

**МЕТОДИ І ТЕХНОЛОГІЇ ЗНИЖЕННЯ ПІКОВИХ НАВАНТАЖЕНЬ
У КОРПОРАТИВНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ
ІЗ ЦЕНТРАЛІЗОВАНИМ ЗБЕРЕЖЕННЯМ ДАНИХ**

В.Й. СІМАШКО

Описано пікові навантаження, які виникають у великих корпоративних інформаційних системах (КІС) із централізованим збереженням даних. Запропоновано методи і технології, які дозволяють знизити або ліквідувати пікові навантаження на сервери та на мережу системи.

ВСТУП

Сучасні тенденції розвитку корпоративних інформаційних систем (КІС) демонструють бурхливе зростання обсягів інформації, яка зберігається, обробляється і передається по мережі КІС. Із цим також пов'язане стрімке зростання кількості персональних комп'ютерів, які входять у корпоративну мережу передачі даних (КМПД) підприємства [1]. У великих КІС одним із сучасних напрямів розвитку є тенденція до централізації зберігання та обробки даних [1]. Насамперед це відбувається в білінгових системах, у системах економічного призначення (банківських, планово-фінансових і системах контролінгу) та деяких інших КІС великих підприємств. У процесі експлуатації таких КІС виникає низка проблем, зокрема:

- великі обсяги бази даних (БД) корпорації;
- значні обсяги обчислень і пов'язана з цим недостатня потужність як робочих місць користувачів, так і серверів;
- великий мережевий трафік між центральним серверним сегментом КМПД та робочими місцями користувачів;
- великий обсяг робіт з підтримки функціональності робочих місць (контроль та поновлення версій драйверів, додатків для користування, шаблонів документів, антивірусних баз тощо).

ПОСТАНОВКА ЗАДАЧІ

Дослідження та публікації з питань побудови КІС великих підприємств [1, 2, 3] недостатньо висвітлюють шляхи вирішення цих проблем. В якості універсального засобу їх вирішення майже завжди пропонується [1, 2] перехід до трьохрівневої структури обробки даних у корпоративній інформаційній системі: сервер БД, сервер додатків і клієнтське робоче місце. Проте практика показує, що така схема повністю не усуває жодної з описаних вище проблем, а лише частково знижує обсяг робіт з підтримки функціональності клієнтських робочих місць.

Описані вище проблеми періодично проявляються у різкому зниженні продуктивності роботи КІС у періоди так званих пікових навантажень: під час підготовки місячних, квартальних і річних звітів; при закритті розрахункових періодів; у випадку розповсюдження на комп'ютери корпорації об'ємних поновлень (нових версій додатків, документів, антивірусних баз тощо); під час великих обсягах пакетного (файлового) завантаження інформації тощо. Надмірне пікове навантаження може призвести до часткової чи повної втрати дієспроможності інформаційних систем підприємства, наслідки цього можуть бути досить відчутними для корпорації. Вирішення цього питання вимагає комплексного підходу.

Мета роботи — описати деякі методи і технології, які дозволяють суттєво знизити пікові навантаження на КІС із централізованим збереженням та обробкою даних, і таким чином підвищити стійкість і надійність функціонування такої КІС.

ПРОБЛЕМИ, ЩО ВИНИКАЮТЬ ПІД ЧАС ЦЕНТРАЛІЗАЦІЇ ОБРОБКИ ДАНИХ, МЕТОДИ І ТЕХНОЛОГІЇ ЇХ ВИРІШЕННЯ

У великих КІС централізація обробки даних полягає не лише в перенесенні всіх існуючих БД в єдиний серверний сегмент корпорації або в єдиний центр обробки, а й у зберіганні інформації. Централізація зазвичай передбачає:

- Створення єдиної БД для всіх підсистем КІС. Зазвичай єдина БД створюється на основі потужної промислової реляційної системи керування БД (РСКБД), такої як Oracle, MS SQL, DB2 тощо. Недаремно одним із важливих якісних показників РСКБД є продуктивність роботи БД при дуже великих обсягах збереженої інформації.

- Програмування безпосередньо в БД якомога більшої кількості бізнес-правил корпорації, а також контроль за дотриманням бізнес-логіки безпосередньо на рівні процедур та функцій БД. Ця тенденція в сучасних КІС спонукала розробників промислових РСКБД за останнє десятиріччя суттєво розширити функціональні можливості процедурних мов програмування БД (таких як — Oracle PL/SQL, MS Tansact-SQL тощо).

- Перенесення на сервер додатків функцій з виконання обчислень, звітів, поновленню версій додатків, збереженню шаблонів документів, виконаних звітів і т.ін.

Не менш важливими факторами, які спонукають до такої централізації — економічні, а саме: велика вартість утримання висококваліфікованих адміністраторів, супроводу та підтримки промислових БД; зведення до мінімуму ймовірності втрати даних. Але централізація збереження та обробки інформації не проходить без наслідків для корпорації. Негативні наслідки цього описано нижче.

Ще донедавна використання реляційних БД визначало застосування технології «клієнт-сервер» у її класичному розумінні. «Класична» технологія «клієнт-сервер» полягає в наступному. Клієнтський додаток, написаний зазвичай на мові програмування високого рівня, формує SQL-запити. Сервер БД виконує ці запити та повертає додатку результати їх виконання.

