

АРХІТЕКТУРА АСУТП

10.1. Завдання проектування

Основне завдання, яке повинні вирішувати інженери АСУ на підприємствах, полягає не тільки в тому, щоб домагатися максимальної продуктивності при мінімальній вартості системи, а й закласти основи розширення системи в майбутньому для задоволення зростаючих вимог підприємства. Оскільки виробничий процес і технології постійно змінюються, система моніторингу та управління повинна адекватно відстежувати ці зміни, тобто легко модифікуватися при зміні завдання і рости в міру розвитку підприємства. Це можливо лише в тому випадку, коли архітектура АСУТП є масштабованою.

10.2. Архітектура АСУТП

У минулому інженери повинні були вибирати між двома взаємовиключними архітектурами - централізованою або розподіленою. На жаль, жодна з них не може задовільняти потреби сучасного підприємства. На початку 80-х років централізована архітектура набула популярності, оскільки

один великий комп'ютер міг здійснювати всі управління виробничим процесом і зберігати дані в єдиній БД (рис. 10.1). Всі оператори в такій системі мали однаковий доступ до даних, оскільки вони зберігалися в одному місці, і тільки один комп'ютер потребував оновлення при зміні вимог виробничого процесу.



Рис.10.1 Централізована архітектура

Однак ця архітектура має ряд серйозних недоліків:

- початкові інвестиції занадто високі для невеликих додатків;
- фіксована ємність системи не допускає послідовного збільшення системи при розширенні підприємства або зміні конфігурації;
- резервування може бути досягнуто тільки дублюванням всієї системи в цілому;
- вимоги до технічної кваліфікації обслуговуючого персоналу дуже високі і часто вимагають дорогого навчання.

При повністю розподіленій архітектурі, популярній наприкінці 80-х років, завдання управління і збору даних виконуються на декількох невеликих комп'ютерах (як правило, ПК). Кожен комп'ютер вирішує власну задачу і працює зі своєю базою даних (БД), як показано на рис. 10.2.

Розподілена система є досить гнучкою при малій величині початкових інвестицій. Вимоги до кваліфікації операторів невисокі, і спеціалізоване навчання потрібно в обмеженому обсязі.



Рис.10.2 Розподілена архітектура

Резервування може застосовуватися вибірково до завдань, критично важливим для підприємства. Така система вирішує проблеми, властиві централізованим архітектурам управління.

Однак розподілені системи також мають свої недоліки:

- обробка даних відбувається на кожному комп'ютері, що призводить до вкрай неефективного використання обчислювальних потужностей;
- відсутня оптимізація запитів до даних виробництва - якщо два оператора запитують одні і ті ж відомості, запит повторюється двічі.

Переваги централізованої архітектури є недоліками розподіленої системи і навпаки.

Усвідомлення того факту, що потрібен принципово інший підхід до побудови АСУТП, привело на початку 90-х років до появи нових розробок. Автором найбільш відомого рішення є фірма CіTechnologies, яка запропонувала в 1992 р. програмний пакет Citect для Windows.

Citect для Windows пропонує інноваційний підхід в реалізації системної архітектури, що використовує найкращі властивості централізованої і розподіленої обробки даних. Архітектура Citect є революційною не лише по відношенню до АСУТП, але поширюється і на інші програми, що працюють з розподіленими ресурсами, об'єднаними в мережі.

Архітектура клієнт-сервер. Основна проблема таких систем полягає в способі розподілу БД. Коли ПК об'єднані в локальну мережу (ЛВМ), БД може бути централізована, а дані надходять до індивідуальним вузлів. Якщо необхідно отримати певну інформацію з БД безпосереднє звернення до БД і пошук у ній здійснюються тільки комп'ютером, відповідальним за управління цією базою. Такий комп'ютер, що володіє необхідним інтерфейсом з обробки запитів, функціонує як сервер інформації. Коли вузол відображення (або інший

клієнт) посилає запит до БД, він не повинен здійснювати пошук в базі самостійно, а лише запитувати ці дані у сервера. Це і становить основу архітектури клієнт-сервер.

