

## **ВИБІР ТЕХНОЛОГІЇ ДЛЯ СТВОРЕННЯ WEB-СИСТЕМ РІЗНОГО РІВНЯ СКЛАДНОСТІ**

*Анотація:* Розглядається проблема вибору технології для розробки web-систем різного рівня складності. Оптимізація вибору технології розробки в залежності від властивостей, якими повинна володіти система, що розробляється. Запобігання надлишковим витратам матеріальних ресурсів. Мінімізація вимог до апаратного забезпечення.

*Ключові слова:* Web-системи, оптимізація, технології розробки, апаратне забезпечення.

### **Вступ**

На даному етапі розвитку мережі Інтернет та технологій, що використовуються для створення web-сайтів, замало говорити про web-сайт як про простий монолітний чи модульний програмний продукт, а більш доцільним є використання поняття web-систем [1].

Web-система – програмний комплекс, що побудований на базі клієнт-серверної чи клієнт-сервісної архітектури, при якому передача даних між клієнтом і сервером відбувається по протоколу обміну даними HTTP.

При розробці web-систем найперша проблема, що виникає – це вибір технології розробки і подальшого функціонування системи. При цьому слід враховувати ряд факторів:

1. складність системи (об'ємність і складність структури програмних та апаратних засобів серверної та клієнтської частини web-системи);
2. завантаженість системи (визначається кількістю клієнтських звернень та часом виконання одного звернення до сервера);
3. гнучкість системи (здатність системи до модифікації та розширення без значних затрат людських та грошових ресурсів, а також без значних змін у програмному коді системи);
4. розподіл обчислювальних ресурсів (алгоритм, за яким система розподіляє серверні ресурси на виконання клієнтських запитів);
5. затрати грошових та людських ресурсів на розробку системи.

В залежності від цих факторів потрібно вибирати технологію, а також архітектуру створюваної системи.

### **Постановка задачі**

При створенні web-системи необхідно в залежності від категорії складності архітектури та системи в цілому обрати найбільш оптимальну технологію розробки цієї системи. Наведемо декілька поширених технологій та прокласифікуємо web-системи за складністю (від найпростіших до більш складних):

1. сайти – візитки (<http://vikka.ck.ua>, <http://ble.in.ua> та ін.);
2. бізнес – сайти (<http://medcurator.ru> та ін.);
3. сайти електронної комерції (інтернет - магазини) (<http://rozetka.com.ua> та ін.);
4. інформаційні портали (<http://ukr.net> та ін.);
5. корпоративні інтегровані інформаційні системи (автоматизація бізнес-процесів);

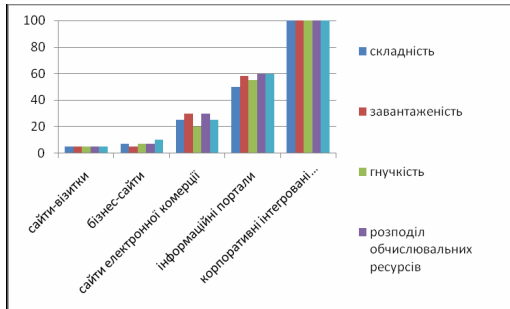


Рис. 1 – Гістограма розподілу поширених web-систем за критерієм впливу факторів

На рис. 1, наведена гістограма розподілу web-систем за їх властивостями, врахування яких є необхідним при безпосередній розробці web-систем.

### ***Сайти-візитки та бізнес-сайти***

Дана категорія web-систем охоплює собою прості монолітні web-системи (рис. 2), що складаються з одної чи декількох web-сторінок. Такі системи не потребують виконання складних обчислювальних задач та не потребують значних ресурсів для виконання серверних та клієнтських програм, а в деяких випадках взагалі є статичними web-сторінками.

Виходячи з цього, можна зробити висновок, що такі системи є монолітними (тобто не потребують концентрації уваги на гнучкості). Дана категорія web-систем охоплює системи:

- які не є складними в розробці;
- не потребують значних затрат грошових та людських ресурсів на їх створення;
- не створюють завантаженості на сервер (причиною є низька відвідуваність та надзвичайно малий час виконання програм на сервері).