Отримані з сервера дані обробляються клієнтським додатком та зберігаються у вигляді файлів, звітів, документів тощо. Отже, під час застосування класичної технології «клієнт-сервер» основний обсяг роботи (з обробки даних, з реалізації бізнес-правил та бізнес-процесів) здійснювався клієнтським додатком. У сучасних умовах різкого зростання обсягів інформації, яка обробляється в КІС, до клієнтського робочого місця з БД передається все більший обсяг інформації, а додаток виконує все більший обсяг обчислень. Унаслідок цього корпорація має постійно вкладати кошти в розвиток пропускну здатності мережі та в підвищення потужності клієнтських комп'ютерів. Такою є плата за централізацію зберігання інформації.

До двох щойно описаних недоліків «класичної» технології «клієнт-сервер» (зростаючий обсяг обчислень, що здійснюють клієнтські комп'ютери; зростаючий мережевий трафік між сервером і клієнтськими комп'ютерами), слід додати третій суттєвий недолік, пов'язаний із все більшою популярністю WEB-інтерфейсів у КІС, які йдуть на заміну традиційним додаткам робочого столу — проблематична реалізація складних бізнес-правил та бізнес-процесів за допомогою WEB-додатків через обмежені можливості мов WEB-програмування.

Для усунення описаних вище недоліків виробники сучасних реляційних БД інтенсивно розвивають і вдосконалюють такі дві технології:

- інтерфейси БД із WEB-додатками та WEB-орієнтованими мовами програмування;
- використання збережених у БД процедур та функцій (stored procedures and functions).

З метою сприяння ефективному написанню клієнтських WEB-додатків для роботи з реляційними БД виробники розширяють підтримку JAVA та NET. Та все ж вирішальну роль у сучасних КІС відіграє розвиток і вдосконалення процедурних мов, на яких програмуються реляційні БД та створюються збережені в БД процедури, функції, тригери та інші об'єкти.

Розглянемо детальніше використання збережених у БД процедур. Згідно зі своєю назвою, процедура зберігається в БД, і незалежно від способу її ініціалізації (виклику) виконується безпосередньо сервером БД. Можливості сучасних процедурних мов БД такі, що результатом виконання процедури може бути не лише традиційний набір вихідних параметрів, але й: масив даних будь-яких типів; модифікація інформації в БД; створення, зміна та знищення будь-яких об'єктів БД; XML-документ тощо. Якщо використати можливість процедури зі збереження результатів її виконання в БД, то мережевий трафік між сервером та клієнтським додатком, і навантаження на клієнтське робоче місце зводяться до мінімуму. Клієнтському додатку необхідно сформулювати та відправити на сервер SQL-запит виконання збереженої процедури, дочекатися результату виконання та відобразити цей результат на дисплеї. Якщо при цьому результат виконання великого обсягу (великий документ, звіт у вигляді значного масиву даних тощо), доцільно зберегти результат у БД, щоб дати можливість користувачеві отримати результат пізніше, у зручний для нього час. Така технологія особливо актуальна для клієнтських робочих місць КІС, віддалених від сервера, які працюють по виділених лініях, або в режимі дозвону (dial-up) по телефонних лініях — так звані «тонкі клієнти». Працівникам віддалених офісів не потрібно очікувати

звільнення каналу зв'язку та робити кілька спроб, щоб вдало виконати складний розрахунок чи звіт, збережена процедура гарантовано буде виконана з першої спроби, а от із переглядом результату, можливо, доведеться зачекати.

Принципово важливо, що під час застосування цієї технології вся бізнес-логіка (алгоритми розрахунків, бізнес-правила тощо) реалізовані в БД, а не в клієнтському додатку. Тип додатку (WEB-додаток або додаток робочого столу) та мова програмування, на якому він написаний, відходять на другорядний план. За клієнтським додатком залишається мінімум функцій, а саме:

- введення даних користувачем;
- формування та відправка на сервер SQL-запитів;
- відображення для користувача даних, отриманих із сервера.

Із такими завданнями впорається навіть мова програмування із дуже обмеженими можливостями.

Також суттєво те, що під час застосування цієї технології будь-яка зміна бізнес-логіки, а також виправлення помилок у програмному забезпеченні (ПЗ) — реалізуються в БД, а не в клієнтському додатку. Тобто, у більшості випадків для заміни версії ПЗ достатньо замінити збережену процедуру (частіше — пакет збережених процедур) в єдиному місці — безпосередньо на сервері в БД. Для великих КІС це в сотні разів швидше, дешевше і надійніше, ніж заміна модулів, що виконуються та бібліотек на кожному комп'ютері корпоративної мережі.

Отже, під час застосування збережених процедур суттєво спрощуються (а отже, здешевлюються) клієнтські додатки; знижується навантаження на клієнтські комп'ютери і на корпоративну мережу (саме тому, знижуються капіталовкладення в обчислювальну техніку і в мережеве обладнання); спрощується контроль і заміна версій ПЗ КІС та виникає низка інших переваг. Звичайно, такі суттєві переваги не даються дарма. Натомість виникають дві інші значніші проблеми:

- різке зростання складності програмування БД;
- різке зростання навантаження на сервер БД.

Іншими словами, можна сказати, що «дешевші комп'ютери та мережа — дорожчі за сервер і БД».