Оптимізація клієнт-серверної обробки даних в Citect. Для того щоб розуміти, як Citect використовує архітектуру клієнт-сервер, необхідно мати чітке уявлення про основні завдання Citect: введення/виведення; відображення; тривоги; графіки; звіти.

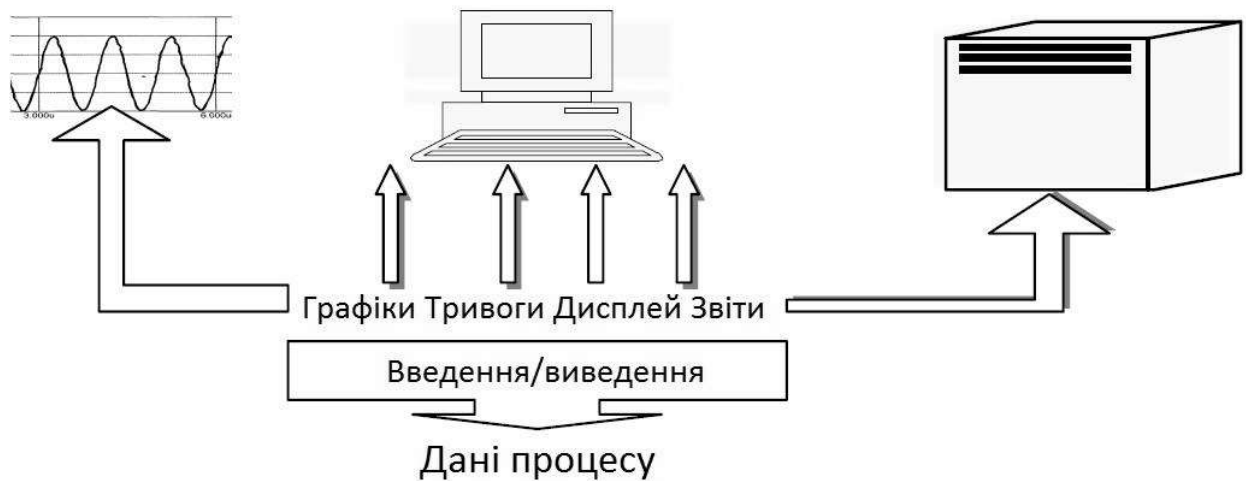


Рис. 10.3 Системна архітектура Citect

Кожне з цих завдань управляє власною БД незалежно від інших. Такий поділ БД підтримується навіть тоді, коли всі завдання виконуються на одному і тому ж комп'ютері (рис. 10.3).

Введення/виведення є інтерфейсом між системою управління-контролю і виробничим процесом. Він оптимізує і управляє процесом обміну даними на підприємстві між фізичними пристроями.

Тривоги відповідають за генерацію сигналів тривоги шляхом аналізу стану дискретних змінних і порівняння значень аналогових змінних із заданим порогом.

Графіки управляють усіма даними, які необхідно відображати з плином часу. Зібрані і оброблені вони надсилаються задачі відображення за запитом останньої.

Звіти. Завдання з обробки звітів збирає необхідні дані, в тому числі з тривог і графіків, і генерує звіти по заданому критерію. Звіти можуть вироблятися періодично, при настанні деякої події або ініціюватися оператором.

Відображення інформації є сполучною ланкою з оператором, основною складовою ЧМІ. Цей процес керує всіма даними, призначеними для відстеження оператором і виконання дій, ініційованих оператором. У рамках відображення здійснюється доступ до всіх тривог, графіків і звітів.

Кожен з описаних вище процесів функціонує незалежно від інших. Введення / виведення, тривоги, графіки та звіти мають спільну рису - оскільки вони потребують доступу до одних і тих же даних виробничого процесу; такі завдання краще вирішуються в рамках централізованої архітектури. Завдання відображення більше підходить розподіленій архітектурі, оскільки, як правило, в системі присутні більше одного оператора.

Клієнт-серверна архітектура - з розподіленими завданнями як клієнтами і загальними завданнями як серверами – оптимізує розподіл інформації між БД. Citect для Windows може підтримувати високу продуктивність навіть при розподілі завдань по багатьом комп'ютерам.

