ВСТУП

Актуальність: На даний момент на ринку є потреба у швидкодіючих системах для організації документообігу на підприємствах, з ефективною організацією баз даних. В сучасних умовах на ринку існують рішення лише для локальної роботи окремих підприємств без можливості їх синхронізації з іншими. Дана система реалізує сумісну роботу юридичних відділів компаній через синхронізацію з «хмарним» сервісом.

Основною проблемою у створенні систем управління портфелем юридичних процесів є мінімізація об'єму дискового простору для зберігання різнотипних даних. Практично в кожній організації постійно зростає обсяг оброблюваних і збережуваних даних. На тлі підвищення вимог ІТ та бізнесу до безпеки, швидкодії та гнучкості таке зростання може привести до появи значних проблем.

Наразі існуючі рішення для збереження нетипізованих даних не забезпечують можливість використання всіх переваг ефективної інфраструктури зберігання.

Враховуючи постійних ріст потреби у швидкодіючих системах для організації документообігу на підприємствах є напрочуд актуальною така система, що дозволить зменшити витрати на утримання серверів баз даних.

Мета: Підвищення ефективності автоматизованої обробки нетипізованих даних інформаційних систем.

Задачі дослідження:

* Аналіз засобів та підходів, що використовуються для автоматизованої обробки даних в інформаційних системах
* вдосконалити методи збереження нетипізованих даних в інформаційних системах
* впровадити розроблений інформаційний метод в інформаційній системі управління портфелем юридичних процесів

Об’єкт: Об’єктом дослідження є процес автоматизованої обробки даних інформаційних систем з нетипізованими даними..

Предмет: Предметом є методи автоматизованої обробки даних інформаційних систем з нетипізованими даними.

Методи дослідження:

* методи теорії реляційних баз даних для розробки структури бази даних
* методи системного аналізу та математичного моделювання для забезпечення перевірки на коректність і подальшого використання і представлення у бажаному вигляді первинної інформації

Наукова новизна:

* Вдосконалено методи зберігання нетипізованих даних у реляційних базах даних, що на відміну від існуючих дозволяє зменшувати об'єм дискового простору для зберігання різнотипних даних та підвищити ефективність і економічну доцільність використання хмарних сервісів, за рахунок розробки власної структури даних на основі реляційної моделі даних.
* Розроблено метод формування структури бази даних, що на відміну від існуючих дозволяє підвищити швидкодію вибірки даних, за рахунок денормалізації структури бази даних.

Практичне значення:Розроблено алгоритми та програмне забезпечення, що дозволяє підвищити ефективність використання дискового простору при зберіганні даних у реляційній базі даних.

Апробація: Стаття в журналі Запорізького національного технічного університету "Радіоелектроніка, інформатика, управління\*.

Участь у весняній конференції Вінницького національного технічного університету 2014 року\*\*.

\* - подано до друку

\*\* - подано до друку

1. СУЧАСНІ ПІДХОДИ ДО ЗБЕРІГАННЯ І ОБРОБКИ ДАНИХ

Ми живемо в інформаційний вік. Нелегко виміряти загальний обсяг електронних даних, але за оцінками ІDС розмір «цифрового всесвіту» в 2006 р становив 0.18 зеттабайт, а до 2011 р повинен був досягти 1.8 зеттабайт продемонструвавши десятикратне зростання за 5 років! Ось тільки кілька прикладів джерел таких обсягів [1]:

1. Нью-Йоркська фондова біржа генерує близько терабайта даних в день.
2. Обсяг сховища соціальної мережі Facebook кожен день збільшується на 500 терабайт.
3. Проект Internet Archive вже зберігає 2 петабайта даних і приростає 20 терабайтами на місяць.
4. Експерименти на Великому адронному колайдері можуть генерувати близько петабайта даних н секунду!

Стрімко зростаючий обсяг інформації ставить перед нами нові складні завдання по організації її зберігання і обробки.

1.1 Що таке «великі дані»?

Отже, даних дуже багато. Але що таке «багато»? Де той поріг, подолавши який, дані стають «великими»? Часто використовується характеристика, дана дослідницькою компанією Gatner: «Великі дані» характеризуються об'ємом, різноманітністю і швидкістю, з якою структуровані і неструктуровані дані надходять по мережах передачі в процесори і сховища, поряд з процесами перетворення цих даних і цінну для бізнесу інформацію » [2].

Як видно з цього визначення, великі дані мають чотири основні характеристики: об'єм, різноманітність, швидкість і цінність.

Розглянемо їх докладніше:

1. Обсяг. Наростаюча кількість даних, що створюються як людьми, так і машинами, пред'являє до IT інфраструктурі нові вимоги відносно зберігання, обробки та надання доступу.
2. Різноманітність. Дані містять різноманітну інформацію, представлену різними структурами. З усім цим, від логів доступу до веб-серверу до операцій по кредитних картах, від результатів наукових експериментів до фотографій і відео, необхідно вміти працювати.
3. Швидкість. Важливо усвідомлювати, що під швидкістю розуміється не тільки швидкість, з якою дані надходять в сховище, а й швидкість з якою важлива інформація з цих даних витягується.
4. Цінність. Великі обсяги даних - це цінний ресурс. але ще ціннішим він стає, якщо дозволяє відповідати на нагальні питання або питання, які можуть виникнути в майбутньому.

1.2 Підходи до роботи з великими даними

Так склалося, що інструменти, що існували до недавнього часу, виявилися не здатні впоратися з великими обсягами. Про проблеми, що прийшли з епохою великих даних, способах їх подолання та нових інструментах поговоримо в наступних розділах.

* + 1. Реляційні СУБД

База даних (скорочено — БД) — впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головне завдання БД — гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування нею.

Реляційна база даних — база даних, основана на реляційній моделі даних.

Реляційна модель даних — логічна модель даних. Вперше була запропонована британським ученим співробітником компанії IBM Едгаром Франком Коддом (E. F. Codd) в 1970 році в статті «A Relational Model of Data for Large Shared Data Banks». В даний час ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні комерційні системи керування базами даних (СКБД).

У реляційній моделі досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій. У згаданій статті Е. Ф. Кодда стверджується, що «реляційна модель надає засоби опису даних на основі тільки їх природної структури, тобто без потреби введення якоїсь додаткової структури для цілей машинного представлення». Іншими словами, подання даних не залежить від способу їх фізичної організації. Це забезпечується за рахунок використання математичного поняття відношення (сама назва «реляційна» походить від англійського relation — «відношення»).

До складу реляційної моделі даних зазвичай включають теорію нормалізації. Крістофер Дейт визначив три складові частини реляційної моделі даних:

* структурна
* маніпуляційна
* цілісна

Структурна частина моделі визначає, що єдиною структурою даних є нормалізоване n-арне відношення. Відношення зручно представляти у формі таблиць, де кожен рядок є кортеж, а кожен стовпець — атрибут, визначений на деякому домені. Даний неформальний підхід до поняття відношення дає більш звичну для розробників і користувачів форму представлення, де реляційна база даних являє собою кінцевий набір таблиць.

Маніпуляційна частина моделі визначає два фундаментальних механізми маніпулювання даними — реляційну алгебра і реляційне числення. Основною функцією маніпуляційної частини реляційної моделі є забезпечення заходів реляційності будь-якої конкретної мови реляційних БД: мова називається реляційною, якщо вона має не меншу виразність і потужність, ніж реляційна алгебра або реляційне числення.

Цілісна частина моделі визначає вимоги цілісності сутностей і цілісності посилань. Перша вимога полягає в тому, що будь-який кортеж будь-якого відношення відмінний від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення має володіти первинним ключем. Вимога цілісності щодо посилань, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється у відношенні, на яке веде посилання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа повинно бути невизначеним (тобто ні на що не вказувати).

Можна провести аналогію між елементами реляційної моделі даних і елементами моделі «сутність-зв'язок». Реляційні відносини відповідають наборам сутностей, а кортежі — сутностям. Тому, як і в моделі «сутність-зв'язок», стовпці в таблиці, що представляє реляційне відношення, називають атрибутами.

Кожен атрибут визначений на домені, тому домен можна розглядати як множина допустимих значень даного атрибуту. Кілька атрибутів одних відношень і навіть атрибути різних відношень можуть бути визначені на одному і тому ж домені.

Іменована множина пар «ім'я атрибута — ім'я домену» називається схемою відношення. Потужність цієї множини — називають ступенем чи «арністю» відносини. Набір іменованих схем відносин являє собою схему бази даних.

Атрибут, значення якого однозначно ідентифікує кортежі, називається ключовим (або просто ключем). Якщо кортежі ідентифікуються тільки зчепленням значень декількох атрибутів, то говорять, що відношення має складовий ключ. Ставлення може містити кілька ключів. Завжди один із ключів оголошується первинним, його значення не можуть оновлюватися. Всі інші ключі відносини називаються можливими ключами.

На відміну від ієрархічної і мережної моделей даних в реляційної відсутнє поняття групових відношень. Для відображення асоціацій між кортежами різних відносин використовується дублювання їх ключів.

Переваги реляційної моделі:

* простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
* суворі правила проектування, які базуються на математичному апараті;
* повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
* для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.
* Недоліки реляційної моделі:
* далеко не завжди предметна область може бути представлена у вигляді «таблиць»;
* в результаті логічного проектування з'являється множина «таблиць». Це призводить до труднощів розуміння структури даних;
* БД займає відносно багато зовнішньої пам'яті;
* відносно низька швидкість доступу до даних.

СУБД повинна забезпечувати реляційну модель роботи з даними. Сама модель має на увазі певний тип зв'язку між сутностями з різних таблиць. Щоб зберігати і працювати з даними, такий тип СУБД повинен мати певну структуру (таблиці). В таблицях кожен стовпець може містити дані різного типу. Кожна запис складається з безлічі атрибутів (стовпців) і має унікальний ключ, що зберігається в тій же таблиці - всі ці дані взаємопов'язані між собою, як описано в реляційної моделі.

Відношення в базах даних можна розглядати як математичну иножину, що містить в собі число атрибутів, які сумарно представляють собою базу даних і інформацію, що зберігається в ній.

При створенні структури таблиці кожне поле запису повинне мати заздалегідь описаний тип (наприклад: рядок, цілочисельне значення і т.д.). Всі СУБД мають у своєму складі різні типи даних, які не завжди взаємозамінні. При роботі з СУБД завжди доводиться стикатися з подібними обмеженнями.

На сьогоднішній день найпопулярнішими і основними реляційними базами даних є:

1. **SQLite**- дуже потужна вбудовувана система управління
2. MySQL- найпопулярніша і поширена СУБД
3. PostgreSQL- найбільш функціональна СУБД
   * + 1. SQLite

Легко вбудовувана в додатки база даних. Так як це система базується на файлах, то вона надає досить широкий набір інструментів для роботи з нею, в порівнянні з мережевими СУБД. При роботі з цією СУБД звернення відбуваються безпосередньо до файлів (в ці файлах зберігаються дані), замість портів і сокетов в мережевих СУБД. Саме тому SQLite дуже бисть, а також потужна завдяки технологіям обслуговуючих бібліотек.

**Типи даних SQLite:**

NULL - значение NULL

INTEGER - знаковое целочисленное значение, использует 1, 2, 3, 4, 6, или 8 байт в зависимости от порядка числа

REAL - число с плавающей точкой, занимает 8 байт для хранения числа в формате IEEE

TEXT - текстовая строка, при хранении используются кодировки UTF-8, UTF-16BE или UTF-16LE

BLOB - тип данных BLOB, массив двоичных данных (предназначенный, в первую очередь, для хранения изображений, аудио и видео).

Переваги SQLite:

* Файлова структура - вся база даних складається з одного файлу, тому її дуже легко переносити на різні машини
* Використовувані стандарти - хоча може здатися, що ця СУБД примітивна, але вона використовує SQL. Деякі особливості опущеними (RIGHT OUTER JOIN або FOR EACH STATEMENT), але основні все-таки підтримуються
* Відмінна при розробці та тестуванні - в процесі розробки додатків часто з'являється необхідність масштабування. SQLite пропонує все що необхідно для цих цілей, тому що складається всього з одного файлу і бібліотеки написаної мовою C.

Недоліки SQLite:

### відсутність системи користувачів - більші СУБД включають в свій склад системи управління правами доступу користувачів. Зазвичай застосування цієї функції не так критично, так як ця СУБД використовується в невеликих додатках.

### відсутність можливості збільшення продуктивності - знову, виходячи з проектування, досить складно вичавити щось більш продуктивне з цієї СУБД.

Коли використовувати SQLite

* вбудовані додатки - якщо вам важлива можливість легкого перенесення програми і не важлива масштабованість. Наприклад однокористувацькі додатки, мобільні додатки або ігри
* прямий доступ до диска - при необхідності безпосередньо звертатися до диска ви можете виграти при переході на цю СУБД в функціоналі і простоті використання SQL мови
* Тестування - використання додаткових процесів при тестуванні функціоналу, дуже уповільнює додаток.

Коли відмовитися від SQLite

* Мультиплеєрні додатки - якщо вам необхідно забезпечити доступ до даних для декількох користувачів, та й до того ж розрізняти їх з прав доступу, то, напевно, повноцінна СУБД (наприклад: MySQL) буде більш логічним вибором
* Запис великих обсягів даних - одне з обмежень SQLite це операції запису. Дозволений тільки один процес запису в проміжок часу, що сильно обмежує продуктивність.

1.2.1.2 MySQL

**MySQL - це найпоширеніша повноцінна серверна СУБД. MySQL дуже функціональна, вільно поширювана СУБД, яка успішно працює з різними сайтами і веб додатками. Навчитися використанню цієї СУБД досить просто, так як на просторах інтернету ви легко знайдете більшу кількість інформації.**

Варто зауважити, що завдяки популярності цієї СУБД, існує величезна кількість різних плагінів і розширень, що полегшують роботу з системою.

Незважаючи на те, що в ній не реалізований весь SQL функціонал, MySQL пропонує досить багато інструментів для розробки додатків. Так як це серверна СУБД, додатки для доступу до даних, на відміну від SQLite працюють зі службами MySQL.