Звичайно, розглядати можливі варіанти побудови структурних схем КІС та приймати рішення про використання тієї чи іншої технології обробки інформації необхідно окремо в кожному конкретному випадку. Проте все більше та більше КІС будуються саме за технологією з основним акцентом на серверне ПЗ [1, 2]. Очевидно, у багатьох випадках це економічно доцільно. Про такі світові тенденції свідчить також велика увага, яку розробники сучасних реляційних систем керування БД приділяють розвитку та вдосконаленню процедурних мов програмування БД. Безумовний лідер з потужності й можливостей — мова Oracle PL/SQL. Засобами PL/SQL можна реалізувати практично будь-який алгоритм, бізнес-правило чи обмеження цілісності. При цьому застосовуються такі об'єкти БД, як процедури, функції, пакети процедур і функцій, тригери тощо. Також БД Oracle мають потужну підтримку мови JAVA. Відразу за Oracle процедурні мови і підтримку

JAVA розвивають виробники практично всіх РСКБД. Інший потужний виробник — фірма Microsoft, яка для свого MS SQL-сервера розробляє процедурну мову Transact-SQL. Для розширення можливостей збережені процедури цієї РСКБД можуть навіть включати фрагменти коду на мові С.

Підсумовуючи, можна зробити висновок, що сучасні процедурні мови програмування БД мають усі необхідні можливості для написання збережених процедур будь-якої складності.

Друга проблема — різке зростання навантаження на сервер БД. Вона частково вирішується нарощуванням потужності апаратної частини сервера, а частково — застосуванням кластерної структури серверного сегмента. Проте потужність сервера все одно не буде безмежною. Технологія, орієнтована на збережені процедури, передбачає виконання лівової частки обчислень саме сервером БД. Сервер додатків не виконує обчислень, за ним залишається лише функція форматування результатів обчислень у вигляді документів, графіків тощо. Тому в моменти пікових навантажень на БД (під час підготовки періодичних звітів, у зв'язку із закриттям розрахункових періодів, під час виконання масових розрахунків або пакетних завантажень тощо) сервер буде працювати на межі можливостей. Навантаження на процесори, дискову систему, пам'ять сервера БД так чи інакше досягатимемо 100 %. І саме це є насправді основною проблемою, на яку рано чи пізно натрапляють адміністратори корпоративної БД та КІС із централізованим збереженням даних. Ця проблема інколи може виникати одразу після введення КІС у промислову експлуатацію. Але зазвичай ця проблема виникає через деякий час, при досягненні «критичних» позначок: критичного обсягу БД, критичної кількості користувачів, критичного набору функціональних модулів.

Дефіцит потужності сервера БД виявляється непостійно, а в моменти пікових навантажень, коли занадто багато користувачів починають одночасно виконувати складні або довготривалі розрахунки, звіти тощо. Усі розрахунки в такі періоди уповільнюються в десятки разів. Починають виникати ситуації, коли сервер БД не може забезпечити цілісність даних на досить тривалому проміжку часу — через модифікацію інформації іншими користувачами під час виконання розрахунку або звіту. У цьому випадку розрахунок (звіт) завершується помилкою, тому його необхідно повторювати знову.

Нестача потужності сервера БД відображається на роботі всіх користувачів системи, а не лише тих, хто «замовляє» складні звіти. Зокрема, оператори КІС, які виконують короткочасні інтерактивні транзакції (обслуговують клієнтів корпорації, працюють із одиночними документами чи проводками тощо) помічають, що транзакції, які зазвичай виконувалися долі секунд, у моменти пікових навантажень виконуються хвилини, або завершуються помилкою, або ж взагалі ніколи не закінчуються («зависають»). Це призводить до значних втрат робочого часу персоналу. Також це може призвести до явних збитків — коли йдеться про прийом готівкових платежів із використанням касових апаратів, або про реєстрацію в БД документів тих осіб, які бажають стати клієнтами корпорації і т.ін.

Перш за все, ця проблема має вирішуватись оптимізацією серверної частини КІС: використанням сучасних можливостей мови SQL та процедур-

ної мови, індексацією даних, налаштуванням параметрів БД, усуненням причин взаємного блокування сесій користувачів тощо. Проте далеко не завжди вдається уникнути дефіциту потужності сервера БД навіть при якісно спроектованій і налаштованій БД, а також грамотному написанні SQL-запитів і збережених процедур. Яким би потужним не був сервер, рано чи пізно виникне ситуація, коли він занадто повільно оброблятиме зростаючі потреби користувачів.

Враховуючи зазначене вище, зрозуміло, що для суттєвого покращення роботи сервера БД та КІС у цілому необхідно за можливістю рівномірно розподілити в часі виконання довготривалих розрахунків та складних звітів. Для цього існує кілька методів. Найпростіший і найдешевший метод (оскільки він не потребує доробки ПЗ системи) — організаційний. Він полягає в інвентаризації прав доступу до системи. Практика показує, що чим більше підприємство, тим більше диференційовані його працівники за рівнем кваліфікації. Особливо це критично для інженерного персоналу, задіяного у важливих технологічних процесах КІС. Досвід показує, навіть приблизно за однакового рівня освіти та досвіду роботи технологічна дисципліна різних виконавців дуже відрізняється. На жаль, не всі відповідальні особи до кінця розуміють, які процеси відбуваються в інформаційній системі, яка послідовність дій необхідна для ефективної роботи, результати яких розрахунків впливають на які звіти тощо. Тому згодом керівники корпорації починають розуміти, що до всіх важливих процесів (розрахунків, звітів, пакетних завантажень тощо) доступ необхідно жорстко обмежувати. Перевірка на практиці показала, що на великих підприємствах, значно розділених у просторі, але із централізованими БД, значно ефективніше централізувати також і групу фахівців з експлуатації КІС, залишивши операторам на місцях право лише на інтерактивні операції по роботі з клієнтами корпорації та із одиночними провідками документів. У такому випадку значно легше контролювати технологічні процеси та розподілити графік їх виконання в часі, знизивши тим самим пікові навантаження на систему.