Масштабна архітектура. Оскільки архітектура клієнт-сервер дозволяє розподіляти підзадачі, конструктори системи не пов'язані звичайними апаратними обмеженнями. Результатом є масштабована архітектура, яка, може бути адаптована до додатків будь-якого розміру - рішення, що скасовує безліч обмежень звичайних систем і забезпечує результати, досі неможливі в АСУ.

Кращим способом виявлення великого потенціалу масштабованої архітектури для будь-якої програми може бути її застосування в серії невеликих прикладів з практики. В невеликих додатках один комп'ютер управляє всіма тривогами, графіками, звітами і завданнями введення / виведення. Система може бути повністю незалежною або інтегрованою в існуючу структуру (рис. 10.4).

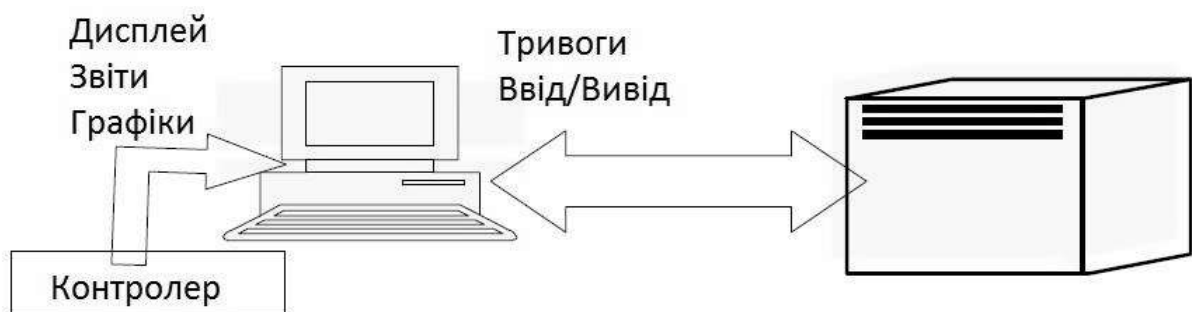


Рис. 10.4 Масштабована архітектура

У міру розширення програми (наприклад, додаються два вузли) додатковий комп'ютер може бути використаний для кожного вузла - і на кожному встановлюється ПЗ Citect для Windows (рис. 10.5).

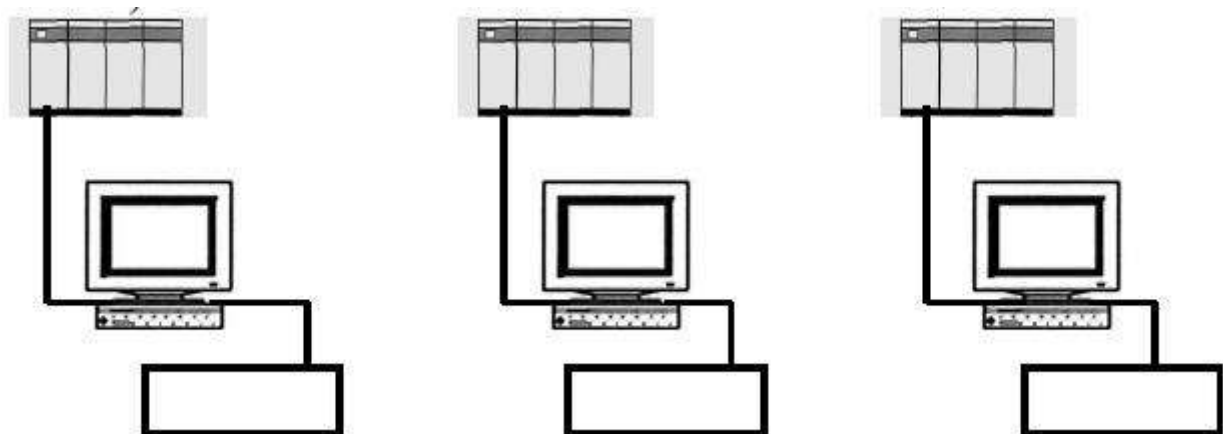


Рис. 10.5 ПЗ Citect встановлено на кожному вузлі



Рис. 10.6 Схема з додаванням локальної мережі

Однак таку схему можна покращити додаванням ЛВМ і виділеного сервера введення/виведення. Така централізація усуває непотрібні обчислення. Завдання відображення розподілені по комп'ютерах, так що кожен оператор може отримувати необхідні дані (рис. 10.6).