**Типи даних MySQL:**

* TINYINT - дуже малі цілочисельні значення
* SMALLINT - малі цілочисельні значення
* MEDIUMINT - середні цілочисельні значення
* INT або INTEGER - стандартні цілочисельні значення
* BIGINT - великі цілочисельні значення
* FLOAT - маленькі значення з плаваючою точкою (точність до одного значення після крапки). Завжди знакові значення
* DOUBLE, BOUBLE PRECISION, REAL - Стандартні значення з плаваючою точкою. завжди знакові
* DECIMAL, NUMERIC - розпаковане значення з плаваючою точкою, завжди знакова.
* DATE - дата
* DATETIME - дата і час в одному значенні
* TIMESTAMP - тимчасова відмітка timestamp
* TIME - час
* YEAR - рік, 2 або 4 числа (4 - по-замовчуванням)
* CHAR - строкове значення фіксованої довжини, праворуч завжди додаються пробіли до вказаної довжини при сортуванні
* VARCHAR - строкове значення змінної довжини
* TINYBLOB, TINYTEXT - значення типу BLOB або TEXT, 255 (2 ^ 8 - 1) символів - максімальния довжина
* BLOB, TEXT - значення типу BLOB або TEXT, 65535 (2 ^ 16 - 1) символів - максімальния довжина
* MEDIUMBLOB, MEDIUMTEXT - значення типу BLOB або TEXT, 16777215 (2 ^ 24 - 1) символів - максімальния довжина
* LONGBLOB, LONGTEXT - значення типу BLOB або TEXT, 4294967296 (2 ^ 32 - 1) символів - максімальния довжина
* ENUM - перерахування
* SET – набір

Переваги MySQL:

* Простота в роботі - встановити MySQL досить просто. Спеціальні програми, наприклад GUI, дозволяє досить легко працювати з БД
* Багатий функціонал - MySQL підтримує більшість функціоналу SQL.
* Безпека - велика кількість функцій забезпечують безпеку, які підтримується за замовчуванням
* Масштабованість - MySQL легко працює з великими обсягами даних і легко масштабується
* Швидкість - спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність.

Недоліки MySQL:

* Відомі обмеження - за задумом в MySQL закладені деякі обмеження функціонала, які іноді необхідні в особливо вимогливих додатках.
* Проблеми з надійністю - через деякі способів обробки даних MySQL (зв'язки, транзакції, аудити) іноді поступається іншим СУБД по надійності.
* Повільна розробка - Хоча MySQL технічно відкрите ПЗ, існують скарги на процес розробки. Варто зауважити, що існують інші досить успішні СУБД створені на базі MySQL, наприклад MariaDB.

Коли слід використовувати MySQL:

* розподілені операції - якщо функціоналу SQLite не вистачає, то варто розглянути MySQL. Так як ця СУБД поєднує в собі просунутий функціонал і вільний доступ до вихідного коду.
* високий рівень безпеки - система безпеки MySQL включає в себе прості і в той же час гідні способи захисту доступу до даних
* Веб сайти і веб додатки - більшість сайтів і онлайн додатків спокійно працюють з MySQL незважаючи на деякі обмеження. Будучи легкої в налаштуванні і масштабованої системою - MySQL перевірена часом.
* Індивідуальні рішення - якщо ви працюєте з яким або специфічним проектом, MySQL легко зможе вам допомогти завдяки широким можливостям у налаштуванні і функціоналом.

Коли краще відмовитися від MySQL:

* Відповідність стандартам -Так як MySQL не ставить для себе метою - повністю соответстввать СТАНДРТ SQL, то ця СУБД неповністю підтримує SQL. Якщо в майбутньому ви плануєте перейти на подібну систему, то MySQL - не кращий вибір.
* Нить - хоча деякі движки БД досить легко виконують паралельне читання, паралельні операції читання-запису можуть створити проблеми
* Нестача функціоналу - деякі движки MySQL, наприклад, не підтримують повнотекстовий пошук.
  + - 1. PostgreSQL

**PostgreSQL є найбільш професійним з усіх трьох розглянутих нами СУБД. Вона вільно розповсюджується і максимально відповідає стандартам SQL. PostgreSQL або Postgres намагаються повністю застосовувати ANSI / ISO SQL стандарти своєчасно з виходом нових версій.**

**Від інших СУБД PostgreSQL відрізняється підтримкою затребуваного об'єктно-орієнтованого та / або реляційного підходу до баз даних. Наприклад, повна підтримка надійних транзакцій, тобто атомарность, послідовність, ізоляційні, міцність (Atomicity, Consistency, Isolation, Durability (ACID).) Завдяки потужним технологіям Postgre дуже продуктивна. Паралельність досягнута не за рахунок блокування операцій читання, а завдяки реалізації управління багатоваріантним паралелізмом (MVCC), що також забезпечує відповідність ACID. PostgreSQL дуже легко розширювати своїми процедурами, які називаються збережені процедури. Ці функції спрощують використання постійно повторюваних операцій.**

**Хоча PostgreSQL і не може похвалитися великою популярністю у відмінності від MySQL, існує досить велика кількість додатків полегшують роботу з PostgreSQL, незважаючи на всю потужність функціоналу. Зараз досить легко встановити цю СУБД використовуючи стандартні менеджери пакетів операційних систем.**

Типи даних PostgreSQL:

* bigint - знакове 8-ми бітове цілочисельне значення
* bigserial - автоматично інкрементіруемое 8-ми бітове цілочисельне значення
* bit [(n)] - рядок постійної довжини
* bit varying [(n)] - рядок змінної довжини
* boolean - булево значення (true / false)
* box - прямокутник на площині
* byte - бінарні дані (масив байтів)
* character varying [(n)] - строкове значення змінної довжини
* character [(n)] - строкове значення постійної довжини
* cidr - IPv4 / IPv6 мережевий адресу
* circle - коло на площині
* date - календарна дата (рік, місяць, день)
* double precision - число з плаваючою точкою подвійної точності (8 байт)
* inet - IPv4 / IPv6 адреса хоста
* integer - знакове 4-ех байтовое цілочисельне значення
* interval [fields] [(p)] - відрізок часу
* line - нескінченна пряма на площині
* lseg - відрізок на площині
* macaddr - MAC адресу
* mony - валютне значення
* numeric [(p, s)] - точне чисельне значення з обраної точністю
* path - геометрична крива на площині
* point - геометрична точка на площині
* polygony - багатокутник на площині
* real - число з плаваючою точкою одинарної точності (4 байта)
* smallint - знакове цілочисельне значення (4 байта)
* serial - автоматичні інкрементіруемое цілочисельне значення (4 байта)
* text - строкове значення змінної довжини
* time [(p)] [without time zone] - час доби (без часового поясу)
* time [(p)] with time zone - час доби (включаючи часовий пояс)
* timestamp [(p)] [without time zone] - дата і час (без часового поясу)
* timestamp [(p)] with time zone - дата і час (з часовим поясом)
* tsquery - текстовий пошуковий запит
* tsvector - документ текстового пошуку
* txid\_snapshot - користувальницький знімок транзакції з ID
* uuid - універсальний унікальний ідентифікатор
* xml - XML дані

Переваги PostgreSQL:

* Відкрите ПО відповідне стандарту SQL - PostgreSQL - безкоштовне ПЗ з відкритим вихідним кодом. Ця СУБД є дуже потужною системою.
* Велике співтовариство - існує досить велика спільнота в якому ви запросто знайдете відповіді на свої запитання
* Велика кількість доповнень - незважаючи на величезну кількість вбудованих функцій, існує дуже багато доповнень, що дозволяють розробляти дані для цієї СУБД і управляти ними.
* Розширення - існує можливість розширення функціоналу за рахунок збереження своїх процедур.
* Об'єктно - PostrgreSQL це не тільки реляційна СУБД, але також і об'єктно-орієнтована з підтримкою успадкування та багато іншого.

Недоліки PostgreSQL:

* Продуктивність - при простих операціях читання PostgreSQL може значно уповільнити сервер і бути повільніше своїх конкурентів, таких як MySQL
* Популярність - за своєю природою, популярністю ця СУБД похвалитися не може, хоча і присутній досить велика спільнота.
* Хостинг - в силу вище перелічених чинників іноді досить складно знайти хостинг з підтримкою цієї СУБД.

Коли використовувати PostgreSQL:

* Цілісність даних - коли надійність і цілісність даних - ваші вимоги, PostgreSQL буде, мабуть, кращим вибором
* Складні користувальницькі процедури - якщо вам необхідно використовувати для користувача процедури, то PostgreSQL має вбудовану підтримку для них
* Інтеграція - якщо в майбутньому ви плануєте перехід на платні СУБД, наприклад Oracle, то зробити це з PostgreSQL буде досить просто в порівнянні з іншими безкоштовними СУБД
* Складна структура даних - порівняно з іншими відкритими СУБД PostgreSQL надає більше можливостей для створення складних структур даних без необхідності жертовать якими або аспектами.

Коли не слід використовувати PostgreSQL:

* Швидкість - якщо швидке читання для вас єдиний фактор, то варто придивитися до інших СУБД
* Проста настройка - якщо вам не потрібна цілісність даних, відповідність ACID або складні структури даних, то настройка PostgreSQL може добряче попсувати вам нерви
* Реплікація - якщо ви не готові витратити час і енергію на те, що міг би з легкістю зробити MySQL, то напевно простіше було б на ньому і залишитися.

1.2.2 NoSQL

До певного моменту, практично єдиною відповіддю на питання «як зберігати й обробляти дані?» була якась реляційна СУБД, але зі збільшенням обсягів з'явилися проблеми, з якими класична реляційна архітектура не справлялася, тому інженерам довелося придумувати нові рішення. Спробуємо уявити ті кроки, які можна зробити, якщо СУБД припиняє справлятися з обсягом виконуваних операцій:

1. Природним першим кроком є ​​спробувати найменш затратні способи. Найпростіший, при наявності фінансів, спосіб це нічого не робити, а просто купити більш потужне обладнання (вертикальне масштабування). Однак нескінченно потужного сервера не існує, а значить вертикальний ріст кінцевий.
2. Більш витратний спосіб - це оптимізувати запити, проаналізувавши плани їх виконання, і створити додаткові індекси. Такий метод може принести тимчасове полегшення, але додаткові індекси породжують додаткові операції, а з ростом обсягів оброблюваних даних ці ​​додаткові операції призводять до деградації.
3. Наступним кроком може бути впровадження каша на читання, при правильній організації такого рішення, ми можемо позбавити СУБД від суттєвої частини операцій читання, але жертвуємо суворої консистентності даних. До того ж, цей підхід призводить до ускладнення клієнтського ПЗ.
4. Вибудовування операцій вставки / оновлення в чергу - непогане рішення, але розмір черги обмежений. До того ж, для забезпечення суворої консистентності, нам доведеться організувати персистентність самої черги, а це непросте завдання.
5. Нарешті, коли всі інші способи перестають працювати, наступає момент переглянути спосіб організації самих даних. В першу чергу - провести денормализация схеми, щоб зменшити число нелокальних звернень.
6. Ну а коли і це не працює, то залишається тільки масштабувати горизонтально, тобто різниця обчислення на різні вузли. Тут доводиться остаточно попрощатися з нормалізацією і зовнішніми ключами, до того ж потрібно відповісти на питання «за якими ознаками розподіляти нові кортежі по вузлах?» І «як справити міграцію існуючої схеми?».  
    Підводячи підсумки, можна зробити висновок, що спроби пристосувати реляційну СУБД до роботи з великими даними призводять до наступного:
7. Відмови від суворої консистентності
8. Відхилень від нормалізації та впровадження надмірності
9. Втрата виразності мови SQL, і необхідності моделювати частину її функцій програмно.
10. Істотне ускладнення клієнтського програмного забезпечення.
11. Складнощі підтримки працездатності та відмовостійкості отриманого рішення.

Необхідно, щоправда, відзначити, що виробники реляційних СУБД усвідомлюють всі ці проблеми і вже почали пропонувати масштабовані кластерні рішення. Однак вартість впровадження та супроводу подібних рішень часто не окупається.

Поглянувши на висновки з попереднього розділу, в голові відразу народжується досить очевидна думка - а чому б не спроектувати архітектуру.

здатну адаптуватися до зростаючих обсягів даних і ефективно їх обробляти? Подібні думки привели до появи руху NoSQL. З разу хочеться звернути увагу, що NoSQL, не має на увазі бездумного відмови від всіх принципів реляційної моделі. Більш того, термін «NoSQL» вперше був використаний в 1908 році для опису реляційної бази даних, що не використала SQL. [3]. Просто теоретики і практики даного підходу справедливо стверджують, що при виборі інструментарію необхідно відштовхуватися від завдання, а реляційні СУБД підхід не завжди, особливо в епоху великих даних. Популярність NoSQL, став набирати в 2009 р у зв'язку з появою великої кількості веб-стартапів, для яких найважливішим завданням є підтримання постійної високої пропускної здатності сховища при необмеженому збільшенні обсягу даних. Розглянемо основні особливості NoSQL підходу [4, 5]:

1. Виняток зайвого ускладнення. Реляційні бази даних виконують величезну кількість різних функцій і забезпечують строгу консистентность даних. Однак для багатьох додатків подібний набір функцій, а також задоволення вимогам ACID є зайвими.
2. Висока пропускна здатність. Багато NoSQL рішення забезпечують набагато вищу пропускну здатність даних ніж традиційні СУБД. Наприклад, колоночного сховище HyperTable, що реалізує підхід Google Bigtable, дозволяє пошуковому движку Zvent зберігати близько мільярда записів на день. Як інший приклад можна привести саму Bigtable, здатну обробити 20 петабайт інформації в день [6].
3. Необмежена горизонтальне масштабування. На противагу реляційним СУБД. NoSQL рішення проектуються для необмеженого горизонтального масштабування. при цьому додавання і видалення вузлів в кластері ніяк не позначається на працездатності системи. додатковою перевагою подібної архітектури є те, що NoSQL кластер може бути розгорнуть на звичайному апаратному забезпеченні, істотно знижуючи вартість всієї системи.
4. Консистентне в жертву продуктивності. при описі підходу NoSQL не можна не згадати теорему САР. дотримуючись цієї теоремі, багато NoSQL бази даних реалізують доступність даних (availability) і стійкість до розділення (partition tolerance), жертвуючи консистентністю на догоду високої продуктивності. І дійсно, для багатьох класів додатків сувора консистентність даних-це те, від чого цілком можна відмовитися.