Організаційні заходи можуть принести бажане зменшення пікових навантажень на сервер БД, але розмір виграшу буде різним у різних конкретних КІС. Зменшення навантаження може бути тимчасовим (враховуючи зростання розмірів БД, кількості користувачів та функціональних модулів КІС), або взагалі невідчутним (якщо специфіка підприємства така, що більшість звітів мають виконувати працівники віддалених підрозділів корпорації).

Тому кардинально вирішити проблему можна лише завдяки певним доробкам ПЗ КІС. Ідея полягає в поєднанні технологій черги, планувальника процесів та розкладу виконання завдань.

Пропонується така технологія роботи інформаційної системи. Процедура БД (розрахунок, звіт тощо) не виконується одразу під час запиту на ініціалізацію її виконання, а ставиться у чергу завдань сервера БД. Для цього назва процедури, код користувача, який замовив виконання та її вхідні параметри та всі вихідні дані виконання процедури мають бути збережені у БД. Якщо результатом виконання є складний документ, то він зберігається в БД у вигляді XML, якщо текстовий звіт або розрахунок — то збереження результатів відбувається у звичайних реляційних таблицях.

Із таких запитів автоматично формується одна, або декілька черг завдань БД — у залежності від параметра, заданого адміністратором КІС. Оптимальна кількість окремих черг завдань залежить від кількох факторів: апаратних параметрів сервера БД, кількості вузлів у кластері, можливістю підтримки сервером РСКБД багатопроцесорності, масштабами та обсягами КІС тощо. Цей параметр підбирається в кожному конкретному випадку під час експлуатації системи.

Кожному конкретному завданню має бути присвоєний його унікальний номер — ідентифікатор. Працівнику корпорації, який ініціював той чи інший звіт або розрахунок, у клієнтський додаток передається посилання (ідентифікатор завдання), по якому він може відкривати результат у зручний для нього час. Якщо декілька користувачів замовили один і той же звіт або розрахунок (з однаковими вхідними параметрами), завдання буде виконане лише один раз, а всім «замовникам» буде передане одне і те ж посилання. Кожне завдання має свій статус: «очікування», «виконання», «виконано успішно», «виконано із помилками». Статус визначає, які дії користувач або адміністратор системи може виконувати із завданням.

Виконання завдань по черзі можна реалізувати кількома способами. Важливу роль у цьому відіграє вибір системи керування БД. Якщо підприємство може придбати та здійснити підтримку сучасної промислової РСКБД, такої як Oracle, то воно отримує у своє розпорядження досить зручний інструмент завдань БД (JOB-и). У цьому випадку виконання завдань по черзі реалізується саме на завданнях БД. Якщо масштаби та вартість КІС такі, що визначають застосування простіших серверів РСКБД, для реалізації черги необхідне написання нескладного додатку. Цей додаток (так званий «монітор черги завдань КІС») виконується на сервері БД або на сервері додатків, періодично перечитує таблицю черги, і за відсутності завдань зі статусом «виконання» та за наявності в ній завдань зі статусом «очікування» ініціює виконання такого завдання.

Алгоритм формування черги завдань також може бути реалізований кількома способами. Найпростіший — «по мірі надходження». Постанова завдання в чергу відбувається завжди «у кінець» тієї черги, яка містить найменше завдань. Такий алгоритм прийнятний для малих і середніх КІС, де пікові навантаження виникають досить рідко.

Для великих систем необхідно застосовувати складніші алгоритми:

По-перше, для кожного конкретного типу завдання в БД має фіксуватися рейтинг (або пріоритет), наприклад: «звичайне», «важливе» і «негайне». У залежності від цього визначається, у яке місце в черзі буде поставлене те чи інше завдання. Доцільно застосувати таке правило: завдання ставиться в ту чергу, в якій є менше завдань із рівним або вищим пріоритетом. Місце завдання в обраній черзі визначається так: наступне після останнього завдання із рівним або вищим пріоритетом.

По-друге, для кожного конкретного типу завдання в БД можна зафіксувати часові межі (інтервал годин доби від ... до ...), в які або не можна це завдання виконувати, або завдання можна виконувати виключно в заданий часовий період. Наприклад, можливі складні розрахункові процеси, для яких є бажаним незмінний стан довідників системи протягом усього виконання розрахунку, або виконання яких у робочий час унеможлиблює нормальну

роботу інтерактивних користувачів КІС. Такі завдання не можна виконувати в робочий час, коли проводиться робота з клієнтами корпорації. Інший приклад: заміна версій серверного та клієнтського ПЗ КІС може проводитися тільки у тому випадку, коли в системі не працюють жодні користувачі та не проводяться жодні розрахунки. Такі завдання необхідно виконувати в суворо визначений, зазвичай у нічний час. У залежності від «заборонених» або «дозволенних» часових інтервалів завдання може очікувати своєї черги (точніше, «свого часу»), навіть якщо воно розташоване першим у черзі.