Оскільки завдання відображення обробляються локально і лише запити до даних надходять на центральний сервер, така система значно покращує продуктивність і гнучкість.

Іншою перевагою такої системи є необхідність наявності тільки одного принтера, доступного з будь-якого комп'ютера в мережі. Додаткові оператори можуть бути легко підключені до системи. Менеджерам і групі якості може бути надано доступ до даних виробничого процесу (рис. 10.7).

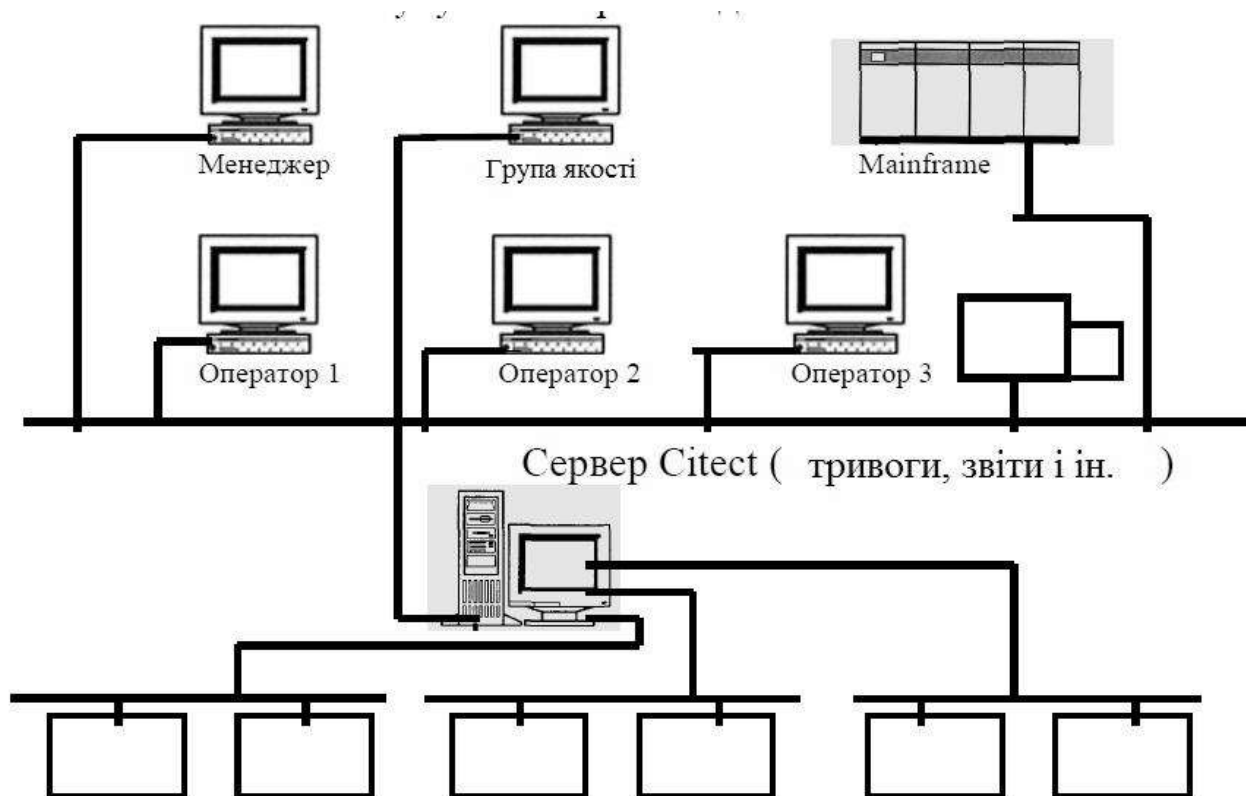


Рис. 10.7 Багаторівнева система по пароллю

Багаторівнева система по пароллю обмежує доступ до даних і захищає обладнання підприємства від несанкціонованих звернень. Наприклад, кожному оператору може бути надано доступ лише до визначених об'єктів на підприємстві; група якості може мати доступ, обмежений тільки читанням даних на всьому підприємстві, в той час як начальник зміни володіє необмеженим доступом. Зв'язок з великим комп'ютером може підтримувати завантаження регламентів і управління випуском продукції, а, в свою чергу, звіти і журнали тривог можуть надходити назад у відділи планування і управління.