1.2.2.1 Класифікація NoSQL сховищ

На сьогоднішній день створено велику кількість NoSQL рішень. Всі вони грунтуються на чотирьох принципах з попереднього розділу, але можуть досить сильно відрізнятися один від одного. Багато теоретики і практики створювали свої власні класифікації, але найбільш простий і загальновживаною можна вважати систему, засновану на використовуваної моделі даних, запропоновану Ріком Кейтелем (Rick Cattel)[7]:

1. Сховища ключ-значення. Відмінною особливістю є проста модель даних - асоціативний масив або словник, що дозволяє працювати з даними по ключу. Основне завдання подібних сховищ - максимальна продуктивність, тому ніяка інформації про структуру значень не зберігається
2. Документні сховища. Модель даних подібних сховищ дозволяє об'єднувати безліч пар ключ-значення в абстракцію, звану «документ». Документи можуть мати вкладену структуру і об'єднуватися в колекції. Однак це скоріше зручний спосіб логічного об'єднання, оскільки ніякої жорсткої схеми у документів немає і безлічі пар ключ-значення, навіть в рамках однієї колекції, можуть бути абсолютно довільними. Робота з документами проводиться по ключу, однак існують рішення, що дозволяють здійснювати запити за значеннями атрибутів.
3. Колоночні сховища. Цей тип здається найбільш схожим з традиційними реляційними СУБД. Модель даних сховищ подібного типу має на увазі зберігання значень як інтерпретованих байтових масивів, адресованих кортежами (ключ рядки, ключ шпальти, мітка часу) [6]. Основою моделі даних є колонка, число колонок для однієї таблиці може бути необмеженим. Колонки по ключам об'єднуються в сімейства, що володіють певним набором властивостей.
4. Сховища на графах. Подібні сховища застосовуються для роботи з даними, які природним чином представляються графами (наприклад, соціальна мережа). Модель даних складається з вершин, ребер і властивостей. Робота з даними здійснюється шляхом обходу графа по ребрах з заданими властивостями.

|  |  |
| --- | --- |
| Тип | Примеры храннлища |
| Сховища ключ-значення | Redis  Scalaris  Riak  Tokyo Turant |
| Документні сховища | SimpleDB  CouchDB  MongoDB |
| Колоночні сховища | BigTable  Hbase  Cassandra |
| Сховища на графах | Neo4j |

Таблиця 1 - Класифікація NoSQL сховищ по моделі даних

1.3. MapReduce

Піонером в області великих даних можна вважати компанію Google, яка в 2003 р описала розподілену файлову систему GFS [8], а в 2004 р представила світу обчислювальну модель MapReduce [9]. Саме ці публікації допомогли розробникам вільного пошукового движка Apache Nutch створити проект Hadoop [10] |, який сьогодні фактично став синонімом терміна «великі дані».

1.3.1 Hadoop

Перед тим як розглянути Hadoop уважніше, слід відповісти на питання: «а навіщо потрібен ще один продукт, якщо багато NoSQL бази даних надають інтерфейси для MapReduce обчислень?» [11, 12, 13]. Відповідь можна отримати, розглянувши продуктивність сучасних жорстких дисків. Середня продуктивність жорсткого диска сьогодні ~ 100 МБ \* с. що означає можливість прочитати 1 ТБ інформації приблизно за 2.5 години. Поліпшити такі гнітючі показники можна паралельним читанням з декількох дисків. Наприклад, той же самий 1 ТБ можна прочитати зі 100 дисків за 2 хвилини. Але ж майже всі NoSQL рішення підтримують горизонтальне масштабування, а отже і паралельні дискові операції? І тут ключовим чинником стає час позиціонування головки. Для того, щоб операції оновлення і читання були ефективними.  
 NoSQL базам (CouchDB, MongoDB) доводиться використовувати структури з довільним доступом, наприклад В-дерева [14]. А значить, якщо відмовитися від довільного оновлення даних і обробляти весь набір послідовно, можна домогтися серйозного приросту продуктивності.  
 Саме цей принцип і покладений в основу архітектури Hadoop. За зберігання н організацію даних в Hadoop кластері відповідає розподілена файлова система HDFS [15]. При проектуванні якої використовувалися такі принципи:

1. Апаратні збої неминучі. Тому HDFS реалізує надійні алгоритми реплікації даних, а для метаданих файлової системи підтримується журнал, що дозволяє відновити необхідну стан.
2. Потокова обробка та великі обсяги. HDFS влаштована таким чином, щоб забезпечити максимальну продуктивність потокового доступу до даних. До того ж структури файлової системи оптимізовані для роботи з великими файлами.
3. Локальність даних. Набагато ефективніше виконувати обчислення рядом з даними. HDFS надає додаткам програмний інтерфейс, який дозволяє виконувати обчислення ближче до необхідним даними, скорочуючи пересилки між вузлами кластера.

Обчислення в Hadoop представляються у вигляді послідовності map і reduce задач. На початку обчислень вхідний безліч даних розбивається на кілька підмножин. Кожна підмножина обробляється на окремому вузлі кластера. Map задача на кожному вузлі отримує на вхід безліч пар ключ-значення і повертає іншу множину map. Далі всі пари групуються по ключу, сортуються і подаються на вхід reduce задачі, яка формує фінальний результат або вхід для іншої map задачі [1].

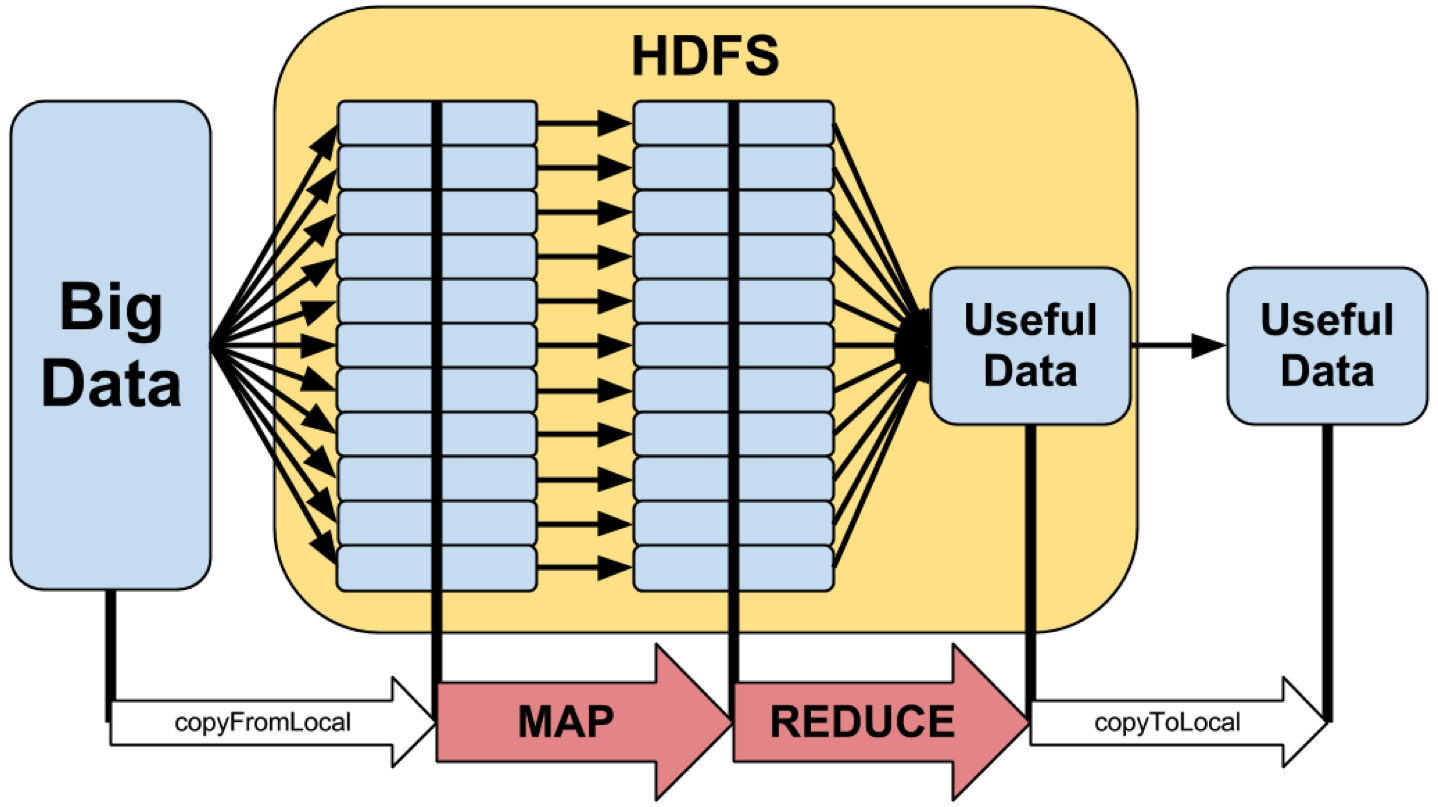


Рисунок 1 - Схема MapReduce обчислень.

Одним із цікавих прикладів ефективності Hadoop є тест швидкості сортування [16], метою якого є сортування записів по 100 байт - 10 байт на ключ і 90 байт на значення. У листопаді 2008 р рекордом в сортуванні 1 ТБ володів кластер компанії Google - 68 с. А в травні 2009 р вийшов звіт Yahoo! в якому стверджувалося, що їх Hadoop кластер відсортував тестовий набір в 1ТБ за 62с! [17].

Навколо Наdоор сформувалася ціла екосистема проектів:

1. Hive - розподілене сховище даних. Hive управляє даними в HDFS і надає мова запитів HiveQL, заснований на SQL. Запити HiveQL автоматично транслюються в MapReduce завдання [18].
2. Pig - середовище виконання і високорівнева мова, для опису обчислень в Наdоор. Програми Pig також транслюються в MapReduce завдання [19].
3. Hbase - розподілена колоночна база даних. Hbase використовує HDFS, як сховище і підтримує як пакетні обчислення, так і довільний доступ [20].
4. ZooKeeper - високодоступний координаційний сервіс, що використовується для побудови розподілених додатків [21].

1.3.2 Disco

Управління процесами в розподіленої системі - складне завдання. А найскладніше впоратися з частковими відмовами, коли помилки в окремих процесах не повинні впливати на обчислення в загальному. MapReduce позбавляє розробника від необхідності думати про помилки, потрібно тільки лише забезпечити код двох функцій: map і reduce. Про інше подбає реалізація, автоматично визначивши невдало завершені завдання і перезапустивши їх. Така можливість виникає, тому що завдання незалежні, бо не мають поділюваних ресурсів [22]. Подібна особливість робить модель акторів [23] ідеальною для реалізації MapReduce фреймворка.

Цим і скористалися розробники дослідного Центру Nokia, створивши проект Disco [24]. Особливістю Disco є те, що ядро системи розроблено на функціональній мові Erlang [25], який здобув славу інструменту, відмінно підходящого для програмування розподілених обчислень на основі моделі акторов. Компанія Ericson, що використовує мову для програмування комутаційних вузлів своєї телефонної мережі, навіть заявила про досягнення показника відмовостійкості обладнання в 99.9999999% [26].

Наdоор, Disco і подібні їм проекти відмінно виконують задачу розподіленої пакетної обробки великих обсягів даних. Фокус на пакетній обробці, зокрема, призводить до того, що обчислення відбуваються з великою затримкою. Подібні затримки можуть бути неприйнятні для цілого класу задач, де відповіді на питання потрібно отримувати негайно.

1.4. Обробка потоків подій в реальному часі

Кілька років з моменту першого публічного релізу Hadoop, пакетна обробка була, мабуть, єдиним способом аналізу великих даних. Однак розвиток таких додатків як пошукові системи реального часу, високочастотна торгівля і соціальні мережі диктувало необхідність миттєво реагувати на нову інформацію. Індустрії все більше не вистачало «Hadoop реального часу». Цю нестачу компенсував фреймворк Ѕ4 (Simple Scalable Streaming System) від компанії Yahoo!

1.4.1 Yahoo! Ѕ4

Розробники Ѕ4 писали, що з самого початку розглядали можливість адаптувати Hadoop для обчислень на необмежених потоках подій в реальному часі, але архітектура Hadoop оптимізована для пакетної обробки статичних даних, що робить створення універсальної системи занадто трудомістким і складним процесом [27].

Проектована система повинна була задовольняти наступним вимогам:

1. Надавати простий програмний інтерфейс потокової обробки даних.
2. Забезпечувати високу доступність кластера.
3. Мінімізувати затримки, використовуючи тільки оперативну пам'ять вузлів кластера.
4. Архітектура повинна бути децентралізованою і симетричною. Всі вузли ідентичні, немає єдиної точки відмови.

Для спрощення первісної реалізації були введені наступні допущення:

1. Відновлення від помилок може призводити до втрати стану процесів на даному вузлі

2. Динамічне додавання і видалення вузлів в кластер неможливо.

Важливо відзначити, що перше допущення так і залишалося в реалізаціях Ѕ4, аж до версії вийшла в серпні 2012 р Додавання механізму контрольних точок дозволило частково вирішити проблему втрати поточного

стану.

Обчислення в Ѕ4 природним чином представляються графом, вершинами якого є обчислювальні елементи (РЕ), ​​а ребрами - потоки подій. Подія представляє з себе кортеж іменованих значень.

Іменування є важливим для реалізації групування потоків. Так як архітектура Ѕ4 увазі зберігання стану в пам'яті РЕ, то часто буває важливо спрямовувати кортежі, що задовольняють певним критеріями, на задані вузли, досягається це об'єднанням потоків по іменам значень в кортежах. Обчислювальний елемент, отримуючи на вході події з одного або декількох потоків, може або створити новий вихідний потік або опублікувати результат. Ще одним примітивом S4, який, однак, не є елементом графа, є адаптер. адаптери виробляють перетворення зовнішніх вхідних / вихідних потоків в необхідний формат.

Про продуктивність Ѕ4 можна судити за результатами, опублікованими авторами в [27]. В якості експерименту наводиться додаток провадить розрахунок CTR різних блоків на сторінці пошукової видачі.

Метою програми є визначення рекламних блоків з низьким CTR і виключення їх подальшого показу. У роботі стверджується, що

продуктивність кластера з 16 чотириядерних серверів з 2 ГБ пам'яті

почала падати при досягненні пропускної здатності в 7268 подій / с або 9.7 Мбтгт / с.

1.4 Висновок

У цій роботі були розглянуті проблеми, які поставили перед реляційними СУБД великі дані. Проаналізувавши можливі шляхи вирішення цих проблем, ми вказали на ті концептуальні обмеження, які не дозволяють класичної реляційної архітектурі справлятися зі стрімко зростаючим обсягом інформації. Далі були розглянуті три підходи до роботи з великими даними: NoSQL, MapReduce і обробка потоків подій в реальному часі. Ми звернули увагу на ті архітектурні особливості, які дозволяють кожному з них ефективно вирішувати поставлену задачу. Важливо відзначити, що жоден з представлених підходів не пропонує вирішення всіх можливих завдань, які виникли в контексті великих даних.

* 1. Вибір напрямку та формулювання задач досліджень

Основною проблемою у створенні систем управління портфелем юридичних процесів є мінімізація об'єму дискового простору для зберігання різнотипних даних. Практично в кожній організації постійно зростає обсяг оброблюваних і збережуваних даних. На тлі підвищення вимог ІТ та бізнесу до безпеки, швидкодії та гнучкості таке зростання може привести до появи значних проблем.

Наразі існуючі рішення для збереження нетипізованих даних не забезпечують можливість використання всіх переваг ефективної інфраструктури зберігання.