І як наслідок, вони не потребують контролю за розподілом ресурсів сервера. Тому для такого типу систем найбільш доцільним вибором є технологія CGI інтерфейсу, оскільки вся розробка зводиться до вкраплення

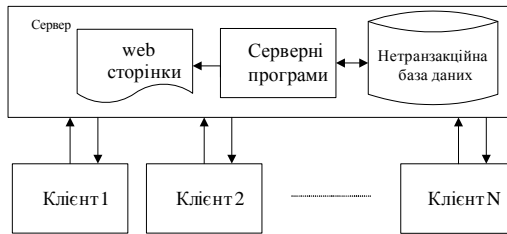


Рис. 2 – Типова архітектура веб систем категорії сайт-візитка та бізнес-сайт

в HTML код коду відповідної серверної мови програмування для роботи з базою даних (БД) [3].

Технологія CGI (з англ. Common gateway interface – “загальний інтерфейс шлюзу”) - стандартний інтерфейс, який використовується для організації взаємодії програми web сервера із зовнішньою програмою (шлюзом). Інтерфейс розроблений таким чином, щоб можна було використовувати будь-яку мову програмування.

З метою оптимізації використання обчислювальних ресурсів сервера при виборі СУБД головним чинником в даному випадку є ресурсна ємність, оскільки транзакції на запис даних в базу даних проводяться рідко. Тому слід обирати систему управління базою даних (СУБД) без підтримки технології контролю транзакцій.

Найбільш популярними мовами програмування, що застосовуються для розробки web-систем на базі CGI є: PHP, Perl, Python, Ruby. Найбільш популярними СУБД без підтримки контролю транзакцій на сьогоднішній день є: SQLite3 та MySQL на базі технології доступу до даних MySQL-SAM.

### **Сайти електронної комерції (інтернет - магазини) та інформаційні портали**

Даний клас web-систем (рис. 2) охоплює системи, які за своєю архітектурою можуть бути як монолітними, так і модульними. Такі системи потребують проведення на сервері більш складних операцій з даними, ніж попередня категорія. Найбільш характерними операціями для даного типу систем є сортування масивів даних (списки, кортежі, словники). Такі системи потребують використання СУБД, з підтримкою ACID - стандарту проведення транзакцій над даними для уникнення випадків втрати даних, оскільки передбачають високий рівень відвідуваності, що, відповідно, провокує значне збільшення звернень до БД [5].

*Абревіатура ACID* описує необхідні властивості транзакції в СУБД :

- *Atomicity (атомарність)* – транзакція є найменшим, неподільним блоком алгоритму зміни даних;
- *Consistency (відсутність протиріч)* – дані знаходяться в стані відсутності протиріч на всьому проміжку часу від початку транзакції до її закінчення;

- *Isolation (ізоляція)* – під час виконання транзакції інші процеси не повинні бачити дані в проміжному стані;
- *Durability (тривалість)* – незалежно від проблем на нижніх рівнях (для прикладу, знеструмлення системи чи збоїв в обладнанні) зміни, що виконані успішно завершеною транзакцією, залишаться збереженими після повернення системи в роботу.

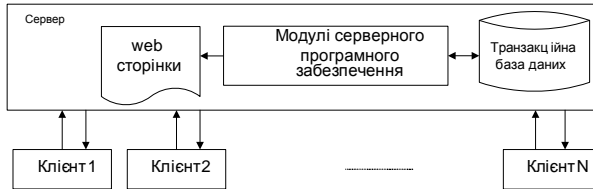


Рис. 3 – Типова архітектура веб систем електронної комерції та інформаційних порталів

Складність таких систем вимагає збільшення грошових затрат та людських ресурсів на розробку системи, тому на вибір технології створення системи впливає можливість повторне використання коду (підхід у програмуванні, при якому одні і ті ж блоки програмного коду використовуються без змін для вирішення різних задач). Таку можливість реалізують технології на базі об'єктно-орієнтованих мов програмування.