Те, що починалося як невеликий додаток, перетворилося в середню за розмірами систему, не потребуючи зміни ні обладнання, ні ПЗ. Інвестиції були збережені на кожній стадії розвитку. Для додатків великого розміру, як правило, потрібен окремий сервер для задач, звітів, тривог і графіків. Як доповнення може бути використаний файловий сервер для зберігання конфігурації БД і загального ПЗ (рис.10.8).

Не всі АСУ (як показано на рис.10.8) настільки великі, але Citect може ефективно використовуватися в системі будь-якого розміру, функціонально розширюючись у міру зростання підприємства. Можливість простого додавання нової апаратури і подальшого розподілу обробки в Citect для Windows є поворотною точкою в історії розвитку АСУТП.

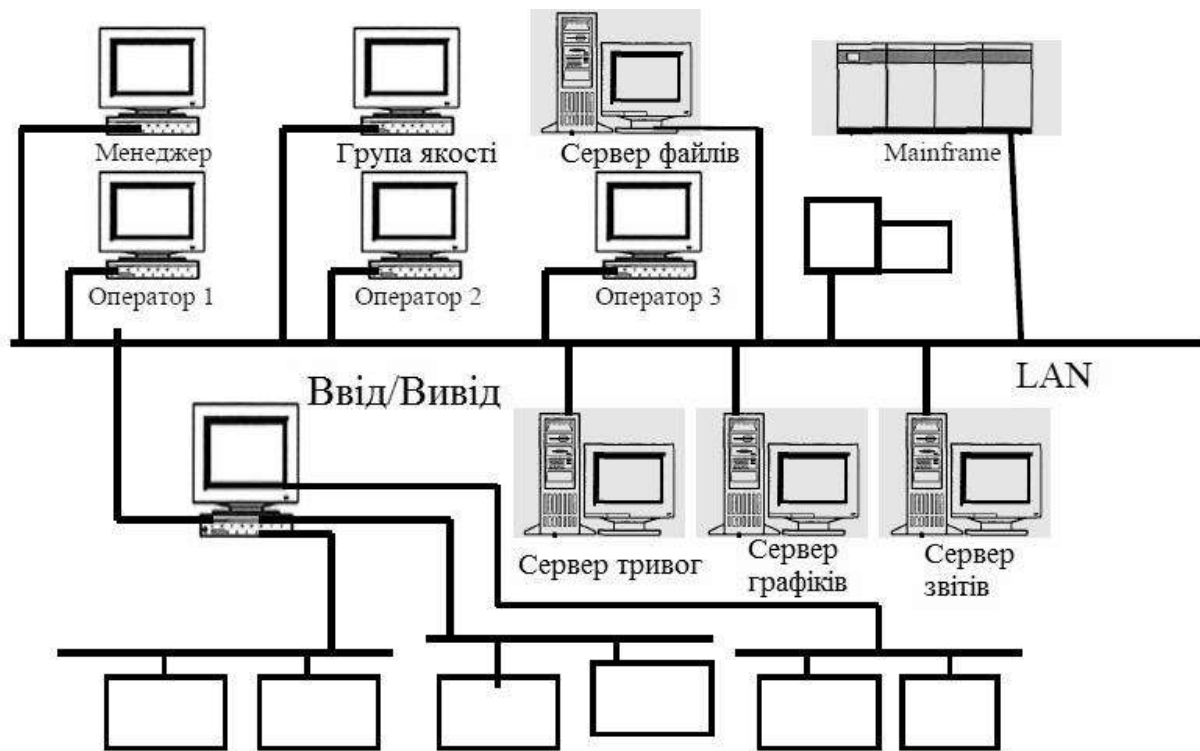


Рис 10.8 Система з окремими серверами