- Master-Slave;
- Master-Master.

При Master-Slave реплікації, на Master сервері дані додаються, видаляються і змінюються. Slave сервер стягує ці оновлення на себе і поступово виконує всі отримані запити. Якщо на Slave сервері буде додана нова таблиця або БД, то дані не потраплять на Master. При Master-Master реплікації дані,які потрапили на обидва сервери будуть об’єднані між собою.

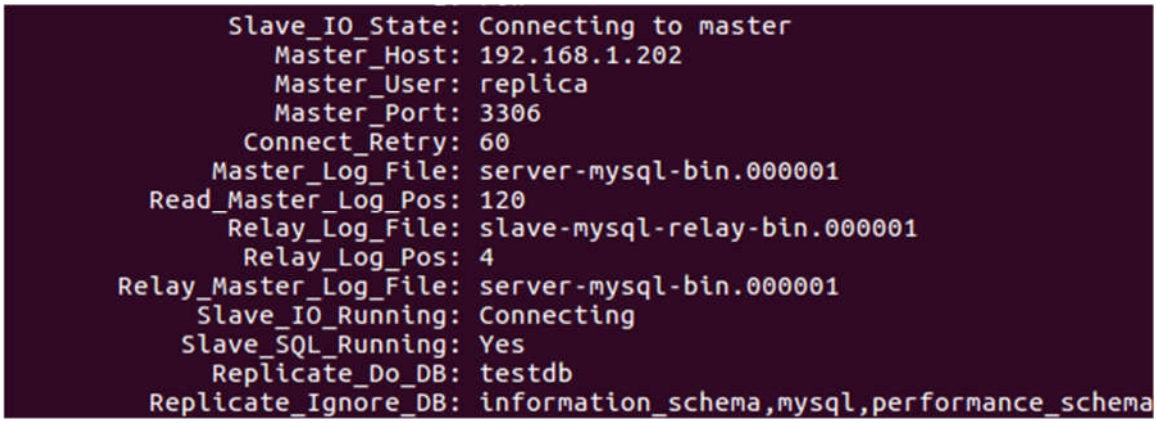
Спочатку потрібно налаштувати Master-сервер. Для цього необхідно додати конфігураційний файл /etc/mysql/my.cnf, який входить в реплікацію.



```
[mysqld]
#Унікальний ідентифікатор сервера
server-id = 1

#Список помилок
log_error = /var/log/mysql/mysql.err

#Шлях до bin-логам сервера (бінлог, який веде майстер)
log-bin = /var/lib/mysql/server-mysql-bin
log-bin-index = /var/lib/mysql/server-mysql-bin.index

#Шлях до relay-логам слейв (бінлог, скачаний з майстра)
relay-log = /var/lib/mysql/slave-mysql-relay-bin
relay-log-index = /var/lib/mysql/slave-mysql-relay-bin.index

#БД, які потрібно / не потрібно реплікувати
replicate-do-db = testdb
replicate-ignore-db=information_schema
replicate-ignore-db=mysql
replicate-ignore-db=performance_schema

#Не вести журнал бін-лога для БД
binlog-ignore-db = information_schema
binlog-ignore-db = mysql
binlog-ignore-db = performance_schema
```

Рисунок 3.14 – Редагування my.conf файлу

Перезавантаживши mysql, запускаємо слейв і дивимося статус.



```
             Slave_IO_State: Connecting to master
                Master_Host: 192.168.1.202
                Master_User: replica
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: server-mysql-bin.000001
        Read_Master_Log_Pos: 120
             Relay_Log_File: slave-mysql-relay-bin.000001
              Relay_Log_Pos: 4
      Relay_Master_Log_File: server-mysql-bin.000001
           Slave_IO_Running: Connecting
          Slave_SQL_Running: Yes
            Replicate_Do_DB: testdb
        Replicate_Ignore_DB: information_schema,mysql,performance_schema
```

Рисунок 3.15 – Старт слейву

З усього цього висновку потрібно звернути увагу на пункт Seconds_Behind_Master (час відставання репліки від мастера), далі на Slave_IO_State (має писати, що чекає новин від мастера), Slave_SQL_Running (Yes). Відставання репліки від мастера (Seconds_Behind_Master), має бути нульовим, але може рости.

```
        Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
```

