

УДК 004.8/681.5

DOI: <https://doi.org/10.32515/2414-3820.2025.55.323-331>**Р.М. Минайленко**, доц., канд. техн. наук, **О.К. Коноплицька-Слободенюк**,**І.А. Лисенко**, канд. техн. наук*Центральноукраїнський національний технічний університет, Кропивницький, Україна**e-mail: aron70@ukr.net*

Технології розподілених об'єктів в інформаційних системах

У статті показано, що розвиток інформаційно-телекомунікаційних технологій досягнув етапу, коли в розподіленому інформаційному середовищі дедалі важливішим стає не лише доступ до інформації та її обмін, але й застосування різноманітних методів аналізу та обробки даних. Активне впровадження комп'ютерних технологій у різні сфери діяльності, нарощування їх обчислювальних потужностей та широке використання мереж різного масштабу спричинили необхідність залучення високопродуктивних розподілених обчислень. Цей процес, у свою чергу, породжує проблему дефіциту обчислювальних ресурсів для виконання складних завдань.

Вирішення цих викликів можливе завдяки впровадженню паралельних і розподілених обчислювальних технологій. Однак нині основна увага приділяється опису продуктів, технологій та методологій для створення малих і середніх інформаційних систем, тоді як принципи побудови великих інтегрованих систем, що об'єднують локальні компоненти, фактично залишаються поза увагою. Унаслідок цього на етапі проектування часто обираються технології, які не задовольняють вимоги масштабних інформаційних систем. Такий підхід значною мірою знижує перспективи подальшого розвитку реалізованих проектів.

комп'ютер, розподілені обчислення, інформаційні системи, обчислювальні ресурси

Постановка проблеми. Сьогодні розвиток інформаційно-телекомунікаційних технологій досяг такого рівня, коли у розподіленому середовищі все більшу роль відіграє не лише можливість доступу до даних та їх обміну, але й потреба в аналізі та обробці цієї інформації. Масове впровадження комп'ютерних технологій у різні сфери діяльності, зростання їх обчислювальної потужності, а також активне використання мереж різного масштабу вимагають застосування високопродуктивних розподілених обчислень. Це, своєю чергою, призводить до нестачі обчислювальних ресурсів для виконання складних процесів. Одним із найбільш ефективних способів вирішення цих викликів є впровадження паралельних і розподілених обчислень. [1,3,4].

Аналіз останніх досліджень і публікацій. У сучасних умовах, коли постійно вдосконалюються виробничі процеси, стрімко розвиваються обчислювальна техніка та прикладне програмне забезпечення, відзначається зростання складності інформаційних систем. Це супроводжується появою нових напрямків, технологій і архітектурних рішень для створення, впровадження та розвитку інформаційних систем. Наразі відбувається перехід до динамічної, адаптивної структури інформаційних систем, побудованої на базі розподілених рішень щодо обробки даних. Сучасний етап розвитку суспільства ставить інформаційні технології на одне з провідних стратегічних місць, концентруючи в галузі значні інтелектуальні й фінансові ресурси. Разом із ростом складності задач, що вирішуються завдяки інформаційним системам (ІС), з'являються нові виклики у створенні, модифікації, підтриманні та інтеграції таких систем. Це дозволяє класифікувати ІС за їх масштабністю і функціональними особливостями на три основні класи:

- малі;
- середні;
- великі (корпоративні системи на рівні державних установ).

Малі ІС охоплюють переважно системи для невеликих підприємств. Їхні основні характеристики включають короткий життєвий цикл, орієнтацію на масове використання та низьку вартість. Важливою особливістю є практично відсутність аналітичної обробки даних і можливості модифікацій без участі розробників. Такі ІС здебільшого функціонують на настільних комп'ютерах із одноманітним апаратним і програмним забезпеченням, часто без належних засобів безпеки.

Системи середнього класу мають більш тривалий життєвий цикл і потенціал для розвитку в корпоративні (великі) ІС. Їхніми ключовими характеристиками є можливість аналітичної обробки даних, наявність персоналу для адміністрування апаратного та програмного забезпечення, інтегровані засоби безпеки й тісна співпраця з установами-розробниками програмного забезпечення для підтримки системних компонентів.

