

## **МЕТОД СТВОРЕННЯ ВІДМОВОСТІЙКОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ КЛАСТЕРУ СЕРВЕРІВ**

Бурхливий розвиток інформаційних технологій, зростання кількості даних, що обробляються та обсягу їх передач, і, в той же час, підвищення вимог до надійності, ступеня готовності, відмовостійкості і масштабованості змушують поновому поглянути на вже відому технологію кластеризації. Дана технологія дозволяє створювати досить гнучкі системи, які здатні вирішити широке коло завдань і задовольнити частину вищеперерахованих вимог.

Постійне зростання вимог до продуктивності, доступності та захищеності кластерних систем вимагає знайдення нових інформаційних технологій. Для вирішення поставленої задачі необхідним є використання системного аналізу для дослідження побудови кластерних систем та знайдення нових інформаційних технологій їх побудови. Потрібно тільки чітко визначити найбільш істотні відмінності між різними видами кластерів, а також знайти переваги тих чи інших систем – з точки зору ефективності застосування їх у справі. Наступним кроком є поєднання переваг для створення нової технології їх побудови.

Важливий внесок у дослідження продуктивності інформаційних систем внесли своїми працями такі вчені як: Джин Амдал, Майкл Флін, Сеймур Крей, Ерік Шмідт, Вірт Ніклаус. В своїх роботах вони відобразили різноманітні аспекти роботи інформаційних систем, у тому числі і тих, що були основані на роботі кластеру серверів.

До показників якості кластеру серверів відносяться: надійність, доступність, відмовостійкість, обчислювальну потужність. Відповідно до тематики було проведено дослідження технологій створення відмовостійких інформаційних систем на основі кластеру серверів. Проведений аналіз кластерних систем та вимог до їх побудови виявив, що найважливішими вимогами є безперервна та безвідмовна робота. Для задоволення даних вимог проведено проектування високодоступної та високопродуктивної кластерної системи, введення системи в експлуатацію та проведення аналізу її роботи при робочому навантаженні з метою знайдення вузьких місць. Проведений аналіз предметної області дозволив сформулювати функціональну та структурну схеми відмовостійкої кластерної системи. До функціональних модулів системи можна віднести:

1. Клієнтський додаток – програма з інтерфейсною частиною за допомогою якої користувач використовує/записує необхідну йому інформацію.
2. Фізичний сервер – апаратне забезпечення на якому проводяться всі необхідні обрахунки та на якому інформація зберігається у фізичному вигляді.
3. SQL-сервер – програмне забезпечення яке обробляє запити користувача шляхом звертання до бази даних за допомогою мови SQL.
4. База даних – впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.

В результаті проведеного експерименту було виявлено та ліквідовано помилки, які були зроблені на стадії проектування. Критерій відмовостійкості та доступності задовольнили шляхом створення відмовостійкого кластеру серверів. Принцип роботи запропонованої системи відносно простий, але відповідає всім поставленим вимогам. Два або більше фізичних сервери, що знаходяться в різних місцях об'єднуються в один логічний сервер. Дані, які користувач використовує, або вносить на сервер при роботі з системою дублюються на двох фізичних серверах. При нормальній роботі системи клієнтська частина системи (додаток) з'єднується з тим фізичним сервером з яким має кращий зв'язок на даний момент. Після внесення даних, вони спочатку з'являються на тому сервері з яким в даний момент з'єднаний клієнт, а через певний час, за допомогою вбудованих алгоритмів дана інформація дублюється на іншому сервері кластеру. У разі відмови роботи сервера з яким працює користувач, додаток встановлює зв'язок з іншим сервером кластеру на якому знаходиться уся необхідна інформація. Виникнення даної ситуації може зменшити продуктивність роботи системи, адже другий сервер може знаходитися за тисячі кілометрів від першого, проте роботу з системою буде продовжено і клієнт далі зможе виконувати необхідні для нього задачі. Запропонований приклад може бути застосований у випадку коли необхідно забезпечити безперервну роботу інформаційної системи. Даний приклад вирішує лише проблему надійності та доступності системи. Якщо необхідно збільшити інші показники якості системи потрібно змінювати алгоритм роботи системи відповідно до поставлених вимог замовника інформаційної системи.