Рисунок 3.16 – Значення Seconds_Behind_Master

Всі налаштування пройшли успішно. Тепер можна налаштовувати Slave-сервер. Для цього потрібно додати конфігураційний файл для mysql-слейву, який входить в реплікацію.

```
[mysqld]
#Унікальний ідентифікатор сервера
server-id = 2

#Список помилок
log_error = /var/log/mysql/mysql.err

#Шлях до relay-логам слейв (бінлог, скачаний з майстра)
relay-log = /var/lib/mysql/slave-mysql-relay-bin
relay-log-index = /var/lib/mysql/slave-mysql-relay-bin.inde

#БД, які потрібно / не потрібно реплицировать
replicate-do-db = testdb
replicate-ignore-db=information_schema
replicate-ignore-db=mysql
replicate-ignore-db=performance_schema

#Щоб не було конфліктів автоінкремента, говоримо сервера,
#щоб id генерувалися починаючи з 4-го додаючи по 10,
# Наприклад 11, 21, 31, 41 ...
auto_increment_increment = 10
auto_increment_offset = 1
```

Рисунок 3.17 – Додавання конфігураційного файлу для слейву

Тепер залишилося запустити репілкацію. У даному випадку Master-сервер повинен бути s1rc0-Inspiron-3521 , тому переходимо на сервер і дивимося MASTER_LOG_FILE і MASTER_LOG_POS .

```
root@s1rc0-Inspiron-3521:~# mysql -u root -p -e 'show master status;'
Enter password:
+------------------------+----------+--------------+-------------------------------------------+
| File                   | Position | Binlog_Do_DB | Binlog_Ignore_DB                          |
+------------------------+----------+--------------+-------------------------------------------+
| server-mysql-bin.000002 |      526 |              | information_schema,mysql,performance_schema |
+------------------------+----------+--------------+-------------------------------------------+
```

Рисунок 3.18 – Лог-файл Master- сервера

Отже MASTER_LOG_FILE = server-mysql-bin.000002 ,MASTER_LOG_POS = 526. Тепер перейшовши назад на Slave і налаштовуємо реплікацію.

```
root@s1rco-slave:~# mysql -u root -p -e "CHANGE MASTER TO MASTER_HOST = '192.168.1.201', MASTER_USER = 're
plica', MASTER_PASSWORD = '%repl2018', MASTER_LOG_FILE = 'server-mysql-bin.000002', MASTER_LOG_POS = 526;"
```

Рисунок 3.19 – Налаштування реплікації

Все пройшло успішно. Тестуємо весь наш ланцюжок реплікації. Перейшовши на перший Master-сервер і додамо новий запис в testdb.users .

```
root@s1rc0-Inspiron-3521:~# mysql -u root -p -e 'USE testdb; INSERT INTO users(name) VALUES ("Server1 record");'
Enter password:
root@s1rc0-Inspiron-3521:~# mysql -u root -p -e 'select * from testdb.users;'
Enter password:
+----+----------------+
| id | name           |
+----+----------------+
|  1 | Serhii         |
|  2 | Server1 record |
+----+----------------+
```

Рисунок 3.20 – Додавання новго запису в таблицю

Тепер необхідно перевірити чи добавився запис в таблицю на Slave-сервері.

```
root@s1rco-slave:~# mysql -u root -p -e 'select * from testdb.users;'
Enter password:
+----+----------------+
| id | name           |
+----+----------------+
|  1 | Serhii         |
| 11 | Server1 record |
+----+----------------+
```

Рисунок 3.21 – Результат на Slave-сервері

Як видно, все в порядку. Реплікація працює, як і передбачалося.

3.3 Моніторинг та діагностика серверів Percona MySQL

Percona Monitoring and Management (PMM) - це платформа з відкритим кодом для управління та моніторингу продуктивності MySQL та MongoDB. Він розроблений компанією Percona у співпраці з експертами в області керованих служб бази даних, підтримки та консалтингу. Платформу можна запускати у власному середовищі для максимальної безпеки та надійності. Вона забезпечує ретельний аналіз за часом для серверів MySQL та MongoDB для забезпечення максимально ефективного використання ваших даних.

Платформа PMM заснована на моделі клієнт-сервер, що забезпечує масштабованість. Вона включає в себе наступні модулі:

- PMM Client встановлений на кожному хості базі даних, який ви

хочете контролювати. Він збирає показники сервера, загальні системні показники та дані Query Analytics для повного огляду продуктивності.

- PMM Server є центральною частиною PMM, яка агрегує зібрані

дані та представляє її у вигляді таблиць, інформаційних панелей та графіків у веб-інтерфейсі.