Тому, враховуючи постійних ріст потреби у швидкодіючих системах для організації документообігу на підприємствах є напрочуд актуальною і така система може мати великий попит у юридичних компаній. Таким чином, для досягнення поставленої мети магістерської дисертації необхідно виконати наступні задачі:

* + вдосконалити методи збереження нетипізованих даних в інформаційних системах;
  + впровадити розроблену інформаційну технологію в інформаційній системі управління портфелем юридичних процесів;

2 ВИБІР ТЕХНІЧНХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір мови програмування

В якості мови програмування для розробки серверної частини було обрано PHP.

Головним фактором мови РНР є практичність. РНР повинен надати програмісту засоби для швидкого та ефективного вирішення поставлених задач. Практичний характер РНР обумовлений п'ятьма важливими характеристиками:

* + - традиційністю;
    - простотою;
    - ефективністю;
    - безпекою;
    - гнучкістю.

Існує ще одна «характеристика», яка робить РНР особливо привабливим: він розповсюджується безкоштовно! Причому, з відкритими вихідними кодами (Open Source).

Традиційність

Мова РНР здаватиметься знайомим програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з Сі, Perl.

Код РНР дуже схожий на той, який зустрічається в типових програмах на С або Pascal. Це помітно знижує початкові зусилля при вивченні РНР. PHP - мова, що поєднує гідності Perl і Сі і спеціально націлений на роботу в Інтернеті, мова з універсальним (правда, за деякими застереженнями) і ясним синтаксисом.

І хоча PHP є досить молодим мовою, він знайшов таку популярність серед web-програмістів, що на даний момент є чи не найпопулярнішою мовою для створення web-додатків (скриптів).

Простота

Сценарій РНР може складатися з 10 000 рядків або з одного рядка - все залежить від специфіки вашого завдання. Вам не доведеться довантажувати бібліотеки, вказувати спеціальні параметри компіляції або що-небудь в цьому роді. Механізм РНР просто починає виконувати код після першої екрануючої послідовності (<?) І продовжує виконання до того моменту, коли він зустріне парну екрануюче послідовність (?>). Якщо код має правильний синтаксис, він виконується в точності так, як вказав програміст.

PHP - мова, яка може бути вбудований безпосередньо в html-код сторінок, які, в свою чергу будуть коректно оброблятися PHP -інтерпретатором. Ми можемо використовувати PHP для написання CGI-сценаріїв і позбутися від безлічі незручних операторів виведення тексту. Ми можемо залучати PHP для формування HTML-документів, позбувшись від безлічі викликів зовнішніх сценаріїв.

Велика розмаїтість функцій PHP позбавлять вас від написання багаторядкових призначених для користувача функцій на C або Pascal.

Ефективність

Ефективність є виключно важливим фактором при програмуванні для багатокористувацьких середовищ, до числа яких належить і web.

Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслюючою інтерпретатором. Такий пристрій «движка» PHP дозволяє обробляти сценарії з достатньо високою швидкістю.

За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше за аналогічні їм програм, написаних на Perl. Однак, щоб не робили розробники PHP, відкомпілювалися виконувані файли будуть працювати значно швидше - в десятки, а іноді і в сотні разів. Але продуктивність PHP цілком достатня для створення цілком серйозних web-додатків. Детально про устрій і характеристиках «движка» PHP можна ознайомитися тут.

Безпека

РНР надає в розпорядження розробників і адміністраторів гнучкі та ефективні засоби безпеки, які умовно діляться на дві категорії: засоби системного рівня і засоби рівня додатки.

1. Засоби безпеки системного рівня

У РНР реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні РНР це забезпечує максимальну свободу дій і безпеку. РНР може працювати в так званому безпечному режимі (safe mode), який обмежує можливості застосування РНР користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання і використання пам'яті (неконтрольований витрата пам'яті негативно впливає на швидкодію сервера). За аналогією з cgi-bin адміністратор також може встановлювати обмеження на каталоги, в яких користувач може переглядати і виконувати сценарії РНР, а також використовувати сценарії РНР для перегляду конфіденційної інформації на сервері (наприклад, файлу passwd).

2. Засоби безпеки рівня додатки

У стандартний набір функцій РНР входить ряд надійних механізмів шифрування. РНР також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції (e-commerce). Інша перевага полягає в тому, що вихідний текст сценаріїв РНР не можна переглянути в браузері, оскільки сценарій компілюється до його відправлення за запитом користувача. Реалізація РНР на стороні сервера запобігає викрадення нетривіальних сценаріїв користувачами, знань яких вистачає хоча б для виконання команди View Source.

Детально про безпеку PHP можна ознайомитися тут

Гнучкість

Оскільки РНР є вбудовуваним (embedded) мовою, він відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча РНР зазвичай рекомендується використовувати в поєднанні з HTML, він з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані додатки РНР легко розширюються в міру необхідності (втім, це відноситься до всіх основних мов програмування).

Немає проблем і з залежністю від браузерів, оскільки перед відправкою клієнту сценарії РНР повністю компілюються на стороні сервера. В сутності, сценарії РНР можуть передаватися будь-яким пристроям з браузерами, включаючи стільникові телефони, електронні записні книжки, пейджери і портативні комп'ютери, не кажучи вже про традиційні ПК. Програмісти, що займаються допоміжними утилітами, можуть запускати РНР в режимі командного рядка.

Оскільки РНР не містить коду, орієнтованого на конкретний web-сервер, користувачі не обмежуються певними серверами (можливо, незнайомими для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold і Zeus - РНР працює на всіх перерахованих серверах. Оскільки ці сервери працюють на різних платформах, РНР в цілому є платформенно-незалежним мовою й існує на таких платформах, як UNIX, Solaris, FreeBSD і Windows 95/98 / NT / 2000 / XP / 2003.

Нарешті, засоби РНР дозволяють програмісту працювати з зовнішніми компонентами, такими як Enterprise Java Beans або СОМ-об'єкти Win32. Завдяки цим новим можливостям РНР займає гідне місце серед сучасних технологій і забезпечує масштабування проектів до необхідних меж.

безкоштовне розповсюдження

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, зробило безсумнівно благотворний вплив на багато проектів, в першу чергу - Linux, хоча і успіх проекту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення РНР, оскільки підтримка користувачів зі всього світу виявилася дуже важливим чинником в розвитку проекту РНР.

Прийняття стратегії Open Source і безкоштовне розповсюдження початкових текстів РНР надало неоціниму послугу користувачам. Вдобавок, чуйне співтовариство користувачів РНР є свого роду «колективної службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання.

2.2 Вибір СУБД

Оскільки в розділі 1 було прийнято рішення використовувати реляційну модель даних, тому розглянемо реляційні СУБД. Серед доступних варіантів на ринку варто відзначити Oracle Database, IBM DB2, Microsoft SQL Server, MySQL, PostgreSQL. Серед перерахованих СУБД ліцензію на вільне програмне забезпечення мають лише MySQL та PostgreSQL. MySQL призначена в основному для малих і середніх програм. В таких програмах MySQL може мати більшу швидкість обробки, зберігання і вибірки даних порівняно з PostgreSQL. Але коли мова іде про великі об'єми даних і високонавантажені системи — краще використовувати PostgreSQL. Крім того PostgreSQL, на відміну від MySQL, повністю відповідає стандарту ACID (Atomicity, Consistency, Isolation, Durability) - набір властивостей, що гарантують надійну роботу транзакцій бази даних: атомарність, узгодженість, ізольованість та довговічність; забезпечує повну підтримку вторинних ключів; має вищу продуктивність підзапитів; повне хешування індексів і повнотекстовий пошук; автоматичне відновлення після збоїв. Отже в якості СУБД оберемо PostgreSQL.

PostgreSQL — об'єктно-реляційна система керування базами даних. Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації.

2.2.1 Основні можливості PostgreSQL

Функції: Функції дозволяють виконувати деякий код безпосередньо сервером бази даних. Ці функції можуть бути написані на SQL, який має деякі примітивні програмні оператори, такі як галуження та цикли. Але гнучкішою буде функція написана на одній із багатьох мов програмування, з якими PostgreSQL може працювати. До таких мов належать:

* Вбудована мова, яка зветься PL/pgSQL, подібна до процедурної мови PL/SQL компанії Oracle.
* Мови розробки сценаріїв: PL/Perl, PL/Python, PL/Tcl, PL/Ruby, PL/sh.
* Класичні мови програмування C, C++, Java (за допомогою PL/Java).

Функції можуть бути написані, а потім виконуватись із привілеями користувача, який її викликав, або із привілеями користувача, який її написав.

Індекси: В PostgreSQL є підтримка індексів наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створити нові типи індексів.

Багатоверсійність (MVCC): PostgreSQL підтримує одночасну модифікацію БД декількома користувачами за допомогою механізму Multiversion Concurrency Control (MVCC). Завдяки цьому виконуються вимоги ACID, і практично відпадає потрібність в блокуванні зчитування.

Типи даних: PostgreSQL підтримує великий набір вбудованих типів даних:

* Числові типи:
* Цілі
* З фіксованою крапкою
* З нефіксованою крапкою
* Грошовий тип
* Символьні типи довільної довжини
* Двійкові типи (включаючи BLOB)
* Типи «дата/час»
* Булевий тип
* Перерахування
* Геометричні примітиви
* Мережеві типи
* IP і IPv6-адреси
* CIDR-формат
* МАС-адреса
* UUID-ідентификатор
* XML-дані
* JSON-дані
* Масиви
* OID-типи
* Псевдотипи

Крім того, користувач може самостійно створювати нові необхідні йому типи та програмувати для них механізми індексування за допомогою GiST.

Об'єкти користувача: PostgreSQL може бути розширено користувачем для власних потреб практично в будь-якому аспекті. Є можливість додавати власні:

* Перетворення типів
* Типи даних
* Домени (для користувача типи з самого початку з накладеними обмеженнями)
* Функції (включаючи агрегатні)
* Індекси
* Оператори (включаючи перевизначення вже існуючих)
* Процедурні мови

Наслідування: Таблиці можуть успадковувати характеристики та набори полів від інших таблиць (батьківських). При цьому дані, які додаються до породженої таблиці, автоматично будуть брати участь (якщо це не вказано окремо) в запитах до батьківської таблиці. Цей функціонал в поточний час не є повністю завершеним. Однак він достатній для практичного використання.

Тригери: Тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, перевіряючий доданий запис на відповідності певним умовам. При написанні функцій для тригерів можуть використовуватися різні мови програмування. Тригери асоціюються з таблицями. Множинні тригери виконуються в алфавітному порядку.

Інші можливості:

* Дотримання принципів ACID
* Відповідність стандартам ANSI SQL-92 і SQL-99
* Підтримка запитів з OUTER JOIN, UNION, UNION ALL, EXCEPT і підзапитів
* Послідовності
* Контроль цілісності
* Реплікація
* Загальні табличні вирази й рекурсивні запити
* Аналітичні функції
* Підтримка Unicode (UTF-8)
* Підтримка регулярних виразів у стилі Perl
* Вбудована підтримка SSL і Kerberos
* Протокол поділюваних блокувань
* Завантажувані розширення, підтримують SHA1, MD5, XML і іншу функціональність (API відкритий)
* Засоби для генерації сумісного з іншими системами SQL-коду та імпорту з інших систем

PostgreSQL базується на мові SQL і підтримує багато з можливостей стандарту SQL: 2011.

На даний момент (версія 9.3), в PostgreSQL є наступні обмеження:

Таблиця 3. ⎯ Характеристики PostgreSQL

|  |  |
| --- | --- |
| Максимальний розмір бази даних | немає обмежень |
| Максимальний розмір таблиці | 32 Тбайт |
| Максимальний розмір запису | 1,6 Тбайт |
| Максимальний розмір поля | 1 Гбайт |
| Максимум записів в таблиці | немає обмежень |
| Максимум полів у записі | 250—1600, в залежності від типів полів |
| Максимум індексів в таблиці | немає обмежень |

Сильними сторонами PostgreSQL вважаються:

* підтримка БД практично необмеженого розміру;
* потужні і надійні механізми транзакцій і реплікації;
* розширювана система вбудованих мов програмування: в стандартній поставці підтримуються PL / pgSQL, PL / Perl, PL / Python і PL / Tcl; додатково можна використовувати PL / Java, PL / PHP, PL / Py, PL / R, PL / Ruby, PL / Scheme, PL / sh і PL / V8, а також є підтримка завантаження C-сумісних модулів [7];
* спадкування;
* легка розширюваність.

2.3 Вибір веб сервера

Apache HTTP-сервер (є скороченням від англомовного a patchy server, "сервер з латками", крім того, тут є алюзія і до індіанського племені апачів. Серед російських користувачів загальноприйнято спотворене вимова апач) - це повнофункціональний, розширюваний веб-сервер, що повністю підтримує протокол HTTP / 1.1, з відкритим кодом.

Сервер може працювати практично на всіх поширених платформах. Існують готові виконувані файли сервера для Windows NT, Windows 9x, OS / 2, Netware 5.x і декількох UNIX-систем. При цьому він дуже простий в установці і конфігурації.

Власне, саме гнучкість конфігурації, а також його надійність, вважаються основними достоїнствами Apache-сервера. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СУБД для аутентифікації користувачів, модифікувати повідомлення про помилки і т. Д. Підтримує IPv6.

Apache налаштовується за допомогою текстових конфігураційних файлів. Основні параметри вже налаштовані «за замовчуванням» і працюватимуть в більшості випадків. Якщо ж функціональності штатного «Апача» не вистачає, то можливо використовувати різноманітні модулі, написані Apache Group і сторонніми розробниками. Важливою перевагою є те, що творці активно спілкуються з користувачами і реагують на всі повідомлення про помилки.

Найпростіша функція, яку може виконувати Apache - стояти на сервері і обслуговувати звичайний HTML-сайт. При отриманні запиту на певну сторінку сервер відправляє в її відповідь браузеру. В якості запиту виступає адреса, набраний в адресному рядку браузера.

Для поділу дизайну і функціональної частини сайту, а також для спрощення зміни статичних об'єктів існує технологія SSI. Вона дозволяє вам помістити всю повторювану інформацію в один файл (наприклад, top.inc), а потім вставляти в сторінки посилання на неї. Потім, якщо знадобиться змінити кілька рядків у цій інформації, то доведеться поміняти їх тільки в одному файлі. Сервер Apache підтримує цю технологію і дозволяє використовувати серверні включення в повному обсязі.

Функції Web-сервера виконує не сам комп'ютер, а програма встановлена ​​на ньому, то є. Коли браузер користувача підключається до Web-серверу і посилає заголовок GET (це прохання передати файл), його запит обробляє саме Apache. Apache перевіряє, чи існує вказаний в заголовку GET файл, і, якщо є, відправляє його разом з заголовками браузеру.

