

## ОСОБЛИВОСТІ АРХІТЕКТУРНОГО І ФУНКЦІОНАЛЬНО-СТРУКТУРНОГО ПРОЕКТУВАННЯ СИСТЕМ ТА МЕРЕЖ ЗОВНІШНЬОГО ЗБЕРІГАННЯ І ОБРОБКИ ДАНИХ

Дано обґрунтування вибору формальних та неформальних методів опису систем і мереж зберігання і обробки даних. Досліджені особливості застосування розглянутих методів в технологіях архітектурного і функціонально-структурного моделювання та проектування на новій платформі багатофункціональних мереж зовнішніх запам'ятовуючих пристроїв. Розглянуті рівні абстракції мережі, запропонована концептуальна схема використання багатофункціонального мережевого зовнішнього запам'ятовуючого пристрою на базі агентно-орієнтованої технології.

**Ключові слова:** зовнішнє зберігання даних, багатофункціональна мережа, віртуальна машина, обробка даних, агентно-орієнтована технологія.

Trembovetskyi M. P., Zaika V. F. State University of Telecommunications, Kyiv

## FEATURES OF ARCHITECTURAL AND FUNCTIONAL-STRUCTURAL DESIGN OF SYSTEMS AND NETWORKS OF EXTERNAL STORAGE AND DATA PROCESSING

The justification for the choice of formal and informal methods for describing the systems and networks of data storage and processing is given. The peculiarities of application of the considered methods in architectural and functional structural modeling technologies and designing on the new platform of multifunctional networks of external storage devices are explored. It is shown that in order to improve the efficiency of implementation it is advisable to use logical-algebraic models and methods, adding to them a formally defined operational semantics, based on procedural knowledge about the functioning of the system. Considered levels of network abstraction, used for virtualization and modeling of complex multifunctional systems and networks of data storage and processing. On this basis, it is possible to define a network of abstract machines over structured memory space and use its description as formal specifications of the hierarchy of virtual machines. The conceptual scheme of use of multifunctional network external storage device based on agent-oriented technology is proposed.

**Keywords:** network external storage, multifunctional system, virtual machine, data processing, agent-oriented technology.

Трембовецкий М. П., Заика В. Ф. Государственный университет телекоммуникаций, Киев

## ОСОБЕННОСТИ АРХИТЕКТУРНОГО И ФУНКЦИОНАЛЬНО-СТРУКТУРНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ И СЕТЕЙ ВНЕШНЕГО ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

Дано обоснование выбора формальных и неформальных методов описания систем и сетей хранения и обработки данных. Исследованы особенности применения рассмотренных методов в технологиях архитектурного и функционально-структурного моделирования и проектирования на новой платформе многофункциональных сетей. Рассмотрены уровни абстракции сети, предложена концептуальная схема использования многофункционального сетевого внешнего запоминающего устройства на базе агентно-ориентированной технологии.

**Ключевые слова:** внешнее хранение данных, многофункциональная сеть, виртуальная машина, обработка данных, агентно-ориентированная технология.

**1. Вступ.** Вирішенню проблем створення систем і мереж зберігання і обробки даних присвячені роботи багатьох вітчизняних і зарубіжних дослідників. Але, в той же час, в існуючих роботах недостатня увага приділена опису реалізацій систем і мереж, побудова яких заснована на інтеграції і консолідації ресурсів зовнішнього зберігання з ресурсами віддаленої обробки даних, на концепціях структурування декларативних і процедурних знань при архітектурному проектуванні, що визначає основну проблемність ситуації. Відомі розробки засновані на практичному досвіді розробників та на застосування змістовних і неформалізованих концептуальних моделей, що, в свою чергу, ускладнює проектування і, як результат, збільшує терміни реалізації проекту.

## **2. Агентно-орієнтована технологія у мережах зовнішнього зберігання даних**

Активність мережевої системи зберігання і обробки даних, що представлена або у вигляді окремої функціонуючої системи, або у вигляді інтегрованої мережі, при формуванні єдиного системного образу може бути підвищена за рахунок організації динамічного управління ресурсами на базі агентно-орієнтованої архітектури [1].

Наприклад, на рис. 1 проілюстровано використання агентно-орієнтованої технології в мережах зовнішнього зберігання даних [2]. Поєднання агентно-базованих і агентно-орієнтованих архітектур, використання абстракцій мобільності агентів і збережених даних дозволяють створювати різноманітні мережеві архітектури систем зберігання і обробки даних. Тому можна вважати, що при реалізації мереж віртуальних машин доцільно використовувати технологію мобільних агентів, причому агент може грати більш важливу роль вже на етапі визначення мережі абстрактних машин у якості абстрактного виконавця (інтерпретатора) логіко-алгебраїчного виразу, що описує роботу модуля абстрактної машини. При реалізації ієрархії віртуальних машин у якості основи реалізації може бути обрана одна з мультиагентних систем, що інтегрована із системою погодження і координації процесів і об'єктів [3, 4].



Рис. 1. Агентно-орієнтована технологія у мережі зовнішнього зберігання даних

Під час опису і, можливо, при реалізації сценарію зручно використовувати абстракцію агента. Терміном "агент" позначимо абстрактний об'єкт, або "виконавець" сценарію. Агент "розглядає" сценарій у якості плану власних дій; він може клонуватися, зливатися з іншими агентами, знищуватися, переходити з одного вузла на інший. У однопроцесорній системі тільки один агент може бути активним, або "поточним", а в мультипроцесорній системі або у багатовузловій обчислювальній мережі можуть бути активними кілька агентів, можливо, взаємодіючи між собою.

В імітаційній моделі, що реалізована на однопроцесорній обчислювальній машині, агенти можуть виконуватися квазіодночасно (даний режим виконання зазвичай реалізується у традиційних імітаційних системах [5, 6]). В останньому випадку робота агента при виконанні імітаційної моделі повинна бути прив'язана до системного часу, причому, послідовності активних фаз повинна відповідати неубутна послідовність моментів системного часу. Як і в загальноцільових імітаційних системах типу Simula 67 [7], під подією будемо розуміти виконання активної фази процесу, що супроводжується зміною стану моделі.