*Об'єктно - орієнтовані мови програмування* – мови програмування, що підтримують парадигму об'єктно-орієнтованого програмування, яке базується на поняттях класів, об'єктів та використовує три наступні принципи: *інкапсуляція, наслідування, поліморфізм* [2].

При розробці систем такої категорії слід звернути увагу на гнучкість систем, що розробляються. Гнучкості можна досягнути використовуючи модульне програмне забезпечення на стороні сервера (замість монолітного програмного коду слід розбивати код на модулі, які можна додавати). Таким чином систему можна буде модифікувати та розширювати без значних змін в початковому коді.

Важливим фактором вибору технології для розробки даного класу систем є оптимізація розподілу серверних обчислювальних ресурсів з метою зменшення навантаження на сервер. В даному випадку використання технології CGI веде за собою значне використання обчислювальних ресурсів, оскільки для кожного користувача в оперативній пам'яті сервера створюється окрема копія виконуваної програми (тобто в пам'ять для кожного користувача завантажується інтерпретатор мови програмування, що в свою чергу виконує програмний код на боці сервера). Вибір цієї технології не є оптимальним. Для покращення ситуації, що складається з використанням CGI слід розглянути Fast CGI (це технологія, що з'явилась як розвиток технології CGI. Різниця в тому, що при використанні

даної технології, сервер не створює для кожного користувача копію програми в пам'яті та WSGI (з англ. Web server gateway interface – стандарт взаємодії між web-сервером та програмою, що виконується на сервері, при якому програма виконується під управлінням web-сервера, написаного на даній мові програмування, але не є повноцінним для використання як виробничий сервер) [4].

Перевагами даних технологій є те, що Fast CGI при зверненні клієнта до сервера, запускає інтерпретатор і виконує скрипти для кожного користувача послідовно, не запускаючи окрему копію інтерпретатора для кожного користувача, а WSGI є багато потоковим додатком, який виконує програму для кожного користувача в окремому потоці. Такі два підходи значно прискорюють роботу і, таким чином, знімають навантаження з сервера.

Тому оптимальним вибором для даного класу систем є мови: PHP, Perl, Python, Ruby на базі fastCGI, фреймворки на базі мови програмування Python: “web2py”, “Django”, “TurboGears”, “Pylons”, фреймворк на базі мови програмування Ruby: “Ruby on rails”. Оптимальними рішеннями для зберігання даних є СУБД MySQL на базі технології збереження даних InnoDB, MS SQL Server, ZODB.

**Корпоративні інтегровані інформаційні системи (автоматизація бізнес-процесів та інформаційних потоків)**

Корпоративні інтегровані інформаційні системи (рис. 4) є web-системами найвищого рівня складності і є власне системами в кожному випадку унікальними, тому для кожної з них варто обрати окремий підхід.

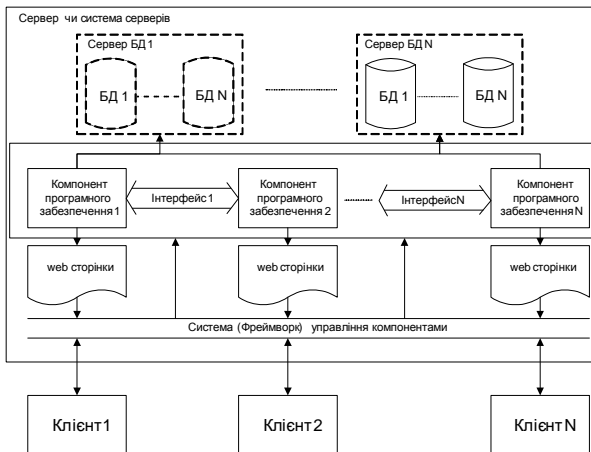


Рис. 4 – Типова архітектура корпоративних web-систем для автоматизації бізнес-процесів та виробничих процесів

Системи такого класу призначені для автоматизації інформаційних потоків повного циклу виробництва: від створення деталей чи окремих частин продукції до виконання періодичних звітів та ведення статистики.