Apache - це свого роду стандарт Web-сервера в Інтернеті. Його основним конкурентом є IIS (Internet Information Server) від Microsoft, який працює під Windows. Apache, хоча й існують його версії під Windows, переважно встановлюється на Unix-подібних операційних системах - Lin ux і FreeBSD. Потрібно також зауважити, що більшість хостерів використовують саме Apache, a НЕ IIS. IIS зазвичай встановлюється на корпоративних серверах, що працюють під Windows.

Сервер Apache був одним з перших серверів, які почали підтримувати віртуальні сервера (хости). Ця можливість дозволяє розміщувати на одному фізичному сервері кілька повноцінних сайтів. У кожного з них може бути свій домен, адміністратор, IP-адресу і так далі.

В Apache існує підтримка технології CGI і PHP, а також можливість підключати інші мови. Це значно спрощує роботу з динамічними Інтернет-сторінками (якими, по суті, сьогодні є всі веб-сторінки).

На даний момент сервер Apache зараз встановлений на 67% серверів по всьому світу.

Ядро Apache включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів.

Теоретично, ядро apache може функціонувати в чистому вигляді, без використання модулів. Однак функціональність такого рішення вкрай обмежена.

Ядро Apache повністю написано на мові програмування C.

Система конфігурації Apache заснована на текстових конфігураційних файлах. Має три умовних рівня конфігурації:

* + - Конфігурація сервера (httpd.conf).
    - Конфігурація віртуального хоста (httpd.conf c версії 2.2, extra / httpd-vhosts.conf).
    - Конфігурація рівня директорії (.htaccess).

Має власну мову конфігураційних файлів, заснований на блоках директив. Практично всі параметри ядра можуть бути змінені через конфігураційні файли, аж до управління MPM. Велика частина модулів має власні параметри.

Частина модулів використовує в своїй роботі конфігураційні файли операційної системи (наприклад / etc / passwd і / etc / hosts).

Крім цього, параметри можуть бути задані через ключі командного рядка.

Крім того, для веб-сервера Apache існує безліч моделей симетричною мультипроцессорности.

Також Apache HTTP Server підтримує модульність. Існує більше 500 модулів, що виконують різні функції. Незважаючи на те, що частина з них розробляється непорсдетсвенно командою Apache Software Foundation, більшість існуючих модулів розроблено сторонніми open source-розробниками.

Модулі можуть бути як включені до складу сервера в момент компіляції, так і завантажені динамічно, через директиви конфігураційного файлу.

За допомогою модулів можна реалізувати наступне:

* 1. Розширення підтримуваних мов програмування
  2. Додавання додаткових функцій чи модифікація основних.
  3. Виправлення помилок
  4. Посилення безпеки.

Частина веб-додатків, наприклад панелі управління ISPmanager і VDSmanager реалізовані у вигляді модуля Apache.

Сервер Apache оснащений вбудованим механізмом віртуальних хостів. Завдяки цьому можливо повноцінно обслуговувати на одному IP-адресі безліч сайтів (доменних імен), відображаючи для кожного з них власне вміст.

Для кожного віртуального хоста можна вказати власні настройки ядра і модулів, обмежити доступ до всього сайту або окремих файлів. Деякі MPM, наприклад Apache-ITK дозволяють запускати процес httpd для кожного віртуального хоста з окремими ідентифікаторами uid і guid.

Також існують модулі, що дозволяють враховувати і обмежувати ресурси сервера (CPU, RAM, трафік) для кожного віртуального хоста.

Інтеграція з іншим ПО і мовами програмування

Для успішної інтеграції сервера з різним програмним забезпеченням, а також з мовами програмування існують додаткові модулі:

* + - PHP (mod\_php).
    - Python (mod python, mod wsgi).
    - Ruby (apache-ruby).
    - Perl (mod perl).
    - ASP (apache-asp).
    - Tcl (rivet)

Крім того, Apache підтримує механізми CGI і FastCGI, що дозволяє виконувати програми на практично всіх мовах програмування, в тому числі C, C ++, Lua, sh, Java.

забезпечення безпеки

Забезпечення безпеки в Apache здійснюється за допомогою різних механізмів, які, крім усього іншого, розмежовують доступ до даних.

Основними є:

* + - Обмеження доступу до певних директоріям або файлам.
    - Механізм авторизації користувачів для доступу до директорії на основі HTTP-аутентифікації (mod\_auth\_basic) і digest-аутентифікації (mod\_auth\_digest).
    - Обмеження доступу до певних директоріям або всьому серверу, засноване на IP-адресах користувачів.
    - Заборона доступу до певних типів файлів для всіх або частини користувачів, наприклад заборона доступу до конфігураційним файлам і файлам баз даних.
    - Існують модулі, що реалізують авторизацію через СУБД або PAM.

В деяких MPM-модулях присутня можливість запуску кожного процесу Apache використовуючи різні uid і gid з відповідними цим користувачам і групам користувачів.

Також, існує механізм suexec, використовуваний для запуску скриптів і CGI-додатків з правами і ідентифікаційними даними користувача.

Для реалізації шифрування даних, що передаються між клієнтом і сервером використовується механізм SSL, реалізований через бібліотеку OpenSSL. Для посвідчення справжності веб-сервера використовуються сертифікати X.509.

Існують зовнішні засоби забезпечення безпеки, наприклад mod\_security.

мови

Можливість визначення сервером місцеположення користувача з'явилася з версії 2.0. З цього моменту також всі сервісні повідомлення, а також повідомлення про помилки та події, відтворюються на кількох мовах з використанням SSI-технології.

Крім того, існує можливість реалізувати засобами сервера відображення різних сторінок для користувачів з різними Локаль. Apache підтримує безліч кодувань, у тому числі Unicode, що дозволяє використовувати сторінки, створені в будь-яких кодуваннях і на будь-яких мовах.

Обробка подій

Адміністратор може встановити власні сторінки і обробники для всіх HTTP помилок і подій, таких як 404 (Not Found) або 403 (Forbidden). У тому числі існує можливість запуску скриптів і відображення повідомлень на різних мовах.

Server Side Includes

У версіях 1.3 і старше був реалізований механізм Server Side Includes, що дозволяє динамічно формувати HTML-документи на стороні сервера.

Управлінням SSI займається модуль mod\_include, включений в базову поставку Apache.

Apache vs IIS

Спори про вибір Apache або IIS є настільки ж давніми, як і суперечки про вибір ОС - Linux або Windows. Тому, перш ніж зробити остаточний вибір, варто адекватно оцінити переваги і недоліки і того, й іншого.

Основні переваги Apache і серверних систем LAMP:

* 1. низькі витрати, так як немає необхідності купувати ліцензії на програмне забезпечення;
  2. гнучке програмування завдяки відкритому вихідному коду;
  3. поліпшена безпека, так як Apache був розроблений для операційної системи не-Windows (а більшість шкідливих програм написано під ОС від Microsoft), то він завжди мав репутацію більш безпечної системи, ніж IIS виробництва Microsoft.

Переваги Internet Information Services (IIS):

1. Windows і IIS підтримуються корпорацією Microsoft, тоді як Apache підтримується тільки призначеним для користувача спільнотою;
2. IIS підтримує платформу.NET Microsoft і сценарії ASPX.
3. модулі дозволяють включити потокову передачу аудіо-і відеоконтенту.

Отже, порівнявши переваги двох даних серверів, можна зробити кілька висновків. По-перше, якщо при виборі сервера вартість ліцензій є головним критерієм, то однозначно варто вибирати комбінацію LAMP, оскільки вона позбавлена ​​від витрат на ліцензування. По-друге, якщо виходити з критерію безпеки, то Apache СНВО лідирує - його система кілька більш ефективна. По-третє, IIS працює тільки на ОС Windows і будь-які варіації у виборі операційної системи знову приведуть до Apache.

Єдиний же варіант, при якому зв'язка IIS на платформі Windows стане найкращим вибором, це максимально зручні для користувача засобу адміністрування як головний критерій. Причому, що виконується сценарій повинен покладатися виключно на ASPX.

Втім, ніхто не заважає розробити готове рішення, яке буде підтримувати роботу на обох серверах.

2.4 Вибір систем побудови звітів

2.4.1 BusinessObjects

BusinessObjects – засіб збору й аналізу ділової інформації, що дозволяє користувачам приймати ефективні, компетентні рішення на основі достовірних даних. Використовуючи ці рішення, користувачі всіх відділів компанії отримують можливість переглядати, форматувати, аналізувати і передавати інформацію в масштабі всієї організації. [8]

Основні можливості:

* розвинені інструменти побудови звітності;
* засоби розширеного аналізу;
* наочність представлення даних ( візуалізація );
* єдиний портал для доступу до інформації;
* засоби пошуку та навігації.

SAP Business Objects забезпечує всебічну підтримку бізнесу. Платформа SAP Business Objects володіє можливостями з'єднання з усіма важливими джерелами даних і забезпечує інтеграцію інформації в рамках різних користувальницьких інтерфейсів. Лінійка продуктів SAP Business Objects включає в себе різні за функціоналом, вартості володіння та впровадженням програмні продукти, які підійдуть для досягнення різних бізнес-цілей.



Рисунок 2.4.1. – SAP BusinessObjects: портфель рішень

SAP BusinessObjects – це Business intelligence аналітика для розширеного аналізу і представлення даних в максимально наочному вигляді, а також для відстеження найважливішої інформації про стан бізнесу в режимі реального часу. Основні користувачі BI системи SAP Business Objects - це вище керівництво компанії.

SAP BusinessObjects забезпечує ефективний пошук та аналіз даних з різних джерел , дослідження залежностей і трендів, моделювання сценаріїв «що - якщо», моніторинг ключових показників і відображення інформації на інформаційному порталі. В якості джерел даних можуть виступати транзакційні системи (ERP, CRM, SCM), бухгалтерські та фінансові програми , корпоративні сховища даних, спеціалізовані галузеві рішення - АСУ ТП, білінг, скоринг, АБС та інше, а також локальні джерела - наприклад файли Excel або бази Access.

Web Intelligence - створення запитів, звітність і аналіз

Основні можливості:

* оперативна побудова нестандартних аналітичних запитів користувачами без необхідності звернення до ІТ;
* створення звітних форм;
* доступ до різних джерел даних;
* Drill-down навігація;
* візуалізація даних;
* робота через web-інтерфейс;
* інтеграція з додатками SAP.
* Області використання:
* проведення глибокого аналізу інформації;
* оперативне створення корпоративної звітності [15].

2.4.2 JasperSoft

JasperServer - вільний, безкоштовний з відкритим кодом сервер звітів, які генеруються за допомогою генератора JasperReports і OLAP-аналізу. JasperServer дозволяє консолідувати всю звітність в одному місці з можливістю доступу та управління нею через WEB-браузер [5].

iReport - графічний редактор звітів для роботи в середовищі NetBeans IDE; створений тією ж організацією, яка створила JasperReports; поставляється також у вигляді окремого додатка.

Провідний виробник засобів бізнес - аналітики (BI) та звітності компанія JasperSoft оголосила про те, що пропоноване нею open source BI - рішення є найпоширенішим у світі - воно завантажено понад 3 млн разів, розгорнуто більш ніж в 80 тис. систем, налічує 8 тис. комерційних клієнтів в 96 країнах, вбудовано в тисячі додатків, локалізовано на 24 мови [4].

JasperSoft охопила ключові сегменти ринків open source і SaaS. Компанія співпрацює більш ніж з 40 відомими виробниками ПО, в тому числі Ingres, MySQL, Red Hat, salesforce.com, Sun Microsystems, і Unisys, що сприяє поширенню її BI -рішень в корпоративному секторі. JasperSoft пропонує також адаптовані пакети Jasper4Oracle E - Business Suite, Jasper4SalesForce.com і Jasper4Sugar .

За даними JasperSoft, за останній рік число передплатників JasperSoft Business Intelligence Suite Professional Edition виросло на 227 %, цим ПЗ користуються Cincom, Ingres, Glovia Services (Fujitsu), SunGard, і Vericept Corporation [17].

Компанія JasperSoft також анонсувала Community Vibrancy Index ( CVI ). Дана методологія, призначена для аналізу та управління спільнотою зареєстрованих користувачів JasperSoft, яке тільки за останній рік зросла на 300% і перевищило 65 тис. чол. Їх зусиллями створюється більше 300 проектів на основі Jasper, 158 з яких було ініційовано в 2007 р.

Після закінчення тестування JasperSoft поширюватиме CVI за ліцензією Creative Commons.

Підтримка різних джерел даних: JDBC, CALS Table Models, XML, CSV, JavaBeans, EJBQL, Hibernate, а також можливість використання власного джерела даних на основі інтерфейсу JRDataSource.

Експорт в різні формати даних: PDF, HTML, XLS, RTF, ODT, CSV, XML, XHTML, DOCX, XLSX.

Використання так званих скріплет (scriptlets) на Java, які можуть бути викликані до або після певних етапів генерації звітів.

2.4.3 Crystal Reports

Crystal Reports – розроблена як допоміжний інструмент для аналізу та інтерпретації важливої ​​інформації при роботі з базою даних. Додаток Crystal Reports дозволяє легко створювати прості звіти. У ньому також передбачені повнофункціональні інструментальні засоби, необхідні для створення складних або спеціалізованих звітів. За допомогою Crystal Reports можна створювати звіт на основі практично будь-якого джерела даних. За допомогою формул, перехресних таблиць, підзвітів і умовного форматування можна зробити дані зрозумілими і виявити важливі взаємозв'язки, які можуть бути приховані в іншій ситуації. Географічні карти і діаграми дозволяють наочно представити дані, якщо опису та цифр недостатньо [15].

Гнучкість Crystal Reports не обмежується створенням звітів - можна публікувати звіти в різних форматах, включаючи Microsoft Word і Excel , для відправки по електронній пошті або розміщення в Інтернеті. Додаткові функції створення веб-звітів дозволяють іншим учасникам робочої групи переглядати і оновлювати спільно використовувані звіти з допомогою веб-браузера.

SQL Server Reporting Services – програмна серверна система створення звітів, розроблена корпорацією Microsoft. Вона може бути використана для підготовки безлічі інтерактивних та друкованих звітів. Система адмініструється через веб-інтерфейс. Reporting services використовують інтерфейс веб-служб для підтримки розробки звичайних звітних додатків.

Користувачі можуть працювати з веб-службою сервера звітів безпосередньо або використовувати диспетчер звітів-веб-додаток, що взаємодіє з веб-службою сервера звітів. За допомогою диспетчера звітів можуть переглядати і управляти звітами, також як і управляти і оперувати джерелами даних і налаштуваннями безпеки. Звіти можуть розсилатися по електронній пошті або записуватися на файлову систему як звичайний файл. Захист виконується на основі ролей і може накладатися на окремі елементи, як наприклад, звіт або джерело даних, каталог елементів або сайт взагалі. Ролі безпеки і права є успадкованими і можуть бути перевизначені .