У загальному випадку дії агента по виконанню ієрархічного сценарію на деякому фіксованому рівні відповідають виконанню послідовностей одиночних або згрупованих подій по реалізації підсценарія нижнього рівня по відношенню до поточного. Дану послідовність дій агента назвемо правилами дій або просто сценарієм його поведінки. На самому нижньому рівні ієрархічної сценарної моделі виконуються модулі абстрактних машин.

Переходячи від абстрактного поняття "агент" до конкретного фізичного, підкреслимо, що в реальній системі або мережі зберігання і обробки даних, агенту може відповідати повідомлення, блок даних, код програми, що переміщується (для мобільного агента) і ін. Стационарним агентом може бути програма, яка перебуває в активному стані (наприклад, та, що опитує вхідні порти) і виконує команди, що надходять по каналах зв'язку від інших програм, які виконуються на інших обчислювальних модулях системи або мережі. У зв'язку з цим, особливо важливо, щоб в логіко-алгебраїчних моделях системи або мережі зберігання і обробки даних були відображені (і збережені в базі даних) поточні просторово-часові відносини між об'єктами предметної області. Агенти можуть встановити прямі зв'язки з базою даних, розміщеною на сервері, і робити до неї запити. За рахунок агентів можуть бути розширені функціональні можливості додатків, наприклад, в частині групової роботи. Агенти можуть працювати як на серверному, так і на клієнтському боках, вирішуючи при цьому довільні завдання.

Використання мобільних агентів, що володіють здатністю переміщатися по мережі, дозволяє створювати гнучкі операційні середовища, в тому числі мережні імітаційні системи. Використання мобільних або стационарних агентів дозволяє здійснити перехід від клієнт-серверної архітектури мережі зберігання даних до більш розвиненою функціонально трирівневої архітектури "клієнт-агент-сервер", що дозволяє перетворити мережу зберігання і обробки даних у динамічну змінну за запитами проектувальника або користувача середу зберігання і обробки даних.

### **3. Розподілені сховища об'єктів**

До іншого класу систем, які придатні для реалізації широкого класу розподілених додатків, відносяться розподілені сховища об'єктів. Наприклад, до таких систем відноситься система JavaSpaces, розроблена фірмою Sun [8, 9]. Записаний до сховища об'єктів може бути знайдений за допомогою об'єктів-шаблонів за значеннями певних полів. До числа позитивних достоїнств даної системи є можливість побудови на її основі відмовостійких систем типу кластерів, так як у разі виходу з ладу одного з компонентів системи, об'єкти можуть бути відновлені.

Організація управління розподіленими ресурсами у мережах зовнішнього зберігання і обробки даних складна, оскільки вони відносяться до класу складних гетерогенних систем. Загальні вимоги до систем управління зберіганням і обробкою даних, що зменшують при реалізації їх вартість і складність, зводяться до наступних [8].

Системи управління повинні бути:

- автоматизованими, щоб звести до мінімуму втручання персоналу;
- стандартизованими, щоб забезпечити взаємодію компонентів програмного забезпечення систем управління;
- відкритими, щоб програмне забезпечення системи управління могло взаємодіяти з різними типами керованих ресурсів в мережі;
- платформо незалежними, щоб можна було працювати з даними різних платформ;
- простими у розгортанні і використанні;
- мати централізоване управління, щоб розподіленими ресурсами можна було керувати з одного місця.

Багатьма з перерахованих властивостей володіє розроблена фірмою Sun технологія Jіго на платформі Java [8]. Jіго дозволяє розробляти системи управління розподіленими ресурсами (системами, пристроями, додатками) у гетерогенних мережах. Основною концепцією Jіго є трирівнева архітектура систем управління (рис. 2).

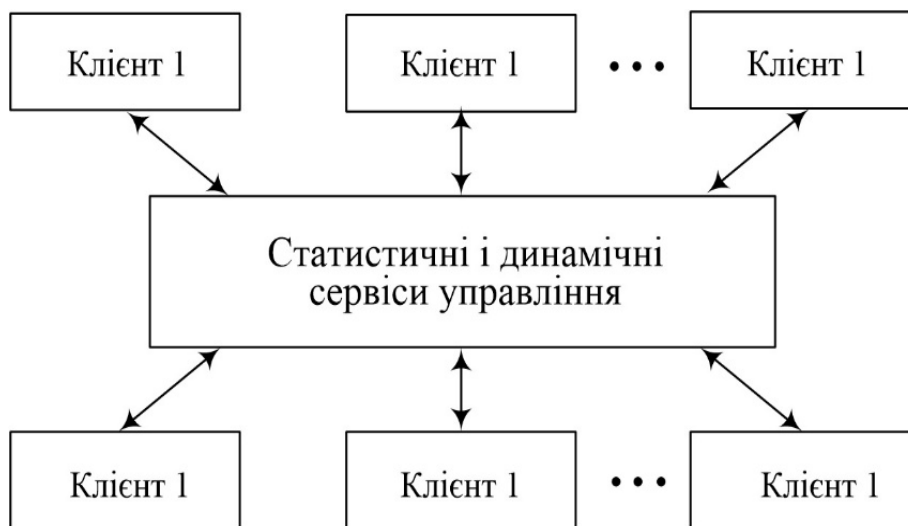


Рис. 2. Інтегрована трирівнева архітектура управління ресурсами у гетеродинних мережах

На клієнтському рівні додатку здійснюють взаємодії із сервісами управління. На середньому рівні технологія Jіго надає статичні і динамічні сервіси управління, а на нижньому рівні знаходяться неоднорідні ресурси. Jіго дозволяє розробляти керуючі додатки на основі наданої інфраструктури управління та сервісів. Інфраструктура управління може використовувати механізми RMI (віддалений виклик методів), Jіні (набір мережових протоколів, програмних модулів і сервісів) і сервер класів. Jіні використовується у якості динамічного сервісу пошуку.