Великі (корпоративні) ІС вирізняються масштабністю і складністю задач, які ними вирішуються. Вони мають тривалий життєвий цикл, часто потребують міграції застарілих систем і використовують різноманітне апаратне та програмне забезпечення, яке може мати коротший життєвий цикл порівняно зі створюваною системою. До їх визначальних рис належать територіальна розподіленість, орієнтація на аналітичну обробку даних і здатність працювати в багатьох предметних областях. Таким чином, класифікація інформаційних систем за їх характеристиками дозволяє оптимально організувати процес їх розробки й подальшого використання з огляду на потреби конкретного проекту чи організації. [1-7]

Постановка завдання. Наразі активно розглядаються питання, пов'язані з описом продуктів, технологій та методологій розробки малих і середніх інформаційних систем (ІС). У той же час технології та методології створення великих ІС, які передбачають об'єднання кількох локальних систем, практично ігноруються та не перебувають у фокусі обговорення. Як наслідок, для розробки масштабних ІС часто обираються технології, які первинно не були адаптовані до виконання таких завдань. Це стає причиною того, що реалізовані проекти не отримують належного розвитку та ефективності [1,5-11].

Виклад основного матеріалу. Сучасний етап розвитку суспільства визначає ІТ-індустрію як ключовий і стратегічний напрямок зосередження інтелектуальних і фінансових ресурсів. Інформація разом із засобами її управління, що включають програмні продукти різного функціонального призначення, отримала статус інформаційних ресурсів. Ці ресурси групуються у рамках інформаційних систем. Об'єднання таких ресурсів через інформаційно-комунікаційну взаємодію формує корпоративні інформаційні ресурси, що часто описуються як Єдиний інформаційний простір. Впровадження такого простору на рівні держави, корпорації чи підприємства стає можливим за умови розробки та дотримання стандартів взаємодії між інформаційними системами та їх окремими компонентами (рис. 1) [1, 2, 8–15].

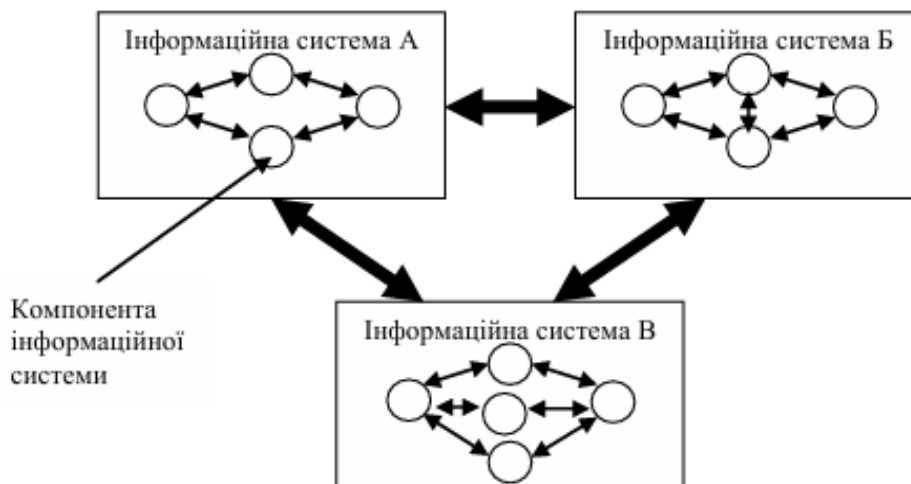


Рисунок 1 – Корпоративні інформаційні ресурси

Джерело: розроблено авторами на підставі [1]

У деяких випадках під ІР розуміються лише дані, коли рішення задачі УІР зводиться до Єдиного простору даних (UDS), рис. 2, а інформаційні системи виступають як клієнт і сервер і взаємодіють один з одним відповідно до схеми, показаної на рис. 3:

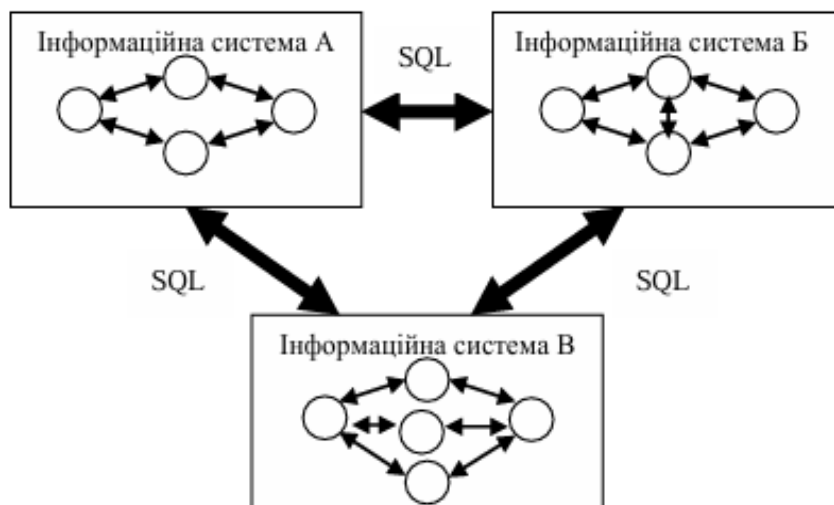


Рисунок 2 - Єдиний простір даних (ЄПД)

Джерело: розроблено авторами на підставі [1]

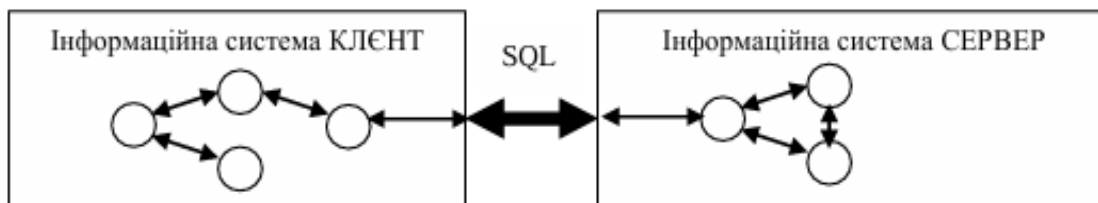


Рисунок 3 - Архітектура доступу до віддалених даних

Джерело: розроблено авторами на підставі [1]

Клієнтська інформаційна система (КІС) надсилає запити до серверної інформаційної системи (СІС) для отримання даних, які потім підлягають подальшій обробці. Як мова запитів використовується SQL, загальноприйнятий стандарт для роботи з реляційними системами управління базами даних. У більшості випадків доступ до віддалених баз даних (БД) здійснюється через продукти, що підтримують протоколи ODBC (Open Database Connectivity) та JDBC (Java Database Connectivity), або через шлюзи [21, с.8], які надаються розробниками СУБД чи сторонніми компаніями.

При побудові єдиного простору даних застосовується архітектура доступу до віддалених даних, що базується на принципах дворівневої клієнт-серверної моделі. Ця архітектура визначає виконання функцій введення і відображення даних, а також прикладних функцій на стороні клієнта, тобто забезпечує методи їхньої обробки. Клієнт передає запити серверу, який обробляє їх та повертає результат у вигляді структурованого блоку даних.

Втім, описаний підхід до взаємодії систем має низку недоліків, що характерні для дворівневої клієнт-серверної архітектури:

- КІС повинна враховувати особливості використовуваної СУБД і структуру віддаленої БД. Це знижує загальний рівень безпеки системи.

- Підтримка та модифікація програм КІС ускладнюються, адже зміни в схемі віддаленої БД на рівні СІС вимагають коригування відповідних програм КІС, що ускладнює обслуговування, оновлення або заміну програмного забезпечення на великій кількості пристроїв.

- Адміністрування бази даних КІС стає надто складним, зокрема управління правами доступу користувачів до неї [8-15].

Розглянутий сценарій має значний недолік - дублювання функцій ІСК (інформаційних систем корпоративного рівня) всередині інших ІСК, що провокує нераціональне використання ресурсів цих систем. Зі зростанням популярності Інтернету та технологій WWW розробники корпоративних ІС приділяють їм дедалі більше уваги. Спочатку WWW створювали як інструмент для забезпечення графічного інтерфейсу в Інтернеті, спрощуючи доступ до інформації, яка розподілена між тисячами комп'ютерів у світі. Її основою стали веб-сторінки, вузли, браузерери та сервери, а навігація стала можливою завдяки гіпертексту (підтримуваному протоколом HTTP) і стандарту HTML.