Доступ до веб-інтерфейсу Percona Monitoring and Management можна отримати використовуючи IP-адресу хоста, на якому працює сервер PMM.

Домашня сторінка PMM, яка відкривається, дає загальний огляд середовища, який був налаштували для моніторингу за допомогою інструмента pmm-admin .

З домашньої сторінки PMM можна отримати доступ до певних інструментів моніторингу або інформаційних панелей. На кожній інформаційній панелі представлена сукупність показників. Це графіки певного типу, які представляють один конкретний аспект, який показує, як змінюються показники часу з часом. Домашню сторінку pmm-admin зображено на рисунку 3.18.

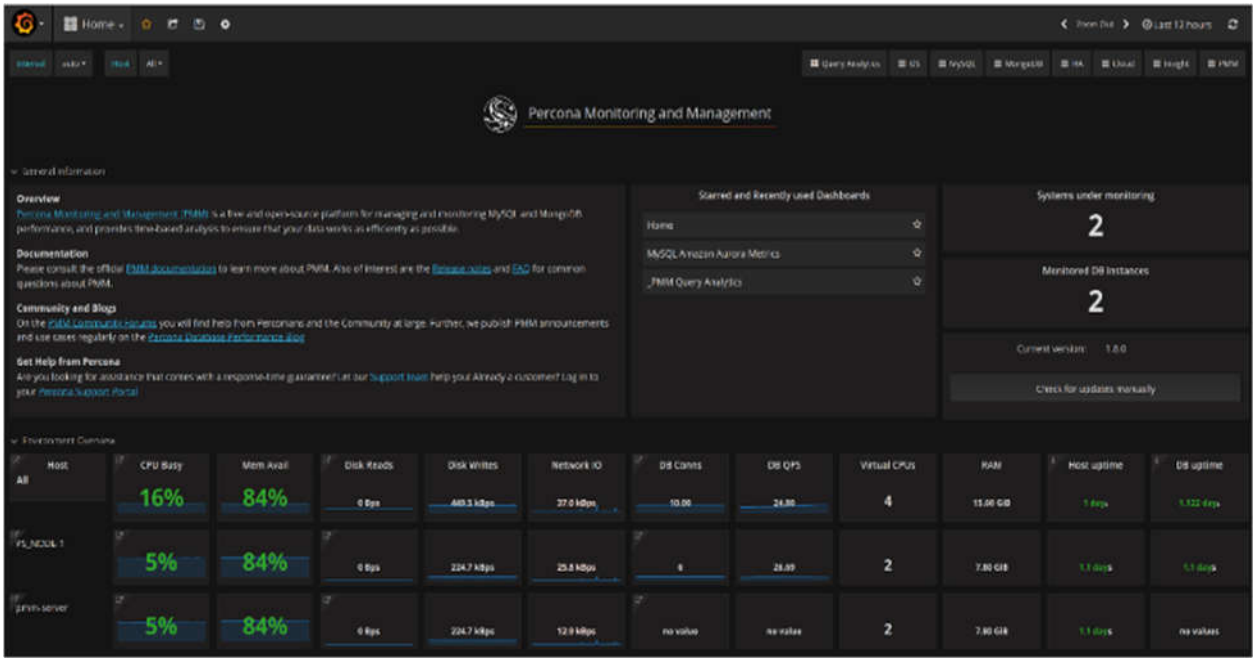


Рисунок 3.22 – Домашня Сторінка PMM

За замовчуванням домашня сторінка PMM містить список найновіших інформаційних панелей та корисних посилань на інформацію, яка може бути потрібною для кращого розуміння PMM.

Домашня сторінка PMM відображає всі хости, які були налаштовані для моніторингу, а також важливі відомості про їх продуктивність, такі як завантаження ЦП, продуктивність диска або активність мережі.

ВИСНОВКИ

В результаті виконання дипломного проекту розроблено систему взаємодії Master-Slave серверів MySQL на основі платформи Percona. Розв’язано наступні задачі.