Керуючі додатки, які використовуються при віртуалізації (або мережовій реалізації) систем зберігання і обробки даних можна створювати, інтегруючи технологію Jіні з пірінговими і агентно-орієнтованими технологіями. Як відомо [8], у мережевому пірінговому додатку кожен вузол здатний виконувати як функції клієнта, так і функції сервера з відносно

мінімальними вимогами до обчислювальної потужності і обсягу пам'яті, забезпечуючи швидкий обмін повідомленнями і спільне використання файлів. Агенти по додатковій пірінговій мережі можуть передавати повідомлення, команди, модулі керуючих програм, забезпечуючи при цьому функціонування вторинної "платформи віртуалізації".

На рис. 3 представлені дві інтегровані мережеві структури, що забезпечують функціонування віртуальної системи зберігання і обробки даних. Зафарбованим кулькам відповідають вузли віртуалізації, які пов'язані швидкодіючими каналами зв'язку (ці зв'язки позначені суцільними лініями). Звичайними кружками позначені вузли вторинної мережі, на яких розміщені програмні агенти, що забезпечують функціонування платформи віртуалізації. Дані вузли пов'язані пунктирними лініями, що позначають звичайні канали зв'язку.

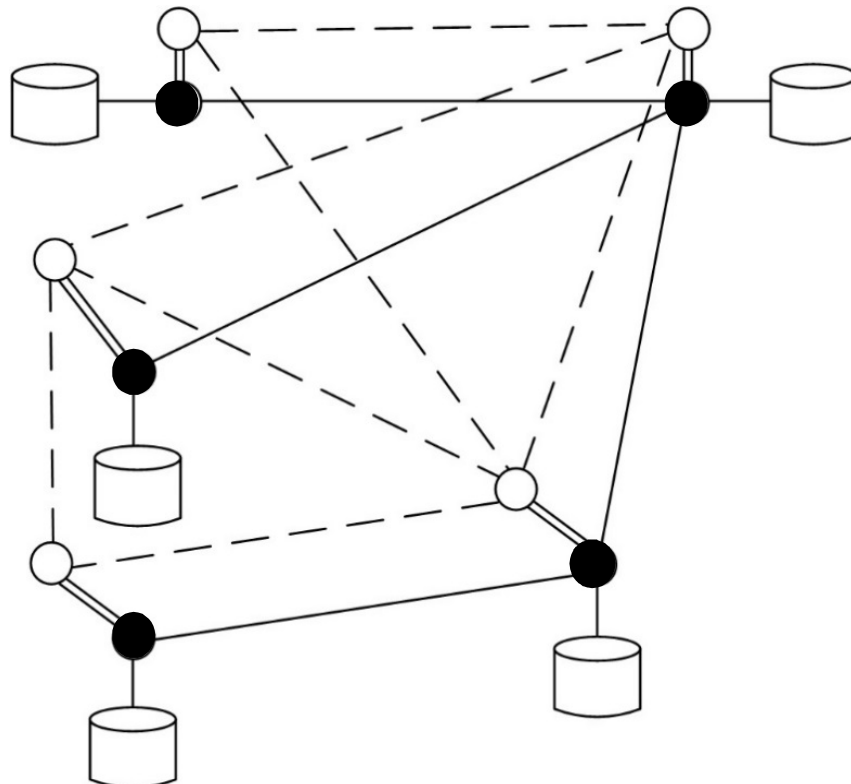


Рис. 3. Первинна мережа зберігання і обробки даних і вторинна мережа підтримки віртуальної архітектури

Швидкодія даної вторинної мережі може бути нижчою, ніж у основної, якщо необхідний режим віртуалізації встановлюється відносно рідко. Такий запропонований варіант мережевої віртуалізації має гнучкість і дозволяє змінювати план віртуалізації у залежності від вимоги користувача мережі.

Для проведення експериментів з віртуальними мережами зберігання і обробки даних запропонована наступна мережева структура, що представлена на рис. 4. Фізична мережа реалізується на базі локальної або глобальної TCP/IP мережі і містить мережеві та обчислювальні ресурси. При ініціюванні даної мережі динамічно створюються до шести логічних рівнів (на рис. 4).

Мережа процесів-демонів представляє середовище для виконання мобільних агентів. Логічні мережі віртуалізації обробки даних і, власне, обробки даних із заданими користувачами топологіями, утворюють другий і третій рівні і створюються на базі мережі процесів-демонів. Логічні мережі віртуалізації зберігання даних і власне зберігання даних,

утворюють четвертий і п'ятий рівні. Як і попередні мережі, вони створюються на базі мережі процесів-демонів при використанні додаткових серверів баз даних (наприклад, MySQL). На останньому, шостому, рівні знаходиться логічна мережа, в якій зберігаються метазнання про структуру інформації, що зберігається у мережі зберігання даних інформації.

Таким чином, на базі фізичної мережі створюється шість абстрактних рівнів для реалізації віртуального операційного середовища, що активно використовує віртуальний структурований простір даних. Програмна реалізація цих рівнів не тільки дозволяє проводити широке коло експериментів по апробації різних архітектурних рішень, але і використовувати мережу на практиці за прямим призначенням для виконання функцій системи зберігання структурованих даних і машини бази даних із розподіленою і паралельною обробкою.