По-третє, алгоритм постановки у чергу може бути адаптивним. Для його реалізації по кожному типу завдання у БД накопичується й усереднюється статистична інформація (наприклад, середня тривалість успішного виконання). У цьому випадку завдання з однаковим пріоритетом, які виконуються миттєво, доцільно ставити у чергу перед завданнями, які виконуються кілька годин (особливо, якщо прогноз показує, що результат цього довготривалого завдання користувач зможе переглянути лише наступного дня).

Також слід зазначити способи, якими можливо формувати чергу. Програмний модуль, який реалізує описані в попередньому абзаці алгоритми, можна вбудувати в клієнтський додаток, яким користувачі системи ініціюють виконання тих чи інших розрахунків, пакетних завантажень чи звітів. Проте більш природно реалізувати цю логіку як процедуру або пакет процедур, збережений у БД, а клієнтський додаток має працювати із цим пакетом. Другий варіант має перевагу з точки зору можливості оперативної зміни алгоритму формування черги під час виникнення критичних ситуацій.

Чим складніша та об'ємніша інформаційна система, і чим відчутніша в ній проблема нестачі ресурсів сервера, тим важливішу роль відіграє можливість застосування технології черги завдань. У будь-якому випадку універсальних рекомендацій з алгоритму формування черги немає. Для кожної конкретної КІС замовником і розробником мають узгоджуватися свої специфічні алгоритми. Також слід зазначити, що для гнучкої експлуатації системи її адміністратори завжди повинні мати можливість ручного впливу на чергу завдань. Тобто, адміністратор повинен мати змогу видалити завдання, перемістити його в черзі, змінити пріоритет тощо.

Під час використання технології черги завдань сервером БД одночасно може виконуватися лише така кількість завдань (розрахунків, звітів, об'ємних пакетних завантажень), яка відповідає кількості окремих черг у цій системі. Практика показує, що під час оптимального підбору кількості черг та завдяки вмілому написанню алгоритму формування черги, пікові навантаження на сервер БД можна звести до мінімуму або взагалі до нуля. Це означає, що навіть у найнапруженіші для КІС періоди експлуатації, ресурси процесорного часу, оперативної пам'яті та дискової системи не будуть наближатися до 100% зайнятості. Практично це призведе до значно швидшої, а значить, ефективнішої роботи інформаційної системи зокрема, та підприємства в цілому.

Описані вище підходи стосувалися технологій і методів, які знімають пікові навантаження як із серверів КІС, так із робочих станцій. Далі зупинимося детальніше на методах зменшення навантажень на КМПД.

Як зазначено вище, зростання обсягів даних, які обробляються і зберігаються в КІС, а також тенденція до централізації збереження інформації призводять до різкого зростання трафіку між серверним сегментом корпорації та кожним комп'ютером КМПД. Для прикладу розглянемо типові завдання, що виконуються звичайним клієнтським місцем КІС на початку кожного дня:

- після увімкнення комп'ютера операційній системі (ОС) необхідно зареєструватися в домені, отримати з сервера домену політики безпеки, пакети поновлень ОС, та, можливо, підключитися через мережу до загальнодоступних ресурсів;
- з антивірусного сервера корпорації необхідно завантажити зміни в антивірусних базах, які надійшли від розробника за минулу добу. Досить часто корпоративні версії антивірусів вимагають щоденного завантаження на комп'ютер не останніх змін, а повних антивірусних баз;
- із серверів КІС необхідно завантажити для клієнтських програм нові версії файлів: модулі, що виконуються, динамічні бібліотеки, шаблони документів тощо.

І це далеко не повний перелік обов'язкових процедур, кожна з яких зводиться до завантаження по мережі певного обсягу даних. Помноживши сумарний обсяг на кількість комп'ютерів, наприклад, в одному підрозділі підприємства, можна оцінити щоденний технологічний трафік між підрозділом та серверним сегментом підприємства. Такий пік навантаження спостерігається на початку кожного робочого дня — ще до того, як працівники розпочали виконувати роботу. Також пікові навантаження можуть виникати і в інший час, коли на сервери підприємства надходять об'ємні поновлення для ОС або антивірусних баз, нові версії клієнтських додатків тощо. Якщо цей підрозділ пов'язаний із серверами корпорації слабким каналом зв'язку, то зрозуміло, що на час пікових навантажень мережі корисна пропускна здатність каналу різко знижується. Крім втрат робочого часу, це призводить і до фінансових втрат, якщо канали зв'язку не є власністю підприємства і за них потрібно платити. Або, якщо канали зв'язку належать підприємству, але вони використовуються також і для надання послуг стороннім користувачам на комерційній основі.

Оскільки постійно нарощувати пропускну здатність КМПД неможливо, необхідно вжити заходів до зниження трафіку. Один з ефективних методів досягнення цієї мети — побудова ієрархічної структури КМПД.

Для ефективної побудови ієрархічної структури необхідно:

- здійснити поділ КІС підприємства на відносно автономні сегменти за їх територіальним розташуванням;
- здійснити поділ інформації на таку, яка важлива для підприємства в цілому, і тому має зберігатися централізовано, і таку, яка важлива для кожного робочого місця, і має бути щоденно та гарантовано доставлена на кожен комп'ютер.

Після такого поділу кожен відносно автономний сегмент КМПД має отримати свій окремий локальний сервер структурного підрозділу. Назвемо його «сервером поновлень структурного підрозділу», або ж «сервером поновлень».

