

пошуку містить спеціальні символи або додатковий компонент імені, яка передує кореню файлової системи. Реалізують цей метод без внесення змін у ядро системи, внаслідок чого він більш простий, ніж інші методи, але менш гнучкий.

Абсолютно прозорі розподілені системи, в яких для звертання до файлів, розташованих на інших машинах, достатньо вказати стандартні складні імена файлів, розпізнання цих файлів як дистанційних є функцією ядра. Маршрути пошуку файлів вказано в їх складних іменах.

1.5. Характеристики розподілених систем

Розподілені системи мають такі характерні риси:

- просторова розподіленість компонент розподіленої системи, тобто вони взаємодіють або локально, або віддалено;
- компоненти розподіленої системи можуть працювати паралельно, через що швидкість роботи зростає порівняно з послідовною роботою;
- кожний стан компонента розглядається локально, тобто з погляду певного обчислювального процесу, запущеного з локального робочого місця.
- компоненти працюють окремо й можуть «випадати», не руйнуючи системи в цілому, незалежно одна від одної, таким чином розподілені системи підлягають частковому системному «випаданню»;
- система працює асинхронно, процеси комунікації й обробки не керуються глобальним системним часом, змінні й процеси синхронізуються;
- у розподіленій системі функції керування розподіляються між різними автономними компонентами, оскільки жодна окрема компонента не може здійснювати весь контроль, що гарантує певний рівень автономії;
- розподілена система може утворюватися як об'єднання наявних систем, тобто, потрібно контекстно-повне керування іменами, що дає можливість однозначно інтерпретувати найменування в рамках адміністративної або технологічної області – такий випадок вважають федеративним керуванням іменами;
- для підвищення потужності розподіленої системи, програми й дані можна переміщувати між різними вузлами, таку концепцію називають **міграці-**

єю, для підтримки якої використовують додаткові механізми, які протоколюють стан та розташування програм і даних;

- розподілена система має використовувати динамічні зміни структури, ця **динамічна реконфігурація** потрібна, наприклад, тоді, коли протягом певного часу очікуються нові з'єднання;

- архітектура комп'ютерів може використовувати різні топології й механізми, зокрема якщо апаратура надходить від різних виробників, то цю особливість називають гетерогенністю;

- розподіленій системі властива еволюція, вона може змінюватися протягом часу її життя;

- джерела відомостей, одиниці обробки й користувачі можуть бути фізично мобільні, а програми й дані можуть переміщуватися між вузлами для одержання даних та ресурсів з інших вузлів системи або підвищення потужності.

Для досягнення цих характерних рис слід виконувати певні вимоги, які висуваються до розподілених систем: прозорість, відкритість, гнучкість, масштабованість, стійкість, безпека, ефективність.

Прозорість є основною вимогою, яка спрощує взаємодію розподілених прикладних програм і приховує фізичне розподілення процесів та ресурсів серед багатьох комп'ютерів. Цей принцип дозволяє замовчувати складність реалізації розподілених систем від користувача та внутрішні процеси в системі за допомогою прозорих для нього функцій, унаслідок чого розподілена система для користувача стає більш зручною. Розподілені системи, які уявляються користувачам і прикладним програмам у вигляді єдиної комп'ютерної системи, називають прозорими (transparent).

Розрізняють багато характерних рис прозорості розподілу, серед яких відокремимо такі:

- прозорість доступу приховує спеціальні механізми доступу для локальних або віддалених служб до ресурсів системи;

- прозорість місця розташування приховує від користувача системні топології, тобто приховує фізичне місце розташування ресурсів (досягається

присвоєнням логічних імен, дозволяє замовчувати різні формати даних, які передаються різним процесам);

- прозорість міграції приховує факт переміщення ресурсів у інше місце;
- прозорість виконання приховує спосіб обробки обчислювальних потоків, коли обчислення виконуються паралельно або послідовно;
- прозорість зміни місця розташування приховує зміну ресурсу під час його обробки;
- прозорість реплікації приховує факт реплікації під час обробки даних;
- прозорість паралельного доступу приховує факт спільного використання ресурсів декількома конкуруючими процесами користувачів;
- прозорість відмови приховує факт виникнення відмов у системі й відновлення ресурсу після їх локалізації;
- збереження приховує факт розташування ресурсу на носіях або кешування ресурсу;
- прозорість з'єднання приховує межі між адміністративними й технологічними областями в системі;
- прозорість групи приховує користувачів від групи під час колективної роботи із системою.