Очевидно, що складність таких систем надзвичайно велика і в кожному окремому випадку, в залежності, від розмірів підприємства чи групи підприємств. Розробка такої системи потребує значних затрат виробничих ресурсів. При виборі архітектури серверного програмного забезпечення обов'язково потрібно враховувати цей фактор. Очевидно, що модульність серверного програмного забезпечення та об'єктно-орієнтований підхід не підходить, оскільки в даному випадку кількість модулів може досягати сотень, а то і тисяч. В даному випадку необхідно використати компонентну архітектуру на базі інтерфейсів, яка дозволяє створювати окремі компоненти, що взаємодіють між собою згідно визначених інтерфейсів і є окремими незалежними частинами програмного забезпечення, що можуть існувати і працювати в різних середовищах без зміни їх програмного коду. Такий підхід дозволяє структурувати програмне забезпечення, а також, у порівнянні з об'єктно-орієнтованим підходом, підвищити ефективність повторного використання коду (застосуванням одного і того ж компонента для виконання різних задач).

Завдяки створенню компонентної архітектури значно росте гнучкість системи, що створюється. В будь-який момент можна змінити налаштування всієї системи шляхом підключення, відключення чи модифікації окремих компонентів без порушення функціонування всієї системи.

Для наочності наведемо декілька порівнянь за вищеозначеними факторами.

#### *Порівняння за фактором гнучкості*

Розглянемо порівняння технологій розробки web-систем з точки зору їхньої гнучкості. Припустимо, що виконано наступні завдання з використанням вищеозначених технологій:

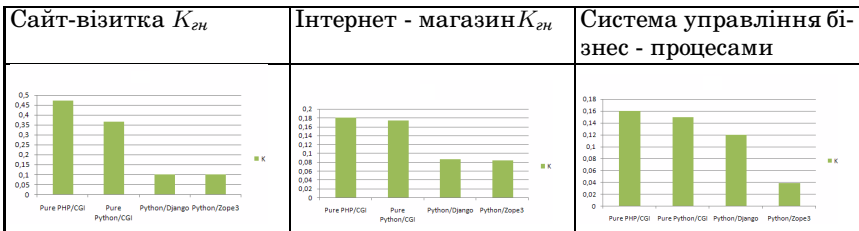
1. *Створити сайт-візитку.*
2. *Створити інтернет - магазин.*
3. *Створити корпоративну систему управління бізнес-процесами.*

Припустимо, що необхідно модифікувати створений додаток, абсолютно переробивши систему пошуку та інтерфейсну частину web-сторінок з результатами пошуку по сайту. Введемо для визначення впливу фактора гнучкості коефіцієнт гнучкості:  $K_{ен} = \frac{m}{n}$ , де  $m$  – кількість рядків коду, які необхідно змінити/додати для реалізації необхідної модифікації;  $n$  – кількість рядків ефективного коду вихідного додатку. Таким чином, отримаємо наступні результати для різних технологій та складності типів web-систем (табл. 1):

Виходячи зі статистик на ринку трудових ресурсів в галузі розробки web-додатків, і прийнявши за одиницю затрати при розробці системи на PHP (спеціалістів з даної технології на ринку найбільше і вартість їх найнижча), отримаємо наступну нормалізовану гістограму:

	Сайт-візитка		Інтернет - магазин		Система управління бізнес - процесами	
	$n$	$m$	$n$	$m$	$n$	$m$
Pure PHP/CGI	1800	850	5000	900	50000	8000
Pure Python/CGI	1500	550	4600	800	40000	6000
Python/Django	100	10	3500	300	25000	3000
Python/Zope3	80	8	3000	250	18000	700

*Порівняння за фактором затрат на розробку грошових та людських ресурсів*



*Порівняння за фактором завантаженості (кількість опрацьованих запитів за секунду)*



При створенні гістограми (рис. 5) було використано багато потоковий клієнтський скрипт, що виконував однаковий запит на вибірку запису з бази даних по первинному ключу. Клієнт і сервер знаходились на одній і тій же робочій станції з наступними характеристиками апаратного забезпечення: процесор - AMD Athlon X2 5600+ (2 ядра по 2.7 ГГц); оперативна пам'ять - Samsung (2048 Мб, 866 МГц); вторинна пам'ять - Seagate (200 Гб, 5400 об/хв.).