Впровадження інтерфейсу CGI (Common Gateway Interface) вирішило проблему обміну даними між веб-сервером і програмами, такими як бази даних, які раніше не могли взаємодіяти напряму з браузерами. Це дозволило реалізувати інтерактивність між кінцевим користувачем і веб-прикладними програмами. Інформація, введена користувачем у браузер, оброблялася на сервері, після чого користувач у відповідь отримував згенеровану HTML-сторінку.

Значна частина розробок в Інтернет-середовищі базується саме на цьому підході. З появою мови програмування Java розробники отримали цілком нові інструменти для створення застосунків в Інтернеті. Проте Java не варто розглядати просто як складову WWW-технологій, адже вона охоплює ширший спектр завдань порівняно з технологіями на базі HTML, HTTP та CGI. Можливості WWW суттєво розширили набір рішень, які можна використовувати під час проектування ІС.

Водночас виникає важливе питання: чим є взаємодіючі ІС, створені на основі певних технологій, і чи здатні вони ефективно вирішувати завдання платформи EIP. Це питання постає через роль взаємодії різних компонентів. Наприклад, браузер функціонує як платформа для відображення інформації, тоді як сервер WWW разом із додатками виступає в ролі виконавця логіки системи й управління доступом до даних.

Така модель відповідає дворівневій архітектурі, де сервер виконує інтелектуальні функції (рис. 4) [1, 11-15].

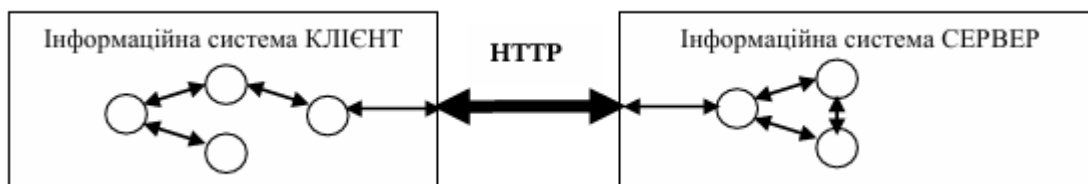


Рисунок 4 – Архітектура з інтелектуальним сервером

Джерело: розроблено авторами на підставі [1]

Технологія WWW може суттєво покращити обмін даними між інформаційно-комп'ютерними системами (ІКС), проте дворівнева архітектура з інтелектуальним сервером має низку суттєвих недоліків. Одним із головних проблем є неможливість перевести процес обробки даних, виконуваний на WWW-сервері, на сторону ІКС. Це пов'язано з тим, що ІКС отримує дані у вигляді HTML-сторінок, що ускладнює організацію їх обробки внутрішніми компонентами системи. Як наслідок, обчислювальні ресурси ІКС використовуються недостатньо ефективно. Додатково виникає суттєва проблема забезпечення безпеки системи, оскільки в середовищі Інтернет відсутнє повністю інтегроване рішення для захисту даних. Це критично для організацій із високими вимогами до безпеки. А також істотно ускладнюється процес адміністрування ресурсів, включаючи управління користувацькими правами доступу.

Концепція єдиного інформаційного простору має передбачати, що інформаційні ресурси ІКС охоплюють не лише дані, а й різноманітні додатки. У межах кожної конкретної ІКС частина методів обробки реалізується у формах додатків, доступних для інших систем. Наприклад, взаємодія між двома ІКС може виглядати так: одна з них отримує вже оброблені дані від другої, після чого ці дані проходять додаткову обробку внутрішніми компонентами першої системи. Такий підхід відповідає принципам розподіленої, peer-to-peer архітектури, де кожен додаток може функціонувати як клієнт або сервер. Це дозволяє спільно вирішувати завдання й уникати дублювання функціональності додатків [1, 19-20].