На додаток до використання окремого сервера звітів, що поставляється з SQL Server, RDL- звіти можна переглядати за допомогою веб-контрола ASP.NET ReportViewer або Windows Forms, контрола ReportViewer. Це дозволяє вбудовувати звіти прямо в веб -сторінки або NET -додатки. Контрол ReportViewer обробляє звіти одним із двох способів : (а) на стороні сервера, де звіт обробляється сервера звітів; і (б) локальна обробка, де відповідний контрол самостійно обробляє RDL -файл.

SSRS змагається з Crystal Reports і іншими Business intelligence-інструментами, і входить до складу Express, Workgroup, Standard і Enterprise версій Microsoft SQL Server як встановленого доповнення. Reporting Services були вперше випущені в 2004 році як доповнення для SQL Server 2000. Друга версія була випущена у вигляді складової частини SQL Server 2005 в листопаді 2005 року. Наступна версія була випущена як частина SQL Server 2008 в серпні 2008. Остання на даний момент версія була випущена як частина SQL Server 2012 в березні 2012 . [12]

У SSRS звіти описуються за допомогою Report Definition Language ( RDL) на мові розмітки XML. Звіти можуть проектуватися за допомогою останніх версій Microsoft Visual Studio (включаючи Visual Studio.NET 2003 і Visual Studio 2005 ) з вхідним в них доповненням Business Intelligence Projects або за допомогою вхідного в комплект Report Builder - спрощеного інструменту, не пропонує повного функціоналу Visual Studio. Звіти, визначені за допомогою RDL, можуть створюватися в безлічі різних форматів, включаючи Excel, PDF, CSV, XML, TIFF (і інших графічних форматах ), а також HTML Web Archive. SQL Server 2008 SSRS також може підготовляти звіти у форматі Microsoft Word ( DOC). [16]

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПОКРАЩЕНИХ МЕТОДІВ ТА АНАЛІЗ

ОТРИМАНИХ РЕЗУЛЬТАТІВ

У даному розділі описуються результати вдосконалити методів збереження нетипізованих даних в інформаційних системах, порівнюються з існуючими аналогами, на реальних прикладах виявляються переваги та недоліки покращеного методу.

* 1. Моделювання ключових подій в судових справах

Просування судових справ визначаються по наявності ключових подій. Виділимо наступні ключові події:

* 1. Подача заяви про відкриття
  2. Постанова про відкриття
  3. Арешт та опис майна
  4. Оцінка майна
  5. Назначення тогрів
  6. Повернення

Ключові події задамо таким співвідношенням:

(1)

де K ⎯ номер ключової події,

x ⎯ вид ключової події,

y ⎯ тип ключової події,

z ⎯ суть ключової події.

Для визначення (1) використаємо матриці знань ключових подій.

* + 1. Матриця знань ключових подій

Ключову подію задамо такою матрицею знань:

(2)

Де ⎯ стрічка-ідентифікатор i-ої ключової події,

⎯ -й вид -ої ключової події,

⎯ -й тип -ої ключової події,

⎯ -а суть -ої ключової події,

N ⎯ кількість ідентифікаторів, що визначають -у ключову подію,

К ⎯ кількість ключових подій.

* + 1. Розробка алгоритму визначення ключової події

Алгоритм дозволяє визначити номер ключової події і складається з таких кроків:

1. Зафіксувати значення ідентифікаторів виду , типу , та суті ;
2. Зафіксувати ;
3. Якщо комбінація , то подія та перейти до кроку 6, інакше перейти до кроку 4;
4. Зафіксувати ;
5. Повторити кроки 3 та 4 R разів
6. Кінець
   1. Моделювання процесу використання судових справ

Задамо судову справу таким вектором

(3)

Де ⎯ кількість судових справ

⎯ вектор стадій виконання судової справи, при цьому

(4)

Якщо , то j-а стадія відсутня. Якщо , стадія відбулась.

Процес виконання ої судової справи є позтивним при такій умові:

(5)

Якщо хоча б одна стадія не виконана, то вважається що справа виконана з помилкою.

* + 1. Помилки виконання судових справ

Судові справи можуть виконуватись з помилками з таблиці 3.2.1

Таблиця 3.2.1. ⎯ можливі помилки в судовій справі

|  |  |  |
| --- | --- | --- |
| Номер помилки | Тип помилки | Опис помилки |
| 1 |  | Стадії 1 нема, стадія 2 є |
| 2 |  | Стадії 1 нема, стадії 2 нема |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |

3.2.2 Алгоритм визначення помилок комплектності

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

На початковому етапі розвитку охорони праці захист організму працюючої людини вирішувався на стадії виробництва. З розвитком продуктивних сил та під впливом науково-технічного прогресу цей шлях вирішення проблем охорони праці перестав відповідати нагальним вимогам техносфери. Виникла необхідність у глибокому вивченні біологічної, фізичної і хімічної сутності впливу шкідливих і небезпечних чинників, у прогнозуванні їх виникнення, щоб завчасно на підставі фундаментальних, технічних і медичних наук впроваджувати заходи захисту від них на стадії проектування, конструювання й виробничої експлуатації.

Нині проблеми охорони праці вирішуються на національному рівні в масштабах держави. Основна увага приділяється усуненню шкідливого впливу технологічних процесів на організм людини шляхом оздоровлення умов праці на виробництві.

Розробка системи управління портфелем юридичних процесів відбувалася в приміщені, яке обладнане комп’ютеризованими робочими місцями. На розробника, згідно ГОСТ 12.0.003-74 [2], могли мати вплив такі небезпечні та шкідливі виробничі фактори:

1. Фізичні: підвищена запиленість та загазованість повітря робочої зони; підвищений рівень шуму на робочому місці; підвищена чи понижена вологість повітря; підвищений рівень статичної електрики; підвищений рівень електромагнітного випромінювання; недостатня освітленість робочої зони.

2. Психофізіологічні: розумове перевантаження; перенапруга аналізаторів; статичне перевантаження.

Відповідно до визначених факторів формуємо рішення щодо безпечного виконання роботи під час розробки системи управління портфелем юридичних процесів.

4.1. Технічні рішення щодо безпечного виконання роботи

4.1.1. Обладнання приміщення та робочого місця

Проектування робочих місць, забезпечених відеотерміналами, відноситься до числа важливих проблем ергономічного проектування в області обчислювальної техніки.

Робоче місце і взаємне розташовання всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи. Зокрема, при організації робочого місця розробника повинні бути дотримані наступні основні умови: оптимальне розміщення устаткування, що входить до складу робочого місця і достатній робочий простір, що дозволяє здійснювати всі необхідні рухи і переміщення.

Приміщення, в якому знаходиться робоче місце системи управління портфелем юридичних процесів має загальну площу 24,75 м22, і висотою стелі 3,2 м. У приміщенні знаходиться 4 робочих місць з ПК (рис.1.1). Кожне робоче місце обладнане робочім столом площею 1,2 м2, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші.

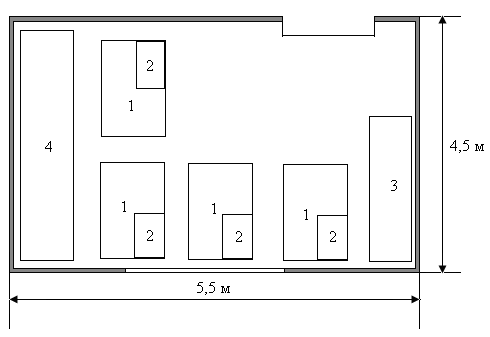


Рис 4.1.1. Схема приміщення

Умовні позначення: 1 – робочі місця (столи); 2 – персональні комп’ютери; 3 – стіл для принтера та копіювального апарату; 4 – шафа для паперів.

Площа приміщення, яка припадає на одного працівника становить 6,2 м2, об’єм повітря робочої зони – 19,8 м3. знайдені значення відповідають вимогам, оскільки площа одного робочого місця працівника, який використовує під час роботи ПК не повинна бути меншою за 6 м2 ,а об’єм не менший за 20м3 [7].

Ергономічними аспектами проектування відеотермінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до того, що розташовує документів на робочому місці (наявність і розміри підставки для документів, можливість різного розміщення документів, відстань від очей користувача до екрану, документа, клавіатури і т.д.), характеристики робочого крісла, вимоги до поверхні робочого столу, можливість регулювання елементів робочого місця.

Головними елементами робочого місця розробника є стіл і крісло. Основним робочим положенням є положення сидячи. Робоча поза сидячи викликає мінімальне стомлення працівника. Раціональне планування робочого місця передбачає чіткий порядок і постійність розміщення предметів, засобів праці і документації. Те, що потрібне для виконання робіт частіше, розташоване в зоні легкої досяжності робочого простору.

Для комфортної роботи стіл повинен задовольняти наступним умовам:

* висота столу повинна бути вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники;
* нижня частина столу повинна бути сконструйована так, щоб дослідника міг зручно сидіти, не був вимушений підтискати ноги;
* поверхня столу повинна володіти властивостями, що виключають появу відблисків в полі зору;
* конструкція столу повинна передбачати наявність висувних ящиків (не менше 3 для зберігання документації, канцелярських обладнань).
* висота робочої поверхні рекомендується в межах 680-760мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути біля 650мм [3].

4.1.2. Електробезпека приміщення

Згідно з ДНАОПБ 0.00-1.32-01 [9], приміщення, де експлуатуються ЕОМ і ПЕОМ, належать до приміщень без підвищеної небезпеки ураження людини електричним струмом. Вимоги електробезпеки і пожежної безпеки у приміщеннях, де встановлені ВДТ ЕОМ і ПЕОМ: ЕОМ і все устаткування для обслуговування, ремонту та налагодження роботи їх, електропроводи і кабелі мають відповідати електробезпеці зони за ДНАОПБ 0.00-1.32-01 та мати апаратуру захисту від струму короткого замикання.

Лінії електромережі ПК, у приміщенні виконана як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників (заземлення або занулення), причому площі перерізу нульового робочого і нульового захисного провідника повинні бути не менші за площу перерізу фазового провідника. Величина напруги цієї мережі становить 380 х 220В (фазна напруга (фаза – «0») – 220В, а міжфазна лінійна (фаза – фаза) – 380В). Категорія умов з небезпеки електротравматизму – без підвищеної небезпеки [8].

Електромережі для під’єднання ВДТ, ЕОМ і ПЕОМ оснащені штепсельними з’єднаннями та електророзетками, які, крім контактів фазового і нульового робочого провідників, мають спеціальні контакти для під’єднання нульового захисного провідника, що під’єднаний раніше ніж вони. Порядок роз’єднання при відімкненні мережі має бути зворотним.

Електромережі штепсельних з’єднань та електричних розеток виконані за магістральною схемою, по 3-6 в одному колі. Оскільки вони розташовані уздовж стін, то провідники прокладені по підлозі в металевих трубах і гнучких металевих рукавах. Металеві трубки і гнучкі металеві рукави заземлені.

Для запобігання електротравмам у приміщенні здійснюються:

1) ізоляція нормально струмоведучих елементів електроустаткування відповідно з вимогами нормативів;

2) захисне заземлення;

3) дотримання протипожежного режиму щодо експлуатації електрообладнання.

* 1. Технічні рішення з гігієни праці та виробничої санітарії
     1. Мікроклімат

Згідно ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [5] мікроклімат виробничих приміщень – умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих з оточенням шляхом конвекції, кондукції, теплового випромінювання та випаровування вологи. Ці умови визначаються поєднанням температури, відносної вологості та швидкості руху повітря, температури оточуючих людину поверхонь та інтенсивністю теплового (інфрачервоного) опромінення.

Мікроклімат виробничих приміщень нормується в залежності від теплових характеристик виробничого приміщення, категорії робіт по важкості і періоду року.

Робота розробника системи управління портфелем юридичних процесів за енерговитратами відноситься до категорії 1 а [1].

Допустимі параметри мікроклімату для категорії 1а наведені в табл.2.1.

Таблиця 4.2.1 – Параметри мікроклімату

|  |  |  |  |
| --- | --- | --- | --- |
| Період року | Допустимі | | |
| t,°C | W, % | V, м/с |
| Теплий | 22-28 | 55 | 0,1-0,2 |
| Холодний | 21-25 | 75 | 0,1 |

Для забезпечення необхідних за нормативами параметрів мікроклімату в приміщенні передбачено парова система опалення та система припливно-витяжної вентиляції. Раз за зміну здійснюється вологе прибирання та за необхідності – провітрювання через вікна та двері.

* + 1. Склад повітря робочої зони

В сучасній техніці застосовується безліч речовин, які можуть потрапляти в повітря і становити небезпеку здоров’ю людей. Для визначення небезпечності досліджується вплив цих речовин на організм людини і встановлюються безпечні для людини концентрації та дози.

В приміщенні, де здійснюється розробка системи управління портфелем юридичних процесів можлива підвищена концентрація пилу та озону. ГДК шкідливих речовин, які знаходяться в досліджуваному приміщені (за нормами ДСН 3.3.6.042-99 [5]), наведені в таблиці 2.2.

Таблиця 4.2.2 – ГДК шкідливих речовин у повітрі

|  |  |  |  |
| --- | --- | --- | --- |
| Назва речовини | ГДК, мг/м3 |  | Клас небезпечності |
| Максимально разова | Середньо добова |
| Пил нетоксичний | 0,5 | 0,15 | 4 |
| Озон | 0,16 | 0,03 | 1 |

Для гігієнічної оцінки повітря, крім хімічного складу, має значен­ня й іонний склад повітря. Чим чистіше повітря, тим більше воно містить легких від'ємних іонів. Параметри іонного складу повітря на робочому місці, що обладнане ПК, повинні відповідати допустимим нормам (табл..2.3).

Таблиця 2.3 – Рівні іонізації повітря приміщень при роботі на ПК

|  |  |  |
| --- | --- | --- |
| Рівні | Кількість іонів в 1 см3 | |
| n+ | n- |
| Мінімально необхідні | 400 | 600 |
| Оптимальні | 1500-3000 | 3000-5000 |
| Максимально необхідні | 50000 | 50000 |

Забезпечення складу повітря робочої зони здійснюється за допомогою системи припливно-витяжної вентиляції, регулярного провітрювання, та вологого прибирання.

* + 1. Виробниче освітлення

**Виробниче освітлення призначене для забезпечення виробничого процесу, переміщення людей, руху транспорту і є обов’язковим для всіх виробничих приміщень.**

Природне освітлення на даному робочому місці є бічне одностороннє.