Оптимальний розподіл обчислювальних ресурсів системи є одним з найважливіших факторів, оскільки такі системи виконують складні об-

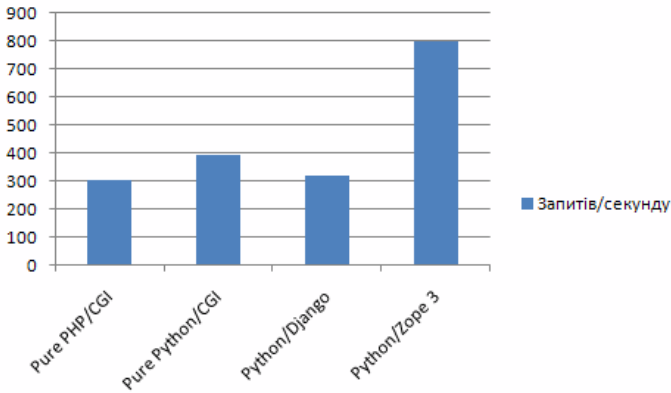


Рис. 5 – Гістограма завантаженості (кількість опрацьованих запитів за секунду)

числення, що спричиняє велику завантаженість серверної частини. Такі технології як CGI та Fast CGI не здатні забезпечувати оптимальний розподіл ресурсів зручним способом, тому при створенні корпоративних систем доцільно використовувати готову систему управління компонентами (фреймворк) чи створити свою. Очевидно, що для реалізації поставленого завдання найбільш доцільним рішенням є використання багато потоковості, що дозволяє управляти виділенням ресурсів для конкретних класів задач. Найбільш розвиненими і актуальними фреймворками, що реалізують компонентний підхід для створення web-систем є Java 2 Enterprise Edition, Zope3 та Microsoft .NET.

### Висновок

Розглянутий в даній статті підхід до вибору технології для створення web-систем визначеної категорії є актуальним, адже коректний вибір технології найчастіше є запорукою успіху оптимальної реалізації проекту. Саме врахування основних властивостей, якими володіють web-системи.

Технологію для розробки web-систем необхідно обирати виходячи з аналізу багатьох факторів. Такий комплексний підхід допомагає скоротити час та ресурси на розробку.

### Література

1. Дзінько Р.І., Лісовиченко О.І., Рибенко Ю.А. Підхід до розробки засобів інтеграції web-орієнтованих інформаційних систем, створених на базі різних технологій // Адаптивні системи автоматичного управління. -2008.- 13(33).-С.29-37.
2. Концептуальний підхід до розробки інтегрованого об'єктно-орієнтованого середовища для моделювання ГВС / О.В. Блажко, В.В. Борисюк, А.А. Гілязов, А.С. Птічнікова, О.І. Лісовиченко // Науковий



вісник Інституту економіки та нових технологій ім. Ю.І. Кравченка  
“Нові технології”.-Кременчук.- 2003.- Вип.1'(2).- С.81-85.

3. Anthony A. Apodaca, Larry Gritz Advanced RenderMan: Creating CGI for Motion Pictures.- Morgan Kaufmann: San Diego, 1999.- 512p. ISBN: 1558606181
4. James Gardner The Definitive Guide to Pylons.-Apress 2008.- 568p. ISBN: 1-59059-934-9
5. A component based services architecture for building distributed applications Bramley, R. Chiu, K. Diwan, S. Gannon, D. Govindaraju, M. Mukhi, N. Temko, B. Yechuri, M. 2000. Department. of Computer. Science., Indiana University, Bloomington, Indiana, USA

Отримано 08.12.2009 р.