Розподіл додатків між різними ІКС сприяє оптимізації навантаження між програмними та апаратними ресурсами, що істотно підвищує ефективність використання інформаційної системи. Знання схеми бази даних є необхідним лише для тих додатків, які безпосередньо обробляють дані цієї бази. Завдяки застосуванню сервісів ІКК, які надаються сервером ІС, а також методам обробки даних, можна усунути проблему, пов'язану зі зміною схеми віддаленої бази даних. Статичність інтерфейсів компонентів, що забезпечують набір послуг в рамках ІКС, досягається шляхом використання об'єктно-орієнтованого аналізу та проектування, а також технологій розподілених об'єктів (CORBA, Java, DCOM) на різних етапах створення інформаційної системи.

Більшість ІС використовуються як додатки в контексті дворівневої клієнт-серверної архітектури. Для зв'язку між клієнтом і сервером часто застосовуються механізми процедурного виклику, які не завжди відповідають загальним стандартам. Їх реалізація тісно пов'язана з ядром системи управління базами даних, що потребує значних обчислювальних ресурсів на стороні сервера. Зі збільшенням робочого навантаження на сервер клієнт-серверні системи в рамках дворівневої архітектури починають більше нагадувати великі комп'ютери (мейнфрейми). При цьому

оброблювані дані та методи їх представлення стають недоступними для використання іншими додатками. Взаємодія такого клієнт-сервера найчастіше організовується через засоби СУБД, що створює додаткове навантаження на серверну частину. [1, 16-20].

Висновки. Сучасні технології відкривають можливість формування інтегрованого середовища як у складі інформаційних систем (ІС), так і в межах концепції ЕІР, яке характеризується такими особливостями [1,2,20]:

- незалежність від апаратного забезпечення та системного програмного забезпечення; – спираються на міжнародні та галузеві стандарти;
- створення єдиної інформаційної моделі, що представляє підприємство як сукупність керованих ресурсів та діяльності, орієнтованої на реалізацію правил управління спільною роботою для кожної конкретної організації;
- забезпечення розширюваності системи з можливістю легко додавати нові компоненти до вже існуючої ІС;
- інтеграція старих додатків (legacy applications) у сучасні інформаційні системи;
- підтримка природної інтеграції нової ІС, що гарантує її життєздатність і поступовий еволюційний розвиток;
- накопичення, розповсюдження і розвиток формалізованих знань спеціалістів;
- істотне скорочення загальних витрат на проектування і впровадження ІС.

Список літератури

1. Минайленко Р.М. Паралельні та розподілені обчислення: навч. посіб. Кропивницький: Видавець Лисенко В. Ф., 2021. 153 с.
2. Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems / R. Braun, H. Siegel et al. *Parallel and Distributed Computing*. 2001. Vol. 61, No. 6, P. 810–837.
3. Heuristics for Scheduling Parameter Sweep Applications in Grid Environments / H. Casanova, A. Legrand et al. *Heterogeneous Computing Workshop (HCW'00): Proceedings of the 9th Workshop (Cancun, Mexico, May 1, 2000)*. IEEE Computer Society, 2000. P. 349–363.
4. Task Scheduling Algorithm in GRID Considering Heterogeneous Environment / S.Y. You, H.Y. Kim et al. *Parallel and Distributed Processing Techniques and Applications (PDPTA '04): Proceedings of the International Conference (Nevada, USA, June 21–24, 2004)*. CSREA Press, 2004. Vol. 1. P. 240–245.
5. New Grid Scheduling and Rescheduling Methods in the GrADS Project / Cooper, A. Dasgupta et al. *International Parallel and Distributed Processing Symposium (IPDPS'04): Proceedings of the 18th International Symposium (Santa Fe, New Mexico USA, April 26–30, 2004)*. IEEE Computer Society, 2004. P. 199–206.
6. Improving Grid Level Throughput Using Job Migration And Rescheduling / K. Kurowski, B. Ludwiczak et al. *Scientific Programming*. 2004. Vol. 12, No. 4. P. 263–273.
7. A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid / A. Takefusa, S. Matsuoka et al. *High Performance Distributed Computing (HPDC-10): Proceedings of the 10th IEEE International Symposium (San Francisco, California, USA, August 7–9, 2001)*. IEEE Computer Society, 2001. P. 406–415.
8. Distributed Dynamic Scheduling of Composite Tasks on Grid Computing Systems / H. Chen, M. Maheswaran. *International Parallel and Distributed Processing Symposium (IPDPS 2002): Proceedings of the 16th International Symposium (Fort Lauderdale, FL, USA, April 15-19, 2002)*. IEEE Computer Society, 2002. P. 88–97.
9. A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids / N. Muthuvelu, J. Liu et al. *rid Computing and e-Research (AusGrid 2005): Proceedings of the 3rd Australasian Workshop (Newcastle, NSW, Australia, January 30 – February 4, 2005)*. Australian Computer Society, 2005. P. 41–48.
10. Scheduling in Heterogeneous Grid Environments: The Effects of Data Migration / H. Shan, L. Olikier et al. *Advanced Computing and Communication (ADCOM 2004): Proceedings of the 12th IEEE International Conference (Ahmedabad Gujarat, India, December 15–18, 2004)*. IEEE Computer Society, 2004. P. 1–8.