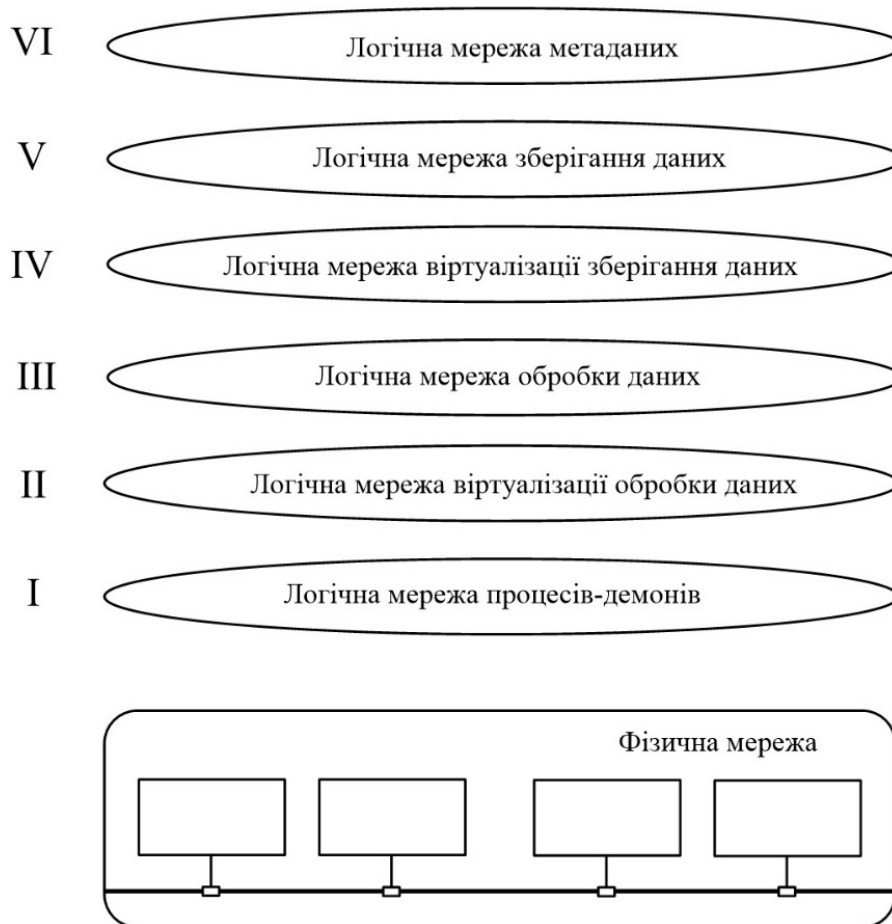


Рис. 4. Рівні абстракції мережі, що використовується для віртуалізації і моделювання складних багатофункціональних систем і мереж зберігання і обробки даних

На основі подібної мережі були створені також реалізації сценарних мереж і мереж абстрактних машин. Сценарні мережі та мережі абстрактних машин можуть бути покладені в основу побудови концептуальних схем, що описують архітектуру систем і мереж зберігання і обробки даних. Розвиваючи дані концептуальні схеми, можна далі побудувати імітаційні поведінкові моделі, після аналізу яких та отримання необхідних характеристик продуктивності системи, можна перейти до її фізичної реалізації. Побудова моделей, що розвиваються, або еволюціонують на єдиній логіко-алгебраїчній основі, дозволить зберегти послідовність формальних специфікацій, що розробляються з метою їх безпосереднього використання на етапах структурного синтезу і фізичної реалізації системи.

#### 4. Розвиток функціональних можливостей мережевих пристроїв зовнішнього зберігання даних

У даний час ринок мереж зовнішнього зберігання даних розвивається у бік стирання умовних граней між різними технологіями зберігання і обробки даних. Наприклад, в одному і тому ж середовищі можуть бути успішно реалізовані дві поширені технології – мережевих пристроїв NAS і мереж зберігання SAN.

Інтенсивно розвиваються і інші мережеві технології зберігання даних, наприклад, технологія NASD підключених до мережі захищених пристроїв (Network Attached Secure Devices), що розроблена в університеті Карнегі-Меллона (США) і ще у кількох компаніях, що співпрацювали з даними університетом. Основна концепція технології NASD – підвищення рівня "інтелектуальності" дискових пристроїв при управлінні мережевими взаємодіями і роботою сховищ даних. Блоки, які реалізують ці функції, можуть розміщуватися безпосередньо у дисковому модулі [10].

Особливий інтерес представляє можливість використання безпосередньо у дисковому модулі блоків, що реалізують серверні протоколи FTP або HTTP, що дозволяє використовувати мережеві пристрої NASD як Web-серверів.

Іншою перспективною технологією є технологія файлових систем прямого доступу DAFS (Direct Access File System) [10]. Дана технологія, у свою чергу, активно використовує нові технології – архітектури віртуального інтерфейсу VIA (Virtual Interface Architecture) і InfiniBand ("послідовна" мережева технологія). Основне призначення пристроїв DAFS – побудова на їх основі масштабованих DAFS-кластерів для роботи з транзакціями. Клієнтські DAFS-з'єднання забезпечують малі часи затримок між системами баз даних і DAFS-пристроями, причому на ці пристрої можна перенести всі обчислення, що пов'язані із визначенням структур даних.

Високоорганізовані мережеві системи зберігання даних, зазвичай можуть виключити засоби підтримки додатків і інший інструментарій користувача.

Рис. 5 ілюструє спрощену еволюцію мережевих пристроїв зберігання даних.