1. Проведено аналітичний огляд сучасних SQL СУБД: MySQL, PostgreSQL, Oracle, Microsoft SQL , DB2, Centaur, Ingres, Teradata Database. Перевагами MySQL є: пропонується багато функцій, навіть у безкоштовній версії; пакет MySQL включений в стандартні репозиторії ОС Linux, що дозволяє встановлювати її елементарно.
2. Проведено огляд основ реплікації Master-Slave серверів.
3. Здійснено перевід MySQL серверів на систему Percona MySQL.
4. Здійснено запуск серверів баз даних у виробничому режимі.
5. Здійснено конфігурацію і налаштування Percona MySQL серверів. Введено в експлуатацію додаткові серверні процедури Master-Slave.
6. Проведено моніторинг та діагностику серверів Percona MySQL.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов: 3-е изд. / В. Г Олифер., Н. А. Олифер // – СПб.: Питер, 2006.

2. Семенов Ю. А. Телекоммуникационные технологии. Интернет-университет информационных технологий [Электронный ресурс]. – Режим доступа: http://book.itep.ru

3. Зайцев Д. А. Моделирование телекоммуникационных сетей в системе NS. / Д. А. Зайцев, Т. Н. Шинкарчук // Наукові праці ОНАЗ ім. О. С. Попова. – 2006.– № 2.

4. Panwar Li. Y. S. On the Performance of MPLS TE Queues for QoS Routing // Panwar Li. Y. Liu C.J. Simulation series. – 2004. – Vol. 36; part 3. – P. 170–174.

5. Зайченко Ю. П. Анализ и оптимизация характеристик сетей MPLS по заданным показателям качества / Ю. П. Зайченко, Ахмед А. М. Шарадка // Вісник національного технічного університету України КПІ сер. Інформатика управління та обчислювальна техніка. Вип. 43. – 113–123 с.

6. Будылдина Н. В. Разработка программного обеспечения для оптимизации мультисервисных сетей / Н. В. Будылдина., П. А. Коновалов // Открытое образование, июнь 2006. – 58 с.

7. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Е.А. Кучерявый // – М.: Наука и Техника. – 2007. – 336 с.

8. Аткинсон Л. Mysql. Библиотека профессионала – М Энергоатомиздат, 2002 – 496 с.

9. Дейт К. Руководство по реляционной СУБД DB2 – М.: Финансы и статистика, 1988 – 320 с.

10. Мейер М.Теория реляционных баз данных – М.: Мир, 1987 – 608 с.

11. Семантична модель: База даних [Електронний ресурс] Режим доступу: http://citforum.ru/database/advanced_intro/27.shtml.

12. Риккарди Г. Системы баз данных. Теория и практика использования в Internet – М.:Вильямс, 2001 – 240с.

13. ДейтК. Дж. Введение в системы баз даннях – СПб: Изд-во "Пітер", 2005 – 1315с.

14. Роберт І.В. Сучасні інформаційні технології в освіті: дидактичні проблеми, перспективи використання ― М.: Школа-Пресс, 1994― 205с.

15. Розділ 2.Основи UML – діаграм: [Електронний ресурс] Режим доступу:https://docs.kde.org/trunk4/uk/kdesdk/umbrello/uml-basics.html

16. Мюллер Р.Дж. Базы данных и UML. Проектирование / Перевод. с англ. Е. Молодцова. – М.: Издательство “Лори”,2002 – 432с.

17. Острей О.Р. Діаграми класів UML як засіб моделювання інформаційної системи моніторингу освіти / М.: - 2008. № 2. – С.85-89.

18. Електронна енциклопедія Вікіпедія: JSON[Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/JSON

19. Компонентне або модульне тестування: [Електронний ресурс] Режим доступу: http://www.protesting.ru/testing/levels/component.html

20. Електронна енциклопедія Вікіпедія: Модульне тестування [Електронний ресурс] Режим доступу: https://uk.wikipedia.org/wiki/Модульне

21. Електронна енциклопедія Вікіпедія: Список кодів стану HTTP [Електронний ресурс] Режим доступу:https://uk.wikipedia.org/wiki/список

22. Шапошников І. Web-сайт своїми руками./ І. Шапошников – СПб: Изд-во "Пітер", 2002 – 390с.

23. Гаевский А. Ю. Самоучитель по созданию Web-страниц: HTML, JavaScript, Dynamic HTML / А. Ю. Гаевский, В. А.Романовский. — К.: А.С.К., 2002 — 472с.

24. Методичні вказівки до написання техніко – економічного розділу для дипломних проектів на здобуття освітньо – кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» / І.Р. Паздрій. – Тернопіль: ТНЕУ, 2015. – 36с.

25. Методичні рекомендації до виконання дипломного проекту з освітньо – кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / Л.О. Дубчак

О.М. Березький, Р.Б Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. – Тернопіль: ТНЕУ, 2016.- 60с.