11. Dong, F., Akl, S.G. Scheduling algorithms for grid computing: State of the art and open problems. Technical Report No. 2006-504 . Queen's University, Canada, 2006. P. 55.
12. Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests / V. Subramani, R. Kettimuthu et al. *High Performance Distributed Computing (HPDC 2002)*: Proceedings of 11th IEEE Symposium (Edinburgh, Scotland, July 23–26, 2002). — IEEE Computer Society, 2002. — P. 359–366.
13. El-Rewini, H., Lewis, T., Ali, H. Task Scheduling in Parallel and Distributed Systems . Prentice Hall, 2010. 290 p.
14. Radulescu, A., Gemund, A.J.C. On the Complexity of List Scheduling Algorithms for Distributed Memory Systems . *Supercomputing (SC'99)*: Proceedings of 13th International Conference (Portland, Oregon, USA, November 13–19, 1999). IEEE Computer Society, 1999. P. 68–75.
15. Sakellariou, R., Zhao, H. A Low-cost Rescheduling Policy for Efficient Mapping of Workflows on Grid Systems . *Scientific Programming*. 2017. Vol. 12, No. 4. P. 253–262.
16. Darbha, S., Agrawal, D.P. Optimal Scheduling Algorithm for Distributed Memory Machines . *IEEE Transactions on Parallel and Distributed Systems*. 1998. Vol. 9, No. 1. P. 87–95.
17. Ranaweera, S., Agrawal, D.P. A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems . *International Parallel and Distributed Processing Symposium (IPDPS'00)*: Proceedings of 14TH International Symposium (Cancun, Mexico, May 1–5, 2018). IEEE Computer Society, 2005. P. 445–450.
18. Bajaj, R., Agrawal, D.P. Improving Scheduling of Tasks in A Heterogeneous Environment . *IEEE Transactions on Parallel and Distributed Systems*. 2004. Vol. 15, No. 2. P. 107–118.
19. Yang, T. DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors . *IEEE Transactions on Parallel and Distributed Systems*. 1994. Vol. 5, No. 9. P. 951–967.
20. Liou, J. A., Palis, M.A. Comparison of General Approaches to Multiprocessor Scheduling . *International Parallel Processing Symposium (IPPS '97)*: Proceedings the 11th International Symposium (Geneva, Switzerland, April 1–5, 1997). IEEE Computer Society, 1996. P. 152–156.
21. Сучасні інтелектуальні технології функціональної медичної діагностики / О.Г. Аврунін, Є.В. Бодяньський, М.В. Калашник та ін. Харків: ХНУРЕ, 2018. 236 с.