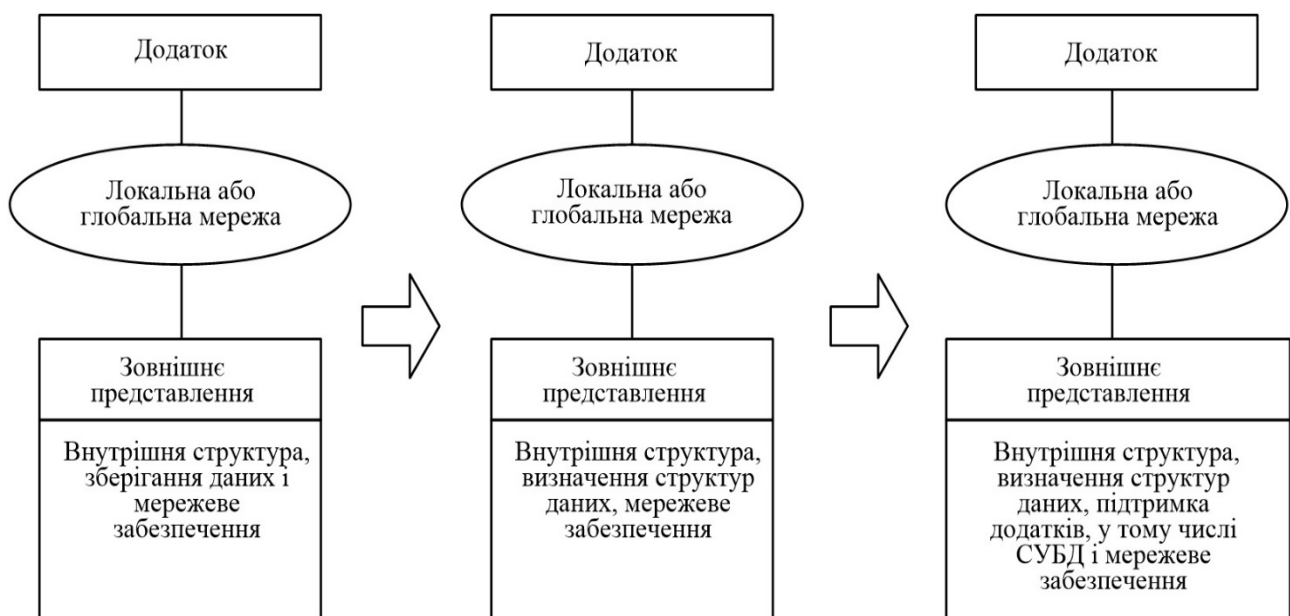


Рис. 5. Спрощена концепція розвитку мережевих пристроїв зовнішнього зберігання даних

Перші дві стадії відповідають відомим мережевим пристроям NAS, NASD, DAFS. Третя структура потрібна для реалізації активних інтелектуальних мереж зовнішнього зберігання і обробки даних, що використовуються в якості інтегрованих мережесвих співпроцесорів.

Варіанти фізичної реалізації мереж зберігання і обробки даних, що відповідають описаним вище стадіях розвитку, представлені на рис. 6 і 7 [10]. На рис. 6 представлена мережа із виділеною системою зберігання даних, а на рис. 7 – інтегрована мережа, у якій розподіл трафіку забезпечує один комутатор.

Той чи інший варіант реалізації мережі зберігання вибирається у залежності від вимог до її продуктивності і вартості.

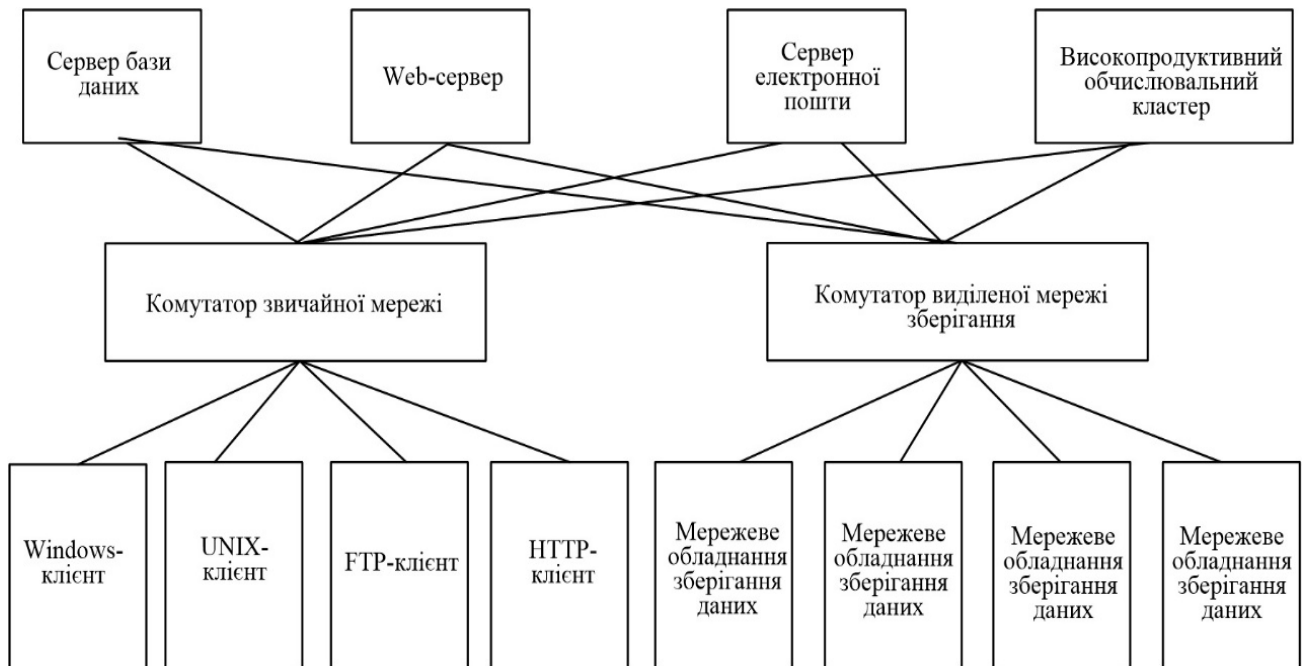


Рис. 6. Розподілена мережа зберігання і обробки даних



Рис. 7. Інтегрована мережа зберігання і обробки даних



Розглянемо далі деякі перспективні розподілені платформи для реалізації активних мереж зберігання даних. Широко поширені моделі взаємодії мережеских додатків, такі, як віддалений виклик процедур, віддалене звернення до методів, орієнтований на повідомлення проміжний рівень, потоки даних, трьохланкова архітектура "клієнт-сервер", добре узгоджуються з концепціями технологій розподіленого зберігання даних.

Агентно-орієнтовані платформи, узагальнена модель яких представлена на рис. 8 [3], також є ефективним засобом реалізації активних інтелектуальних систем зовнішнього зберігання і обробки даних. Існуючі агентно-орієнтовані платформи дозволяють вибрати конкретні модифікації агентів для реалізації функцій агентів-серверів і агентів-демонів. Підвищення ролі агентів у розподілених системах відзначається, наприклад, у [3, 4]. У цих роботах дані визначення поняття "програмний агент" і класифікація агентів, що опирається на їх властивості. Програмний агент визначається як "автономний процес, здатний реагувати на середовище виконання і викликати зміни у середовищі виконання, можливо, у кооперації із користувачами або з іншими агентами".