Немає жодного сенсу завантажувати напряму з центрального сервера корпорації окремо на кожне робоче місце, скажімо, антивірусні бази розміром у десятки мегабайтів, або нові версії файлів для клієнтських програм. Це породжує невиправдано високі пікові обсяги мережевого трафіку. Оскільки такого роду інформація однакова для абсолютно всіх робочих місць, то достатньо завантажити інформацію на сервер поновлень структурного підрозділу, а всі робочі місця отримають її вже по локальній мережі. Це знизить технологічний трафік поновлення версій між серверним сегментом корпорації та конкретним структурним підрозділом рівно в стільки разів, скільки комп'ютерів є у локальній мережі цього структурного підрозділу.

Практика показує, що у якості сервера поновлень можна використати звичайний персональний комп'ютер. Єдина вимога — безперебійне живлення та цілодобовий режим роботи. Зазвичай на сервер поновлень, у першу чергу, інсталиують корпоративний антивірусний сервер, налаштовують його таким чином, щоб поновлення антивірусних баз надходили на нього із центрального антивірусного сервера. Антивірусні клієнтські програми всіх робочих місць підрозділу налаштовують таким чином, щоб поновлення антивірусних баз надходили на них із сервера поновлень цього підрозділу. Також на сервері поновлень інсталиуються FTP та HTTP-сервери. Вони використовуються всіма інформаційними системами як проміжні «перевалочні» вузли для надходження в локальну мережу підрозділу нових версій клієнтських програм: модулів, що виконуються, динамічних бібліотек, шаблонів документів тощо. Також FTP та HTTP-сервери використовуються для внутрішнього обміну файлами між комп'ютерами локальної мережі структурного підрозділу.

Велика корпорація, яка охоплює значну територію, повинна мати кілька рівнів ієрархії КМПД. Наприклад підприємство, яке охоплює масштаби цілого регіону чи країни, матиме такі рівні ієрархії:

- центральний серверний сегмент корпорації;
- сервери поновлень обласних центрів, кожен із яких підпорядковується центральному серверному сегменту;
- сервери поновлень районних центрів, кожен із яких підпорядковується своєму серверу поновлень обласного центру;
- звичайні робочі місця, кожне із яких підпорядковується своєму серверу поновлень районного або обласного центру.

Нові версії всіх файлів (антивірусних баз, клієнтських програм, шаблонів документів тощо) автоматично розповсюджуються «зверху вниз» — від центрального серверного сегмента корпорації до сервера поновлень найнижчого рівня. Завантаження нових версій файлів із сервера поновлень на кожен комп'ютер КМПД ініціюється відповідними клієнтськими додатками під час їх запуску.

Така ієрархічна структура мережі передачі даних забезпечує мінімізацію технологічного трафіку по розповсюдженню нових версій файлів, а отже, зменшення прямих затрат на КМПД. Не менш важливо, що внаслідок зникнення пікових навантажень мережа, а значить і КІС у цілому, починає працювати значно стабільніше та надійніше. Платою за ці переваги є необхідність підтримки ієрархічної структури серверів поновлень. Проте

за сучасних можливостей віддаленого адміністрування ці додаткові затрати можуть бути значно менші, ніж отриманий вигреш.

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ЗАПРОПОНОВАНИХ ТЕХНОЛОГІЙ

Для оцінювання ефективності запропонованих методів було здійснено їх експериментальну перевірку на двох промислових інформаційних системах. Аналітичним способом оцінити очікуваний вигреш від впровадження описаних вище технологій складно. Розрахунок навантаження на серверний сегмент підприємства — це складна багатопараметрична задача, яка майже не піддається формалізації, а спроби вивести деякі аналітичні залежності носять емпіричний характер [2]. Було проведено експеримент на двох промислових системах:

- Дворівнева («класична» — «клієнт-сервер») інформаційна система. Сервер БД: ОС Windows 2003 Standart Edition Service Pack 2, РСКБД Oracle 9.2.0.7, апаратна платформа Intel E7505, 2 процесори Intel Xeon 2.4 ГГц, ОЗУ 2 ГБ, дисковий масив на базі SATA RAID контролера, обсяг БД 80 ГБ, кількість зареєстрованих користувачів приблизно 250.

- Трирівнева інформаційна система. Сервер БД: ОС Windows 2003 SE SP2, Oracle 10.2.0.4, апаратна платформа Intel E7525, 4 процесори Intel Xeon 3.4 ГГц, ОЗУ 4 ГБ, дисковий масив на SCSI RAID контролері, обсяг БД 700 ГБ, кількість зареєстрованих користувачів приблизно 450.

В обох системах було реалізовано найпростіший варіант організації черги: без пріоритету завдань, без часових інтервалів виконання завдань та без адаптивних алгоритмів (рис. 1). Згідно із цією блок-схемою, кожне завдання отримує свій незмінний ідентифікатор, а також має поточний статус (від «очікування виконання» до «виконано успішно»). Користувач, який в інтерактивному режимі замовляє той чи інший звіт чи ініціює процес, одразу отримує значення ідентифікатора. По цьому ідентифікатору в будь-який час можна переглянути не лише всі параметри завдання, а й його поточний статус, і якщо статус рівний «виконано успішно», отримати результат виконання. Спосіб отримання результатів завдання (документи, файли, інтерактивний перегляд) залежить від назви (тобто від типу) завдання. Крім того, користувач має можливість вибрати з БД ідентифікатори і назви всіх завдань, які цей користувач ініціював будь-коли.