Сучасні норми визначають, що мінімальна освітленість встановлюється за характеристикою зорової роботи з найменшим розміром об'єкта розрізнення, контрастом об'єкта із фоном і характеристикою фону.

Нормованим параметром природного освітлення являється коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Норми освітленості при штучному освітленні та КПО (для ІІІпояса світлового клімату) при природному та сумісному освітленні зазначені у таблиці 2.5:

Таблиця 4.2.5 - Норми освітленості в приміщенні

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Характеристика зорової роботи | Найменший розмір об'єкта розрізнювання | Розряд зорової роботи | Підрозряд зорової роботи | Контраст об'єкта розрізнення з фоном | Характеристика фона | Освітленість,  лк | | КПО, | | | |
| Штучне освітлення | | Природнє освітлення | | Сумісне освітлення | |
| Комбіноване | Загальне | Верхнє або верхнє  і бокове | Бокове | Верхнє або верхнє  і бокове | Бокове |
| Дуже  високої точності | Від 0,15  до 0,3 | ІІ | г | великий | світлий | 1000 | 300 | 7 | 2,5 | 4,2 | 1,5 |

Вимоги, які ставляться до раціонального освітлення: достатня освітленість робочого місця (нормована); рівномірне освітлення; відсутність тіней, особливо рухомих, на робочій поверхні; захист від сліпучої дії джерела світла; вірний вибір напрямку світла - дотримані.

В приміщенні використовуються світильники типу ОД. Кожен світильник комплектується двома лампами.

4.2.4.Виробничий шум

Шумна робочому місці розробника системи управління портфелем юридичних процесів виникає під час роботи системних блоків, принтерів, сканерів, обладнання для кондиціювання повітря, а також вентиляторів систем охолодження і трансформаторів.

Санітарні норми виробничого шуму, ультразвуку та інфразвуку відображені в ДСН 3.3.6.037-99 [4]. Для умов виконання роботи (характер робіт і характер шуму) допустимі рівні звукового тиску повинні відповідати ГС, а рівні звуку не повинні перевищувати 50 дБА - дивись таблицю 2.5.

Таблиця 4.2.5 – Допустимі рівні звукового тиску і рівні звуку для постійного широкополосного шуму

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Характер робіт | Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах зі середньогеометричними частинами (Гц) | | | | | | | | | Допуст-мий рівень звуку, дБА |
| 32 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |  |
| Виробничі приміщення | 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |

Для забезпечення допустимих параметрів шуму (поліпшення шумового клімату) в приміщенні доцільно використовувати комп’ютери з пасивним охолодженням.

4.2.5. Виробничі випромінювання

ПК під час роботи випромінює електромагнітну енергію, відповідно розробник піддається впливу електромагнітних полів з високочастотним випромінюванням. Інтенсивність електромагнітного поля нормується відповідно до ДСанПіН 3.3.2.007-98 [3] і оцінюється на робочому місці напруженістю електричного поля (у В/м) для електричної складової і напруженістю магнітного поля (у А/м) для магнітної складової. Допустимі значення параметрів електромагнітних випромінювань від монітору комп’ютера представлені в табл. 2.6.

Таблиця 4.2.6 – Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

|  |  |
| --- | --- |
| Найменування параметра | Допустимі значення |
| Напруженість електричної складової електромагнітного поля на відстані 50 см від поверхні відеомонітору | 10 В/м |
| Напруженість магнітної складової електромагнітного поля на відстані 50 см від поверхні відеомонітору | 0,3 А/м |
| Напруженість електростатичного поля не повинна перевищувати: | для дорослих користувачів 20кВ/м  для дітей 15кВ/м |

У відповідності до вимог і захисту від електромагнітних і електростатичних полів, допускається застосування екранних фільтрів і інших засобів індивідуальної захисту, які мають відповідний гігієнічний сертифікат. Крім того, доцільно дотримуватися рекомендованого режиму роботи за ПК.

4.2.6. Оцінка умов праці

Результати аналізу умов праці відповідно до категорії робіт га гігієнічною класифікацією [1] розробника системи управління портфелем юридичних процесів занесемо до карти умов праці (таблиця 2.7).

Таблиця 4.2.7 – Фрагмент карти умов праці на робочому місці

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Чинники виробничого  середовища і трудового процесу | Значення чинника (ГДК, ГДР) | | 3 клас - небезпечні і шкідливі умови | | | | Тривалість дії чинника,  % за зміну |
|  | Норма | Факт | 1 ст. | 2 ст. | 3 ст. | 4 ст. |  |
| **1. Мікроклімат:** |  |  |  |  |  |  |  |
| Температура повітря, 0С | 22...28°С | 276°С |  |  |  |  |  |
| Швидкість руху повітря, м/с | 0,1-0,2 м/с | 0,1 м/с |  |  |  |  |  |
| Відносна вологість, % | 40...60% | 55 % |  |  |  |  |  |
| **2. Освітлення:**  Природне (КПО, %) | 1,5 | 1,5 |  |  |  |  |  |
| Штучне, лк | 300...500 | 400 |  |  |  |  |  |
| **3 Напруженість трудового процесу**  Інтелектуальні навантаження |  |  | + |  |  |  |  |
| Розподіл функції за ступенем складності завдання |  |  |  |  |  |  |  |
| Навантаження на зоровий аналізатор |  | 5,0...1,1 мм більше 50 % часу | + |  |  |  |  |
| Спостереження за екранами ВДТ, годин за зміну |  | Більше 4 | + |  |  |  |  |
| **5. Режим праці**  Фактична тривалість робочого дня, год | 7 год | 7 год |  |  |  |  |  |
| Наявність регламентованих перерв та їх тривалість, % часу зміни | Більше 7% | від 3 % до 7 % часу зміни | + |  |  |  |  |
| **Загальна кількість чинників** |  |  | 4 |  |  |  |  |

Таким чином, за результатами проведеного дослідження, можна зробити висновок, що умови праці розробникасистеми управління портфелем юридичних процесів відносяться до допустимих. Періодичний вплив мають 4 шкідливі фактори І ступеня – це фактори напруженості трудового процесу.

Напруженість праці – характеристика трудового процесу, що відображає навантаження переважно на нервову систему. Напруженість праці визначається ступенем складності завдання; характером виконуваної роботи; сенсорним навантаженням (зорові, слухові аналізатори); емоційним навантаженням, монотонністю навантаження; щільністю робочого дня.

Організм людини може безхворобливо переносити вплив небезпечних та шкідливих чинників тільки доти, доки вони не перевищують оптимальних і допустимих рівнів та часу витривалості, що зумовлено функціональними можливостями людського організму. При роботі в умовах перевищення гігієнічних нормативів відбувається зниження працездатності, розвивається втомлення, яке суб'єктивно сприймається як втома.

В зв’язку з цим слід передбачити додаткові перерви протягом робочого дня, які рекомендовані згідно ДСанПіН 3.3.2.007-98 [3] під час роботи за ПК.

5 БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

* + 1. Оцінка пожежної безпеки

До основних джерел небезпеки в приміщенні, де здійснюється розробка системи управління портфелем юридичних процесів, відносяться персональні комп’ютери, периферійне обладнання, мережеві пристрої, розводка електричної мережі.

Ступінь вогнестійкості будинку, в якому розташоване приміщення, визначаємо за такими показниками конструкцій будівлі, як горючість, займистість, здатність поширювати полум’я, димоутворювання та токсичність продуктів горіння та за межами вогнестійкості його будівельних конструкцій і межами поширення вогню по цих конструкціях [1,2]. Дані занесемо в таблиці 5... та 5...

Таблиця. 5.1.1. ⎯ Показники пожежної небезпеки елементів і матеріалів конструкцій будівлі

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Конструкції/  Матеріал | Горючість | Займистість | Поширення  полум’я | Димоутво-рюваність | Токсичність продуктів горіння |
| Стіни, перегородки/  Цегла | НГ | - | - | - | - |
| Сходова площадка/  залізобетон | НГ | - | - | - | - |
| Перекриття / шифер | НГ | - | - | - | - |
| Вікна/  деревина | Г3 | В3 | РП3 | Д2 | Т2 |
| Підлога/  деревина | Г3 | В3 | РП3 | Д2 | Т2 |
| Оздоблення/  лакофарбові покриття | Г4 | В3 | РП4 | Д3 | Т3 |
| Каркас дах/  деревина | Г3 | В3 | РП3 | Д2 | Т2 |

Позначення в таблиці 5...: НГ – негорючі, Г3 – горючі; Г4 – підвищеної горючості; В3 – легкозаймисті; РП3 – помірно поширюють; РП4 – значно поширюють; Д2 – з помірною димоутворювальною здатністю; Д3 – з високою димоутворювальною здатністю; Т2 – помірно небезпечні; Т3 високонебезпечні.

Таблиця. 5.1.2. ⎯ Визначення ступеня вогнестійкості будівельних конструкцій

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ступінь  вогнестійкості  будинків | Мінімальні межі вогнестійкості будівельних конструкцій (у хвилинах) та максимальні межі поширення вогню по них (см) | | | | | | | | |
| стіни | | | | колони | сходові площадки, косоури, сходи,  балки, марші сходових кліток | перекриття міжповерхові (у т. ч. горищні та над підвалами) | елементи суміщених покриттів | |
|  | несучі та сходових кліток | самонесучі | зовнішні ненесучі | внутрішні ненесучі (перегородки) | плити,  настили, прогони | балки, ферми, арки, рами |
| ІІІ | REI 120  M0 | REI 60  M0 | E15, M0  E30, M1 | EI 15  M1 | R 120  M0 | R 60  M0 | REI 45  M1 | Не нормуються | |

Згідно даних наведених у таблицях і з урахуванням [1,2,4] приміщення, де здійснюється розробка системи системи управління портфелем юридичних процесів, можна віднести до категорії вибухопожежонебезпеки В (легкозаймисті, горючі і важкогорючі, а також речовини та матеріали, які здатні горіти; тверді горючі та важкогорючі речовини і матеріали, за умови, що приміщення, в яких вони знаходяться, не відносяться до категорій А, Б і питома пожежна навантага для твердих і рідких легкозаймистих та горючих речовин на окремих ділянках площею не менше 10 м2 кожна перевищує 180 МДж/м2). Клас приміщення і зон з вибухо- і пожежонебезпеки П-ІІа (приміщення, у якому знаходяться тверді горючі речовини та матеріали.) [1,4].

* + 1. Правила пожежної безпеки

До правил пожежної безпеки приміщення відносяться:

* відсутність транзитних електричних проводів і кабелів в приміщенні обчислювального центру;
* заборона експлуатації кабелів і проводів з пошкодженою або такою, що в процесі експлуатації втратила захисні властивості, ізоляцією;
* знеструмлення кабелів та проводів з неізольованими струмопровідними жилами;
* запобігання використання саморобних подовжувачів, які не відповідають вимогам ПУЕ та ПБЕ, що пред'являються до переносних (пересувних) електропроводок;
* заборона застосування для опалення приміщення в холодний період електронагрівального обладнання;
* заборона користування пошкодженими розетками, відгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
* запобігання підвішування світильників безпосередньо на струмопровідні проводи, експлуатація їх зі знятими ковпаками (розсіювачами);
* запобігання використання ПЕОМ та периферійних пристроїв в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів;
* запобігання використання в пожежонебезпечних зонах світильників з лампами розжарювання без захисного суцільного скла (ковпаків), а також з відбивачами і розсіювачами, виготовленими з горючих матеріалів;
* запобігання залишення без догляду при виході з приміщення увімкненого електрообладнання тощо;
* запобігання влаштування та експлуатація тимчасових електромереж (винятком можуть бути тимчасові ілюмінаційні установки і електропроводки, які живлять місця проведення будівельних, тимчасових ремонтно-монтажних та аварійних робіт).

5.1.3. Засоби пожежогасіння

До основних засобів пожежогасіння, що використовуються в таких приміщеннях, відносять вогнегасники, розміщення яких в приміщеннях позначається встановленим на висоті 2-2,5 м вказівним знаком.

Згідно категорії пожежовибухонебезпеки будівлі, класу приміщення і за вибухо- і пожежонебезпекою П-ІІа, в приміщенні, де здійснюється розробка системи управління портфелем юридичних процесів, має бути встановлено по одному порошковому або вуглекислотному вогнегаснику із зарядом речовини від 3 кг[5] на кожні 20 м2 площі.

5.1.4 Інженерний захист працівників та населення

Інженерний захист працівників та населення являє собою комплекс заходів, що реалізуються на етапі проектування та будівництва потенційно-небезпечних об’єктів з метою запобігання та розвитку надзвичайних ситуацій пов’язаних з ними, та створення фонду захисних споруд призначених для укриття населення в разі виникнення надзвичайних ситуацій.

При проектуванні й експлуатації споруд та інших об‘єктів господарювання, наслідки діяльності яких можуть шкідливо вплинути на небезпеку населення і навколишнього середовища, обов‘язково розробляються і здійснюються заходи інженерного захисту, які повинні передбачати:

* 1. Облік при розробці генеральних планів забудови населених пунктів і веденні містобудування можливих проявів в окремих регіонах і на окремих територіях небезпечних та катастрофічних явищ.
  2. Раціональне розміщення об‘єктів підвищеної небезпеки з урахуванням можливих наслідків їх діяльності у випадку виникнення аварій для безпеки населення і навколишнього середовища.
  3. Будівництво будинків, будівель, споруд, інженерних мереж і транспортних комунікацій із заданими рівнями безпеки і надійності.
  4. Розробку і впровадження заходів безаварійного функціонування об‘єктів підвищеної небезпеки.
  5. Створення комплексної схеми захисту населених пунктів і об‘єктів господарювання від небезпечних природних процесів.
  6. Розробку і здійснення регіональних та місцевих планів запобігання й ліквідації наслідків надзвичайних ситуацій техногенного і природного характеру.
  7. Організацію будівництва протизсувних, протипаводкових, протиселевих, протилавинних, протиерозійних та інших інженерних споруд спеціального призначення.
  8. Реалізацію заходів [санітарної охорони](http://udec.ntu-kpi.kiev.ua/lspace/oborona_udec_demo/schedule.nsf/d862e82eafb758368525663c004f385c/eeef896b293b3fcdc22570150049bbf9?OpenDocument#_c0n8gbo05tk2r61fi0ng0bs05tk2u0805to2u21fg0nn0bo85t82u0_) території.

Укриттю в захисних спорудах, у разі потреби, підлягає населення відповідно до його приналежності до груп (працююча зміна, населення, яке проживає в небезпечних зонах).