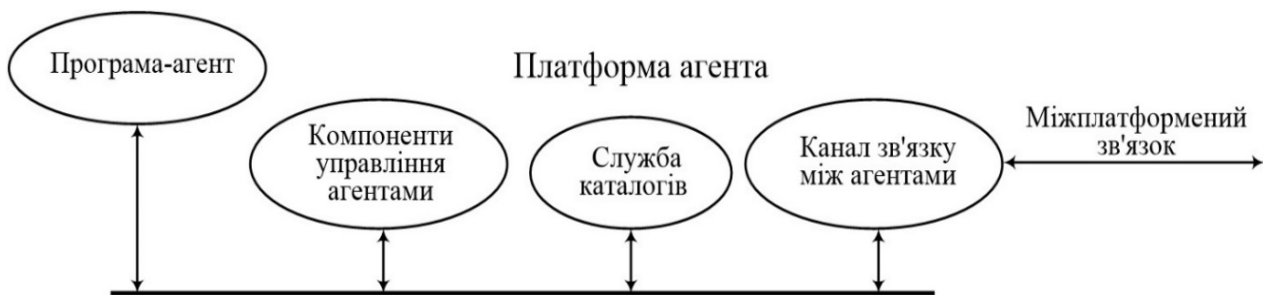


Рис. 8. Узагальнена модель програмного агента

Платформа агентів (рис. 8) містить всі необхідні служби, які використовуються типовою мультиагентною системою: компонент управління агентами надає механізми створення, знищення та виявлення агентів; локальна служба каталогів дозволяє віддаленим агентам визначати, які агенти знаходяться на локальному вузлі; через канали зв'язку між агентами реалізуються взаємодії між усіма платформами агентів. Розрізняються кооперативні, мобільні, інтерфейсні, інформаційні та інтелектуальні агенти [3].

Ефективний розподіл даних і функцій управління серед машин локальної мережі реалізовано у безсерверній файлової системі xFS, розробленої у Берклі [3]. У системі xFS використовуються три типи процесів: сервер зберігання – процес, який відповідає за зберігання частин файлу; менеджер метаданих – процес, що визначає реальне розміщення блоків файлів, які можуть розміщуватися на декількох серверах зберігання; клієнт – процес, який приймає запити користувачів на виконання операцій з файлами і має можливість кешування і передачі кешованих даних іншим клієнтам. Кожен з трьох процесів може розміщуватися на будь-якій станції локальної мережі.

На рис. 9 представлені варіанти розподілу процесів xFS: варіант повністю симетричної системи, варіант системи з довільним розміщенням процесів і система з віртуальним RAID-масивом, реалізованим на серверах зберігання. У цілому побудова файлової системи xFS відповідає розподіленій реалізації NAS-пристрою.

У системі xFS блоки даних одного файлу можуть розміщуватися на кількох серверах зберігання. Зв'язок між процесами реалізується за допомогою активних повідомлень, що дещо ускладнює взаємодію цих процесів. Особливий інтерес представляє організація локального кешування блоків, а також кооперативне кешування, що враховує наявність кеш-пам'яті у кожного клієнта, і надає можливість отримання копії блоку даних з кеш-пам'яті будь-якого іншого клієнта.