References

1. Mynaylenko, R.M. (2021). *Parallel and distributed computing: a textbook*. Kropyvnytskyi: Publisher Lysenko V. F. [in Ukrainian].
2. Braun, R. A., Siegel, H. et al. (2001). Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems . *Parallel and Distributed Computing*. Vol. 61, No. 6, P. 810–837.
3. Casanova, H., Legrand, A. et al. (2000). Heuristics for Scheduling Parameter Sweep Applications in Grid Environments . *Heterogeneous Computing Workshop (HCW'00)*: Proceedings of the 9th Workshop (Cancun, Mexico, May 1, 2000). IEEE Computer Society, P. 349–363.
4. You, S.Y., Kim, H.Y. et al. (2004). Task Scheduling Algorithm in GRID Considering Heterogeneous Environment . *Parallel and Distributed Processing Techniques and Applications (PDPTA '04)*: Proceedings of the International Conference (Nevada, USA, June 21–24, 2004). CSREA Press, Vol. 1. P. 240–245.
5. Cooper, K., Dasgupta, A. et al. (2004). New Grid Scheduling and Rescheduling Methods in the GrADS Project . *International Parallel and Distributed Processing Symposium (IPDPS'04)*: Proceedings of the 18th International Symposium (Santa Fe, New Mexico USA, April 26–30, 2004). IEEE Computer Society, P. 199–206.
6. Kurowski, K., Ludwiczak, B. et al. (2004). Improving Grid Level Throughput Using Job Migration And Rescheduling . *Scientific Programming*. Vol. 12, No. 4. P. 263–273.
7. Takefusa, A., Matsuoka, S. et al. (2001). A Study of Deadline Scheduling for Client-Server Systems on the Computational Grid . *High Performance Distributed Computing (HPDC-10)*: Proceedings of the 10th IEEE International Symposium (San Francisco, California, USA, August 7–9, 2001). IEEE Computer Society, P. 406–415.
8. Chen, H., & Maheswaran, M. (2002). Distributed Dynamic Scheduling of Composite Tasks on Grid Computing Systems . *International Parallel and Distributed Processing Symposium (IPDPS 2002)*: Proceedings of the 16th International Symposium (Fort Lauderdale, FL, USA, April 15-19, 2002). IEEE Computer Society, P. 88–97.
9. Muthuvelu, N. A, Liu, J. et al. (2005). Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids . *Grid Computing and e-Research (AusGrid*

- 2005): Proceedings of the 3rd Australasian Workshop (Newcas-tle, NSW, Australia, January 30 – February 4, 2005). Australian Computer Society, P. 41–48.
10. Shan, H., Olikier, L. et al. (2004). Scheduling in Heterogeneous Grid Environments: The Effects of Data Migration . *Advanced Computing and Communication (ADCOM 2004)*: Proceedings of the 12th IEEE International Conference (Ahmedabad Gujarat, India, De-cember 15–18, 2004). IEEE Computer Society, P. 1–8.
 11. Dong, F., & Akl, S.G. (2006). Scheduling algorithms for grid computing: State of the art and open problems. *Technical Report*, No. 2006-504 , Queen’s University, Canada, P. 55.
 12. Subramani, V., Kettimuthu, R. et al. (2002). Distributed Job Scheduling on Computational Grids using Multiple Simultaneous Requests . *High Performance Distributed Computing (HPDC 2002)*: Proceedings of 11th IEEE Symposium (Edinburgh, Scotland, July 23–26, 2002). IEEE Computer Society, P. 359–366.
 13. El-Rewini, H., Lewis, T., & Ali, H. (2010). Task Scheduling in Parallel and Distributed Systems . Prentice Hall, 290 p.
 14. Radulescu, A., & Gemund, A.J.C. (1999). On the Complexity of List Scheduling Algorithms for Distributed Memory Systems . *Supercomputing (SC'99)*: Proceedings of 13th International Conference (Portland, Oregon, USA, November 13–19, 1999). IEEE Computer Society, P. 68–75.
 15. Sakellariou, R., & Zhao, H. (2017). A Low-cost Rescheduling Policy for Efficient Mapping of Workflows on Grid Systems . *Scientific Programming*. Vol. 12, No. 4. P. 253–262.
 16. Darbha, S., & Agrawal, D.P. (1998). Optimal Scheduling Algorithm for Distributed Memory Machines . *IEEE Transactions on Parallel and Distributed Systems*. Vol. 9, No. 1. P. 87–95.
 17. Ranaweera, S., & Agrawal, D.P. (2005). A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems . *International Parallel and Distributed Processing Sym-posium (IPDPS'00)*: Proceedings of 14TH International Symposium (Cancun, Mexico, May 1–5, 2018). IEEE Computer Society, P. 445–450.
 18. Bajaj, R., & Agrawal, D.P. (2004). Improving Scheduling of Tasks in A Heterogeneous Environment . *IEEE Transactions on Parallel and Distributed Systems*. Vol. 15, No. 2. P. 107–118.
 19. Yang, T., & Gerasoulis, A. (1994). DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors . *EEE Transactions on Parallel and Distributed Systems*. Vol. 5, No. 9. P. 951–967.
 20. Liou, J. A., & Palis, M.A. (1996). Comparison of General Approaches to Multiprocessor Scheduling . *International Parallel Processing Symposium (IPPS '97)*: Proceedings the 11th International Symposium (Geneva, Switzerland, April 1–5, 1997). IEEE Computer Society, P. 152–156.
 21. Avrunin, O.H., Bodianskyi, Ye.V., Kalashnyk, M.V., Semenets, V.V., & Filatov, V.O. (2018). *Modern intelligent technologies of functional medical diagnostics*. Kharkiv: KhNURE [in Ukrainian].