Виконання завдань по черзі реалізує проста програма на сервері додатків, яка виконується постійно, шукає в кожній черзі завдання зі статусом «очікування», і стартує їх одне за одним.

Експериментальні дані збиралися таким чином. Засобами Oracle в БД постійно фіксувалися всі збої, що виникали (неможливість розпочати сесію, невиконання запиту тощо) і причини цих збоїв (коди помилок Oracle); а тричі на хвилину реєструвалася кількість активних сесій користувачів. Засобами ОС також тричі на хвилину у файлах на сервері БД реєструвалися: середній для всіх процесорів відсоток зайнятості, обсяг вільної оперативної пам'яті, середній для всіх жорстких дисків відсоток активності.

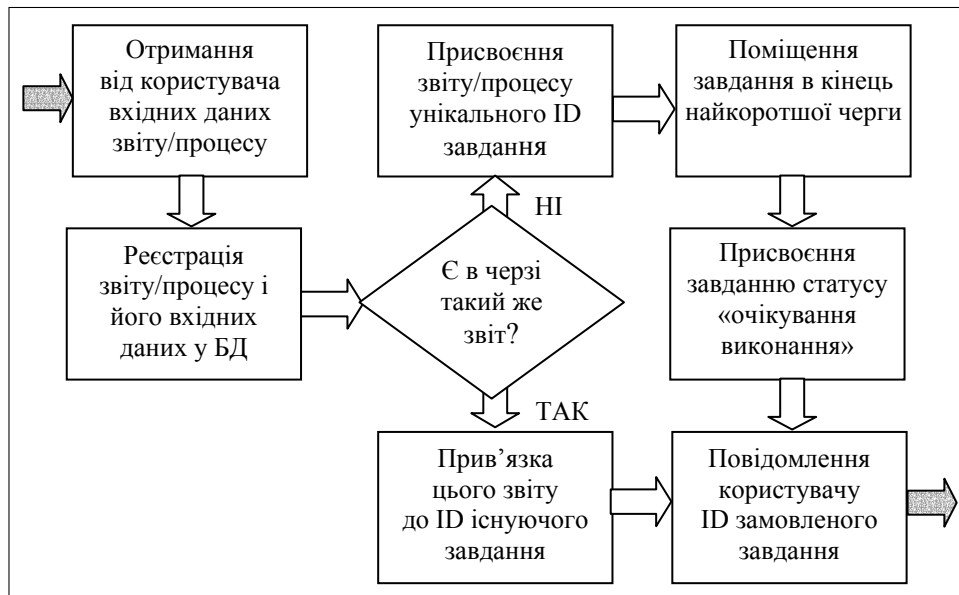


Рис. 1. Блок-схема

Накопичення даних було проведене таким чином. Протягом двох місяців реєструвалися та зберігалися всі експериментальні дані під час промислової експлуатації систем без організації черги завдань. Потім протягом місяця в обох системах було проведено впровадження описаної вище технології черги. На цьому етапі дослідним шляхом проведено вибір оптимальних налаштувань для обох систем, але експериментальні дані на цьому етапі не реєструвалися і не зберігалися. Після організації черги завдань знову реєструвалися і зберігалися всі експериментальні дані, також протягом двох місяців.

По завершенню збору інформації всі дані вимірів були завантажені в БД, синхронізовані в часі, нормовані (кількість збоїв — до кількості сесій, вільна пам'ять — до її загального обсягу тощо). Після нормування всі дані було усереднено по часу і по двох інформаційних системах. Для усереднення було використано не всі календарні дні, а лише ті, на яких спостерігалось суттєве навантаження на БД: крім вихідних та періоду «затишшя». У цих конкретних системах «затишшя» припадає на період із 10 по 25 число місяця, у ці дні складних звітів та масових операцій майже немає. Після усереднення будувалися графіки і діаграми, розраховувалися інтегральні показники, і на основі їх аналізу робилися висновки.

Перед початком збору даних для системи № 1 була підібрана оптимальна кількість черг, рівна 3, а для системи № 2 — рівна 7 окремих черг. За більшої кількості черг зниження пікових навантажень не відбувається. Таким чином виявлено залежність оптимальної кількості черг від кількості процесорів (тобто, ядер процесорів): у системі № 1 — 4 ядра, у № 2 — 8 ядер. Очевидно, що одне ядро обслуговує потреби ОС, а всі інші — вільні для обслуговування БД.

Графіки залежності вільної пам'яті ОС виявилися неінформативними, і тому в цій роботі не наводяться. Це пов'язано із особливостями ОС Windows 2003 Standart Edition, яка не виділяє під кожен окремий процес (та-

кож і під Oracle) оперативну пам'ять, більшу певної фіксованої величини. Нестача пам'яті виявляється не на рівні системи, а на рівні самого сервера Oracle, і стандартними засобами цю нестачу виміряти не вдалося.