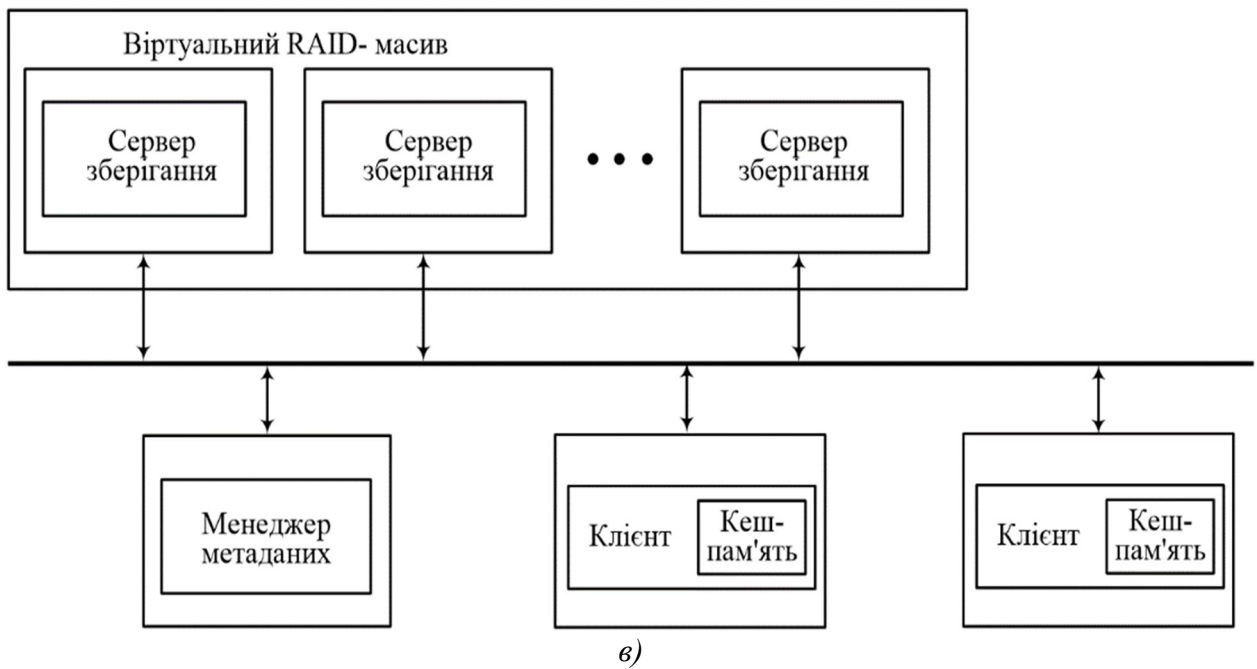
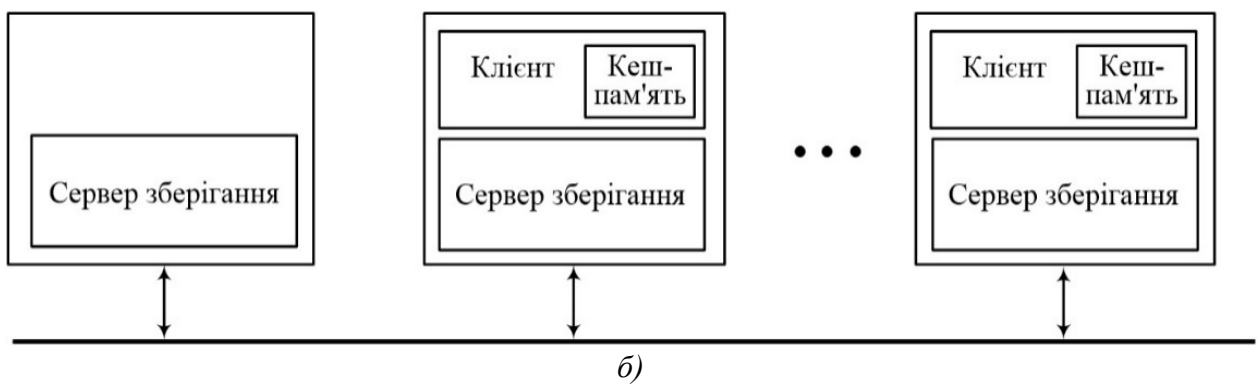
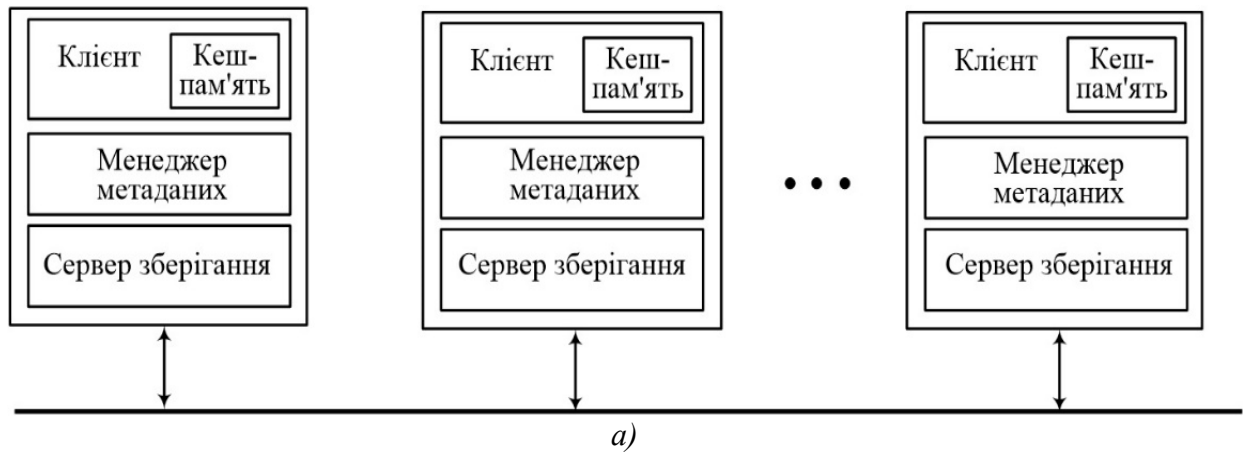


Рис. 9. Варіанти розподілу процесів xFS:  
*а* – повністю симетрична система; *б* – довільне розміщення процесів;  
*в* – реалізація RAID – масиву на серверах зберігання

На рис. 10 представлена запропонована концептуальна схема використання багатофункціонального мережевого зовнішнього запам'ятовуючого пристрою на базі агентно-орієнтованої технології. У якості платформи для реалізації різноманітних мережевих сховищ даних пропонується використовувати фізичні та логічні структури, що реалізуються на основі концептуальної схеми, представленої на рис. 4.