Roman Minailenko, Assoc. Prof., PhD tech. sci., **Oksana Konoplitska-Slobodeniuk**, **Iryna Lysenko**, PhD tech. sci.

Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

Distributed Object Technologies in Information Systems

The article shows that the development of information and telecommunications technologies is at a stage when in a distributed information and telecommunications environment, not only the need for access and exchange of information, but also the implementation of various types of analysis and processing of this information is becoming increasingly important. The widespread introduction of computer technologies into all types of activities, the constant increase in their computing power, the use of computer networks of various scales requires the use of a significant amount of high-performance distributed computing, which in turn leads to a shortage of computing resources when performing various computing processes. An effective way to solve these problems is to use parallel and distributed computing.

Currently, issues are being discussed regarding the description of products, technologies and methodologies for creating small and medium-sized information systems. At the same time, technologies and methodologies for building large information systems, which combine a set of local information systems, are practically not considered and discussed. The consequence of this is that even at the design stage, technologies for creating a large information system are chosen that do not meet the requirements. For this reason, the projects being implemented do not receive proper development.

The modern level of development of society defines the IT industry as a leading and strategic direction of concentration of intellectual and financial resources. Information and tools for its management (software products of various functional purposes) have acquired the status of information resources (IR). The latter are concentrated within the framework of IS. The unification of resources on the basis of information and communication interaction of IS brings them to the level of corporate information resources. This unification is often called the Unified Information Space (UIS). The implementation of UIS at the level of the state,

corporation, enterprise is possible in the case of the creation with subsequent observance of the standard for the interaction between IS and their individual applications

In the concept of a single information space, it is necessary to provide that the information resources of the IS, in relation to it, act both as data and as various IS applications. Then, in each of the ISs, part of the data processing methods is implemented as applications accessible from other ISs, in particular, in the case of interaction of two ISs, the first is used by services provided by the second, as a result of which it receives already processed data that can be subjected to further processing by the components of the first IS. This approach corresponds to a distributed, peer-to-peer architecture of interaction. According to this architecture, any applications from different ISs can act as both a client and a server in relation to each other, jointly solving certain tasks. This approach minimizes duplication of applications. The distribution of applications across different ISs makes it possible to achieve an optimal balance of application loading and hardware, which will lead to the effective use of information resources of the systems as a whole.

Modern technologies make it possible to create an integrated environment within the IS, and within the framework of the EIP concept, which has the following properties:

- does not depend on hardware and system software;
- is based on international and industrial standards;
- allows you to develop a single information model of representing an enterprise as a set of managed resources and activity flows, configured to implement the rules for managing the collective activities of each specific enterprise;
- ensures system extensibility, i.e. simplicity and ease of adding new components to existing IS;
- allows you to integrate old applications (legacy applications) into new IS;
- assumes natural integration of created IS, which guarantees viability and evolutionary development;
- allows you to accumulate, replicate and develop formalized knowledge of specialists;
- significantly reduces the total costs of creating IS.

computer, distributed computing, information systems, computing resources

Одержано (Received) 12.09.2025

Прорецензовано (Reviewed) 30.10.2025

Прийнято до друку (Approved) 23.12.2025