Відкрита розподілена система (open distributed system) – це система, що пропонує стандартні засоби та служби доступу до системи широкому колу користувачів, які використовують стандартні синтаксис і семантику всіх протоколів взаємодії. Всі протоколи взаємодії компонентів усередині розподіленої системи в ідеальному випадку ґрунтуються на загальнодоступних стандартах, що дозволяє використовувати для створення компонент різні засоби розробки й різні операційні системи, а кожна компонента має точну й повну специфікацію своїх сервісів. За таких умов компоненти розподіленої системи можуть бути створені незалежними розробниками. У разі порушення цієї вимоги може стати неможливим поява розподіленої системи, яка охоплює кілька незалежних організацій.

Важливою характеристикою, що забезпечує відкритість розподілених систем, є наявність загальних специфікацій інтерфейсів, які підтримуються

службами розподілених систем для реалізації інтерфейсів різними виробниками програмного забезпечення. Такі специфікації інтерфейсів є однозначними для різних реалізацій конкретного інтерфейсу.

Гнучкість – простота конфігурування системи, яка складається з різних компонентів різних виробників; можливість перенесення системи з одних операційних систем на інші, навіть в іншу файлову систему. Не має виникати ускладнень під час введення в систему нових компонентів або заміни наявних, при цьому інші компоненти, з якими не проводилося жодних дій, залишаються незмінними. Для забезпечення масштабованості слід використовувати децентралізовані дані, алгоритми, служби. У разі географічної масштабованості збільшується ризик атакування ресурсів з боку територіально віддалених користувачів, а процедура узгодження вимагає реєстрації користувачів у одному інформаційному просторі власної області або домену, а також реєстрації користувачів одного домену в іншому з наданням їм відповідних прав доступу. Для забезпечення гнучкості розподіленої системи її слід будувати за принципом модульності, коли кожний її компонент – це певний модуль, що має своє функціональне призначення. У той же час кожний компонент підтримує принцип федеративності для співіснування автономних модулів на підставі певних стандартизованих правил. Така побудова розподілених систем забезпечує ефективну керованість, надійність функціонування системи й підтримку значень показників функціонування на заданому рівні, зокрема швидкодії.

Масштабованість обчислювальних систем має кілька аспектів, найбільш важливим з яких є можливість долучити до розподіленої системи нові комп'ютери для підвищення продуктивності системи, зумовленого необхідністю балансування навантаження (load balancing) на сервери системи. Проблема масштабування має враховувати ефективність розподілу ресурсів серверів, які обслуговують запити клієнтів.

Масштабованість системи може вимірюватися за трьома різними показниками: по-перше, система може бути масштабованою за розміром, що означає легкість під'єднання до неї додаткових користувачів і ресурсів; по-друге, система може масштабуватися географічно, тобто користувачі й ресурси

можуть бути рознесені у просторі; по-третє, система може бути масштабованою адміністративно, тобто бути простою в керуванні під час її роботи в адміністративно незалежних організаціях. На жаль, система, якій властива масштабованість за одним або декількома із цих параметрів, у разі масштабування часто втрачає продуктивність.

Основні підходи, що забезпечують масштабованість розподіленої системи:

1. **Приховування від користувача часу очікування зв'язку** передбачає можливість не враховувати час очікування відповіді від віддаленого сервера, але прикладні програми розраховують на синхронний вид зв'язку, коли передбачається переривання активного процесу під час відповіді на раніше надісланий запит. За таких умов прикладна програма викликає оброблювача запитів, який і погоджує всі обчислювальні процеси в системі й забезпечує логічну цілісність даних. За рахунок такого принципу обробки запит користувача в розподіленій системі має бути або коректно виконаний повністю, або не виконаний взагалі. Ситуація, коли частина компонент системи коректно обробила запит, що надійшов, а частина – ні, є найгіршою й неможливою. Такий підхід характерний для розподілених систем, які використовують технологію обробки пакетів у процесі передачі інформації з мережі.

2. Розподіл передбачає **розбиття компонентів системи на дрібні частини** й подальше рознесення цих частин фізичними територіально розподіленими вузлами системи. Прикладом розподілу є система доменних імен Internet (DNS).