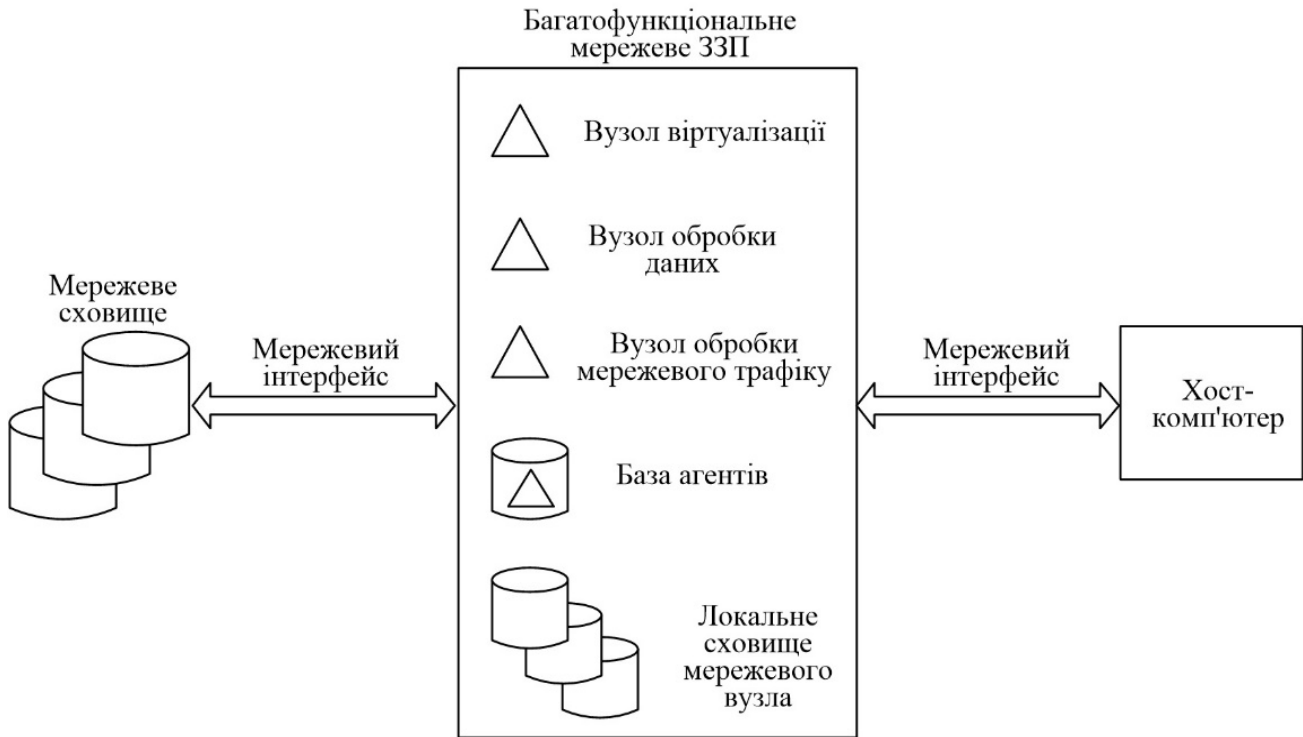


Рис. 10. Концептуальна схема використання багатофункціонального мережевого зовнішнього запам'ятовуючого пристрою

### Висновки

У статті пропонується розвинути існуючі методи формалізованого опису архітектурних, функціонально-структурних і структурно-топологічних властивостей систем і мереж зовнішнього зберігання і обробки даних. Показано, що для підвищення ефективності реалізації доцільно використовувати логіко-алгебраїчні моделі та методи, додавши до них формально певну операційну семантику, засновану на процедурних знаннях про функціонування системи. На даній основі можливо визначити мережу абстрактних машин над структурованим простором пам'яті і використовувати її опис у якості формальних специфікацій ієрархії віртуальних машин. Даний підхід дозволить підвищити якість застосовуваних рішень і прискорити процес проектування.

### Список використаної літератури

1. Корнеев В. В. Базы данных. Интеллектуальная обработка информации / В. В. Корнеев, А. Ф. Гареев, С. В. Васютин, В. В. Райх. – Москва: Нолидж, 2000. – 352 с.
2. Наик Д. Системы хранения данных в Windows / Д. Наик. – Москва: Издательский дом "Вильямс", 2005. – 432 с.
3. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. – Санкт Петербург: Питер, 2003. – 877 с.

4. Хьюз К. Параллельное и распределенное программирование на C++ / К. Хьюз, Т. Хьюз. – Москва: Издательский дом "Вильямс", 2004. – 672 с.
5. Максимей И. В. Имитационное моделирование на ЭВМ / И. В. Максимей. – Москва: Радио и связь, 1988. – 232 с.
6. Кудрявцев Е. М. GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев. – Москва: ДМК Пресс, 2004. – 320 с.
7. Андрианов А. Н. Программирование на языке СИМУЛА 67 / А. Н. Андрианов, С. П. Бычков, А. И. Хорошилов. – Москва: Наука, 1985. – 370 с.
8. Дейтел Х. М. Технология программирования на Java 2: Книга 2. Распределенные приложения / Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри. – Москва: Бинوم-Пресс, 2003. – 464 с.
9. Freeman, E., Hupfer, S., Arnold, R. JavaSpaces, Principles, Patterns and Practice. – Reading, MA: Addison-Wesley, 1999. – 368 p.
10. Фарли М. Сети хранения данных / М. Фарли. – Москва: Лори, 2003. – 550 с.

#### References

1. Korneyev V.V., Gareev A.F., Vasiutin S.V., VV Raich V.V. "Databases. Intelligent data processing." *Moskva: Nolidzh* (2000): 352.
2. Naik D. "Storage Systems into directory Windows." *Moskva: Williams* (2005): 432.
3. Tanenbaum E., Van Steen E. E. "Distributed systems. Principles and paradigms." *Sankt Peterburg: Piter* (2003): 877.
4. Hughes K., Hughes T. "Parallel and Distributed Programming in C ++." *Moskva, Williams* (2004): 672.
5. Maksimei I. V. "Simulation modeling on the computer." *Moskva: Radio i sviaz* (1988): 232.
6. Kudryavtsev E. M. "GPSS World. Based on simulation of different systems." *Moskva: DMK Press* (2004): 320.
7. Andrianov A. N., Bychko S.P., Khoroshilov A. I. "Programming SIMULA 67." *Moskva: Nauka* (1985): 370.
8. Deitel X. M., Deitel P.J., Santry S.I. "Technology of programming in Java 2. Book. 2. Distributed applications." *Moskva: Binom-Press* (2003): 464.
9. Freeman E., Hupfer S., Arnold R., "JavaSpaces, Principles, Patterns and Practice." *MA, Addison-Wesley* (1999): 368.
10. Farley M. "Network Storage." *Moskva: Laurie* (2003): 550.

#### Автори статті

**Трембовецький Максим Петрович** – доктор технічних наук, завідувач кафедри енергоефективних технологій, Державний університет телекомунікацій, Київ. Тел. +380 (50) 961 27 22. E-mail: maksimtremboff@gmail.com.

**Заїка Віктор Федорович** – доктор технічних наук, завідувач кафедри телекомунікаційних систем та мереж, Державний університет телекомунікацій, Київ. Тел. +380 (66) 102 80 04. E-mail: vfzaika@mail.ru.

#### Authors of the article

**Trembovetskyi Maksym Petrovych** – sciences doctor (technical), head of the department of energy efficient technologies, State University of Telecommunications, Kyiv. Tel. +380 (50) 961 27 22. E-mail: maksimtremboff@gmail.com.

**Zaika Viktor Fedorovich** - sciences doctor (technical), head of the department of telecommunication systems and networks, State University of Telecommunications, Kyiv. Tel. +380 (66) 102 80 04. E-mail: vfzaika@mail.ru.

Дата надходження  
в редакцію: 11.09.2017 р.

Рецензент:  
доктор технічних наук, професор М. М. Степанов  
*Державний університет телекомунікацій, Київ*