Створення фонду захисних споруд забезпечується шляхом:

* комплексного освоєння підземного простору міст і населених пунктів для взаємопогоджуваного розміщення в ньому споруд і приміщень соціально-побутового, воєнного і господарського призначення з урахуванням необхідності пристосування і використання частини приміщень для укриття населення у випадку виникнення надзвичайних ситуацій техногенного і природного характеру;
* обстеження й взяття на облік підземних і наземних будівель і споруд, які відповідають вимогам захисту споруд підземного простору міст, гірничих виробок і природних пустот;
* дообладнування з урахуванням реальної обстановки підвальних й інших заглиблених приміщень;
* будівництва заглиблених споруд, окремо розташованих від об‘єктів виробничого призначення і пристосованих для захисту;
* масового будівництва в період загрози виникнення надзвичайних ситуацій техногенного і природного характеру найпростіших сховищ і укриттів;
* будівництва окремих сховищ і протирадіаційних укриттів.

Перелік таких сховищ, укриттів та інших захисних споруд, які необхідно будувати, щорічно визначається спеціально уповноваженим центральним органом виконавчої влади, до компетенції якої віднесені питання захисту населення і території від надзвичайної ситуації техногенного і природного характеру, і затверджуються Кабінетом Міністрів України.

Висновок

В підрозділі проаналізовано рівень забезпечення пожежної безпеки в приміщенні, де здійснюється розробка системи управління портфелем юридичних процесів. Наведено основні правила пожежної безпеки. Відповідно призначенню приміщення та використовуваних в ньому приладів і обладнання здійснено вибір необхідних засобів пожежогасіння.

Розглянуто питання інженерного захисту працівників підприємства та населення в надзвичайних ситуаціях.

ЛІТЕРАТУРА

1. Том Уайт. Hadoop: The Deﬁnitive Guide. 3rd Edition / Том Уайт. – CY Reilly Media. – 2012. – 688 с.
2. Марк А. Бейер. The Importance of Big Data: A Deﬁnition / Марк А. Бейер, Дуглас Лени. – Reilly Media. – 2012. – 459с
3. Карло Строці. NoSQL: A Relational Database Management / Карло Строці. – Reilly Media. – 2013. – 553с.
4. Ярослав Покорний. NoSQL databases: a step to database scalability in web emirunment / Ярослав Покорний. – ACM New York. NY. USA. – 201l. –283с.
5. Фей Чанг. BigTablе: a distributed storage system for structured data / Фей Чанг, Джефрі Дін, Санья Гемават. – USENIX Association Berkeley. CA. USA. 2006. – 604с.
6. Рік Каттел. Scalable SQL and NoSQL data stores / Рік Каттел. – AC M SIGMOD Record. – ACM New York. NY. USA. – 2010. – 470с.
7. Санья Гемават. The Google File System / Санья Гемават , Говард Гобінов. – ACM New York. NY. USA. Octoher. – 2003. – 326с.
8. Джефрі Дін. MapReduce: simpliﬁed data processing on large clusters. / Джефрі Дін, Санья Гемават. – USENIX Asociation Berkeley. C A. USA. – 2004. – 350 с
9. Apache Hadoop [Електронний ресурс]: Apache Hadoop // Apache Hadoop – Режим доступу: http://hadoop.apache.org. – Назва з екрану.
10. Apache CouchDb [Електронний ресурс]: Apache CouchDb // Apache CouchDb – Режим доступу: http://couchdb.apache.org. – Назва з екрану.
11. MongoDB [Електронний ресурс]: MongoDB // MongoDB – Режим доступу: http://mongodb.org. – Назва з екрану.
12. Riak [Електронний ресурс]: Riak // Riak – Режим доступу: http://basho.com/products/riak-overview. – Назва з екрану.
13. Кріс Андерсон. CouchDB: The Deﬁnitive Guide. / Кріс Андерсон, Ян Леонард, Ной Слатер. – O'Re1'lly Mam. – 2011. – 242с.
14. Констянтин Швачко. Robert Chansler. The Hadoop / Констянтин Швачко, Хйронг Куанг, Санья Радія. – O'Re1'lly Mam. – 2010. – 212с.
15. Мохамед Хатіб. Mass Storage Systems and Technologies / Мохамед Хатіб, Майкл Фактор. – MSST. – 2010. – 168с.
16. Sort Benchmark Home Page [Електронний ресурс] Sort Benchmark // Sort Benchmark – Режим доступу: http://sortebenchmark.org. – Назва з екрану.
17. Аджей Ананд. Hadoop Sorts a Petabyte in 16.25 hours and a Terabyte in 62 Seconds / Аджей Ананд – MSST. – 2009. – 219с
18. Apache Hive [Електронний ресурс]: Apache Hive // Apache Hive – Режим доступу: http://hive.apache.org. – Назва з екрану.
19. Apache Pig [Електронний ресурс]: Apache Pig // Apache Pig – Режим доступу: http://pig.apache.org. – Назва з екрану.
20. Apache Hbase [Електронний ресурс]: Apache Hbase // Apache Hhase – Режим доступу: <http://hbase.apache.org>. – Назва з екрану.
21. [Documentation | Jaspersoft Community](https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CEAQFjAC&url=http%3A%2F%2Fcommunity.jaspersoft.com%2Fdocumentation&ei=zA6GU_TOE-_14QS4t4CADA&usg=AFQjCNGeegQmweTAo70zuiLe5smRJ0y8Bg&sig2=tdlT-okQ9GJhOZmQ98-uFg&bvm=bv.67720277,d.bGE&cad=rjt) : [електронний ресурс] – Режим доступу до ресурсу : [http://community.jaspersoft.com/documentation? version=9036](http://community.jaspersoft.com/documentation?%20version=9036) - Назва з екрану.
22. JasperSoft – самое распространенное BI-решение в мире : [електронний ресурс] – Режим доступу до ресурсу : [http://ko.com.ua/jaspersoft\_ samoe\_rasprostranennoe\_bi-reshenie\_v\_mire\_35042](http://ko.com.ua/jaspersoft_%20samoe_rasprostranennoe_bi-reshenie_v_mire_35042) - Назва з екрану.
23. JasperReports : [електронний ресурс] – Режим доступу до ресурсу : <http://ru.wikipedia.org/wiki/JasperReports> - Назва з екрану.
24. JasperWave Report Designer : [електронний ресурс] – Режим доступу до ресурсу : <http://www.obnovi-soft.ru/jaspersoft-studio> - Назва з екрану.
25. SAP BusinessObjects - решение для бизнес-аналитики : [електронний ресурс] – Режим доступу до ресурсу : <http://www.softline-erp.ru/sapbusinessobjects/default.aspx> - Назва з екрану.
26. [50 открытых Business Intelligence продуктов](https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&cad=rja&uact=8&ved=0CIcBEBYwCQ&url=http%3A%2F%2Fniap.oaosu.ru%2Fpost%2F5%2F50%2B%25D0%25BE%25D1%2582%25D0%25BA%25D1%2580%25D1%258B%25D1%2582%25D1%258B%25D1%2585%2BBusiness%2BIntelligence%2B%25D0%25BF%25D1%2580%25D0%25BE%25D0%25B4%25D1%2583%25D0%25BA%25D1%2582%25D0%25BE%25D0%25B2%2B&ei=zA6GU_TOE-_14QS4t4CADA&usg=AFQjCNHuOxIG8kJ23219y3YMI0hd0-Ro2w&sig2=Pg1Dn5tjZiZIv7mqJ922-g&bvm=bv.67720277,d.bGE) аналитики : [електронний ресурс] – Режим доступу до ресурсу : [http://niap.oaosu.ru/post /5/50+%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85+Business+Intelligence+%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%BE%D0%B2](http://niap.oaosu.ru/post%20/5/50+%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85+Business+Intelligence+%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%BE%D0%B2)+ - Назва з екрану.
27. [Основы бизнес-анализа](http://www.uml2.ru/articles/22-system-analysis/464-) : [електронний ресурс] – Режим доступу до ресурсу : <http://www.uml2.ru/articles/22-system-analysis/464-> - Назва з екрану.
28. Бізнесова аналітика : [електронний ресурс] – Режим доступу до ресурсу : <http://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%B7%D0%BD%D0%B5%D1%81%D0%BE%D0%B2%D0%B0_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D1%82%D0%B8%D0%BA%D0%B0> - Назва з екрану.
29. Покращення ефективності бізнес-аналізу й роботи працівників завдяки бізнес-аналітиці та засобам звітування системи Microsoft Dynamics AX : [електронний ресурс] – Режим доступу до ресурсу : <http://www.microsoft.com/uk-ua/dynamics/products/ax-business-intelligence.aspx> - Назва з екрану.
30. Бизнес-анализ в ИТ : [електронний ресурс] – Режим доступу до ресурсу : <http://www.linkedin.com/groups/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7-%D0%B2-%D0%98%D0%A2-4630157> - Назва з екрану.
31. Описание SAP Crystal Reports : [електронний ресурс] – Режим доступу до ресурсу : <http://www.interface.ru/home.asp?artId=16510> - Назва з екрану.
32. SAP Crystal Reports : [електронний ресурс] – Режим доступу до ресурсу : <http://www.mont.ru/vendors/PageInfo.aspx?PageID=1831&ProviderID=44>
33. [Web Intelligence Consortium](http://wi-consortium.org/) : [електронний ресурс] – Режим доступу до ресурсу : <http://wi-consortium.org/> - Назва з екрану.
34. Web Intelligence Report Server : [електронний ресурс] – Режим доступу до ресурсу : <http://www.businessobjectstips.com/tips/server/sizing-limits-to-web-intelligence-report-server-maximum-simultaneous-connections/> - Назва з екрану.
35. SQL Server Reporting Services : [електронний ресурс] – Режим доступу до ресурсу : <http://ru.wikipedia.org/wiki/SQL_Server_Reporting_Services> - Назва з екрану.
36. Отчетность и интерактивный анализ данных с помощью Reporting Services (SSRS) Services : [електронний ресурс] – Режим доступу до ресурсу : <http://microsoftbi.ru/basics/reporting/> - Назва з екрану.
37. Интеграция SQL Server 2008 R2 Reporting Services и SharePoint 2010 : [електронний ресурс] – Режим доступу до ресурсу : <http://www.oszone.net/12998/SQL-Server-2008-R2-Reporting-Services-SharePoint-2010> - Назва з екрану.
38. SQL Server Reporting Services (SSRS) : [електронний ресурс] – Режим доступу до ресурсу : <http://www.mssqltips.com/sqlservertutorial/222/sql-server-reporting-services-ssrs/> - Назва з екрану.
39. Pro SQL Server 2008 Reporting Services) : [електронний ресурс] – Режим доступу до ресурсу : <http://forcoder.ru/databases/pro-sql-server-2008-reporting-services-440> - Назва з екрану.
40. Microsoft SQL Server Standard Edition. Коммерческие лицензии : [електронний ресурс] – Режим доступу до ресурсу : <http://www.legasoft.ru/licenzionnoe-po/servernie-sistemi-i-prilozheniya/microsoft/microsoft-sql-server-standard-edition-2008r2/> - Назва з екрану.
41. Usiness intelligence suites : [електронний ресурс] – Режим доступу до ресурсу : [http://www.open-source-guide.com/en/Solutions/Applications/Business-inte lligence-suites/Jaspersoft](http://www.open-source-guide.com/en/Solutions/Applications/Business-inte%20lligence-suites/Jaspersoft) - Назва з екрану.
42. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/v4137400-86>
43. ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация.
44. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ЕОМ - [Електронний ресурс] - Режим доступу: <http://document.ua/derz-nor4881.html>
45. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. - [Електронний ресурс] - Режим доступу: <http://document.ua/sanitarni-normi-virobnichogo-shumu-ultrazvuku-ta-infrazvuku-nor4878.html>
46. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. - [Електронний ресурс] - Режим доступу: <http://mozdocs.kiev.ua/view.php?id=1972>
47. Закон України «Про охорону праці» від 14.10.1992 № **2694-XII**- [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/2694-12>
48. НПАОП 0.00-1.28-10 Правила охраны труда при эксплуатации электронно -вычислительных машин. - [Електронний ресурс] - Режим доступу: <http://document.ua/pravila-ohoroni-praci-pid-chas-ekspluataciyi-elektronno-obch-nor17970.html>
49. Правила улаштування електроустановок - [Електронний ресурс] - Режим доступу: <http://www.energiy.com.ua/PUE.html>
50. ДНАОПБ 0.00-1.32-01 Правила будови електроустановок та електрообладнання. - [Електронний ресурс] - Режим доступу: <http://aer.net.ua/public/files/OSBB/zakonodavctvo/INSCE/DNAOP%200.00-1.32-01..pdf>

ДОДАТКИ

Додаток А (обов'язковий)

Схема програми запиту по стадіях документообороту юридичної компанії



Додаток Б (обов'язковий)

Змн.

Лист

№ докум.

Підпис

Дата

Розроб.

Перевір.

Реценз.

Н. Контр.

Затверд.

Літ.

Маса

Масштаб

Лист

Листів 1

ВНТУ, гр. 1СІ-10б

Бевз О.М.

Бойко О.Р.

Т. Контр.

Бевз О.М.

Гринюк Т.С.

Квєтний Р.Н

08-02.БДР.004.00.000 СП1

Автоматизація побудови звітів за допомогою JasperServer та IReport

Схема програми запиту по кількості та сумі поповнень скретч картами

Схема програми по виконанню часових нормативів по стадіях документообороту юридичної компанії

Додаток В (обов'язковий)

Змн.

Лист

№ докум.

Підпис

Дата

Розроб.

Перевір.

Реценз.

Н. Контр.

Затверд.

Літ.

Маса

Масштаб

Лист

Листів 1

ВНТУ, гр. 1СІ-10б

Бевз О.М.

Бойко О.Р.

Т. Контр.

Бевз О.М.

Непийвода А.Ю.

Квєтний Р.Н

08-02.БДР.004.00.000 СП2

Автоматизація побудови звітів за допомогою JasperServer та IReport

# Схема програми по виконанню часових нормативів по стадіях

Вихідні дані

1 РОЗРОБКА АЛГОРИТМУ ЗВІТУ ПО ВИКОНАННЮ ЧАСОВИХ НОРМАТИВІВ ПО СТАДІЯХ ДОКУМЕНТООБОРОТУ ЮРИДИЧНОЇ КОМПАНІЇ





Змн.

Лист

№ докум.

Підпис

Дата

Розроб.

Перевір.

Реценз.

Н. Контр.

Затверд.

Літ.

Маса

Масштаб

Лист

Листів 1

ВНТУ, гр. 1СІ-10б

Бевз О.М.

Бойко О.Р.

Т. Контр.

Бевз О.М.

Непийвода А.Ю.

Квєтний Р.Н

08-02.БДР.004.00.000 ПД

Автоматизація побудови звітів за допомогою JasperServer та IReport

# Вихідні дані