3. **Реплікація** не тільки підвищує доступність мережних ресурсів, але й допомагає вирівняти завантаження компонентів системи, що приводить до підвищення продуктивності. **Кешування** (caching) є особливою формою реплікації, причому розбіжності між ними нерідко малопомітні або взагалі штучні, оскільки і під час реплікації, результатом кешування є створення копії ресурсу зазвичай у безпосередній близькості від клієнта, який використовує цей ресурс. Кешування – це дія, яку виконує споживач ресурсу, а не його власник. На масштабованість може погано вплинути один істотний недолік кешування й реплікації, оскільки створюється низка копій ресурсу, модифікація однієї копії робить її відмінною від інших, відповідно, кешування й реплікація зумовлюють проблеми несуперечності даних (consistency).

Стійкість. Під стійкістю розуміємо можливість дублювання декількома комп'ютерами функцій або ж можливість автоматичного розподілу функцій усередині системи у разі виходу з ладу одного з комп'ютерів. У ідеальному випадку немає унікальної точки збою, тобто вихід з ладу будь-якого комп'ютера не заважає обслуговуванню запитів користувача. Стійкість розподіленої системи пов'язана з поняттям масштабованості, але не еквівалентна йому.

Приклад. Припустімо, система використовує набір серверів, які обробляють запити, й одного диспетчера запитів, який розподіляє запити користувачів між серверами. Таку систему можна вважати добре масштабованою, однак диспетчер є вразливим місцем такої системи. Натомість, система з єдиним сервером може бути стійкою, якщо наявний механізм його автоматичної заміни у разі виходу з ладу.

Виконання вимог стійкості й масштабованості зазвичай спричиняє деякі додаткові витрати, що на практиці не завжди доцільно. Однак технології, які використовують під час побудови розподілених систем, мають допускати принципову можливість створення стійких і високомасштабованих систем.

Безпека. Для кожного компонента, який утворює розподілену систему, має бути використана функція авторизації не тільки для компонент, які його викликають, але і для користувачів. Дані, передані між компонентами, слід захистити як від спотворення, так і від перегляду третіми сторонами.

Ефективність. Під ефективністю розподілених систем будемо розуміти мінімізацію накладних витрат, спричинених розподіленим характером системи. Оскільки ефективність у такому вузькому значенні може суперечити безпеці, відкритості й надійності системи, то вимога ефективності в цьому контексті є найменш пріоритетною. Наприклад, на підтримку логічної цілісності даних у розподіленій системі можуть витрачатися значні ресурси часу й пам'яті, однак система з недостовірними даними не потрібна користувачам, тому такі додаткові витрати є виправданими.

1.6. Висновки

Розподілені системи складаються з автономних комп'ютерів, які працюють спільно, створюючи уявлення про роботу в єдиній зв'язній системі. Їх перевага полягає в тому, що вони спрощують інтеграцію різних приклад-

них програм, які працюють на різних комп'ютерах, у єдину систему, добре масштабуються за умови правильного проектування.

Розмір розподілених систем обмежується тільки розміром базової мережі.

Розподілені системи класифікують за принципами, в основі яких лежать характеристики апаратного та програмного забезпечення.

Реалізація розподілених систем характеризується складністю програмного забезпечення, зниженням продуктивності й наявністю проблем з безпекою.

Розподілені операційні системи використовують для керування апаратним забезпеченням взаємозалежних комп'ютерних систем та сприймають у вигляді єдиної системи.

Мережні операційні системи ефективно поєднують різні комп'ютери, які працюють під керуванням своїх операційних систем, забезпечуючи користувачам доступ до локальних служб кожного з вузлів. Однак мережні операційні системи не створюють у користувача відчуття роботи з єдиною системою, що характерно для розподілених операційних систем.

1.7. Запитання для самоконтролю

1. Дайте визначення поняттю «розподілена система».
2. Яким є основне завдання розподіленої системи?
3. Коли та чим була зумовлена відмова від централізації?
4. Назвіть переваги розподілених систем.
5. Які недоліки мають розподілені системи?
6. Наведіть класифікацію розподілених систем відповідно до апаратного забезпечення.
7. Наведіть класифікацію розподілених систем відповідно до програмного забезпечення.
8. Які характерні риси мають розподілені системи?
9. Яким вимогам мають задовольняти розподілені системи? Коротко охарактеризуйте кожна з них.
10. Дайте визначення поняттю «суперкомп'ютер».

11. Назвіть основні компоненти паралельних комп'ютерів.
12. Чому апаратура комп'ютерів розвивається у напрямку «розподілення»?
13. Наведіть приклади реалізації апаратури зі спільною пам'яттю для ПК.
14. Наведіть приклади реалізації апаратури з розділюваною пам'яттю для ПК.