Показником нестачі пам'яті в межах квоти, віділеної ОС серверу Oracle, є графік «Кількість збоїв» (рис. 2). Йдеться про збої БД, які (як показав аналіз кодів помилок сервера Oracle) у 90 % випадків виникали через неможливість сервером виділити область пам'яті під поточну сесію або для виконання запиту. На графіку «Кількість збоїв» нормована до поточної кількості активних сесій користувачів, і в деякі моменти цей показник досягав 30 %, тобто кожна третя сесія завершувалася аварійно через нестачу пам'яті. Можна побачити, що до впровадження черги середньостатистична частота збоїв сервера БД у моменти найбільшого навантаження рівна приблизно 10...12 %, а після впровадження черги збої припинилися взагалі. За допомогою черги завдань проблему нестачі пам'яті для сервера Oracle було повністю вирішено.

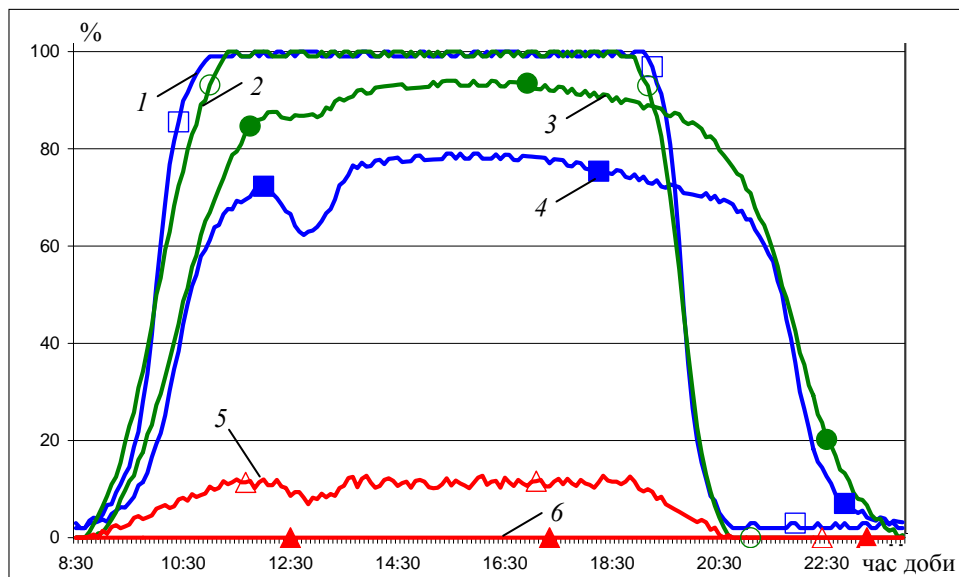


Рис. 2. Усереднені показники пікових навантажень на сервер БД, %: 1 — зайнятість процесів без черги, 2 — активність дисків без черги, 3 — активність дисків із чергою, 4 — зайнятість процесів із чергою, 5 — кількість збоїв без черги, 6 — кількість збоїв із чергою

Графіки зайнятості процесорів та дисків виявилися дуже подібними. Після впровадження черги ці показники знизилися, їх усереднені значення жодного разу не досягли 100 %. Проте в часі графіки розтягнулися приблизно на 2,5 години завдяки тому, що виконання звітів рівномірніше розподілилося в часі. Розраховано інтегральний показник: відношення площі під усередненим графіком до і після впровадження черги. Для зайнятості процесорів ця частка рівна 1,16; для активності дисків — 1,06. Отже, у періоди найбільших навантажень на БД середньостатистична завантаженість сервера в даних конкретних інформаційних системах знизилася в середньому на 11 %. Це відбулося завдяки тому, що кожен процес став успішно завершуватися з першої спроби, а ідентичні звіти виконувалися лише по одному разу.

Окрім навантаження на сервер, експериментальним шляхом проведено оцінку мережевого трафіку у двох напрямках: «від» серверного сегмента та «до» серверного сегмента корпоративної мережі. Засобами моніторингу апаратного мережевого маршрутизатора щодоби реєструвався сумарний (вхідний і вихідний) мережевий трафік у точці входу до серверного сегмента підприємства. Реєстрація проводилася протягом 30 діб до та 30 діб після повного переходу КМПД на ієрархічну схему розповсюдження всіх поновлень системного та прикладного ПЗ. Було введено лише один рівень ієрархії — сервери поновлень структурних підрозділів. Ці сервери завантажують усі поновлення безпосередньо із центральних серверів підприємства та «віддають» файли поновлень усім робочим місцям «свого» підрозділу. Завдяки зниженню цього виду технологічного трафіку середньомісячний обсяг інформації, переданої через точку входу в серверний сегмент мережі, знизився майже на 7 %.

ВИСНОВКИ

У роботі розглянуто сучасні тенденції побудови та розвитку КІС із централізованим збереженням та обробкою даних. Описано технічні та технологічні проблеми, що виникають у більшості таких КІС. Для вирішення типових проблем запропоновано поєднання організаційних, технологічних і технічних методів. Експериментально підтверджено, що запропоновані методи і технології дозволяють уникнути вузьких місць у КІС, знизити або взагалі ліквідувати пікові навантаження на сервер БД і на мережу КІС, і в результаті знизити капіталовкладення в КІС.

ЛІТЕРАТУРА

1. *Остер Д.* Архитектура систем планирования и управления ресурсами корпорации. — М.: Вершина, 2006. — 208 с.
2. *Спириус Ф.* Корпоративные базы данных: Проектирование и оптимизация. — М.: Вершина, 2007. — 316 с.
3. *Пфаулер Дж.* Корпоративные программные приложения. — СПб.: ВHV, 2008. — 512 с.

Надійшла 12.05.2009