

М. В. ТКАЧУК, д-р техн. наук, проф. каф. «АСУ», НТУ «ХПІ»;
О. В. ВЕКШИН, аспірант, каф. «АСУ», НТУ «ХПІ»;
В. В. КОСЕНКО, студент, каф. «АСУ», НТУ «ХПІ»

ДЕЯКІ ПРОБЛЕМИ РОЗРОБКИ АДАПТИВНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОБІЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПІДХІД ДО ЇХ ВИРІШЕННЯ

Розглянуті деякі особливості мобільних інформаційних систем та визначена актуальність проблеми розробки адаптивних програмних рішень для забезпечення їх ефективного функціонування. Запропоновано підхід до вирішення цієї задачі на основі застосування методу аналізу прецедентів, розроблено алгоритм адаптивного управління розміром кеш-пам'яті мобільного пристрою та наведена архітектура його програмних компонентів на платформі Android.

Рассмотрены некоторые особенности мобильных информационных систем и обоснована актуальность проблемы разработки адаптивных программных решений для обеспечения их эффективного функционирования. Предложен подход к решению этой задачи на основе использования метода анализа прецедентов, разработан алгоритм адаптивного управления размером кеш-памяти мобильного устройства и приведена архитектура его программных компонентов на платформе Android.

Some features of mobile information systems are considered and problem actuality of adaptive software solutions for their effective functioning is defined. Case-based reasoning approach to solve this task is proposed, the adaptive control algorithm for cash-memory of mobile device is elaborated, and its software components architecture on Android platform is given.

1. Актуальність розробки адаптивного програмного забезпечення для мобільних систем. Мета дослідження. На даний момент мобільні інформаційні системи (МІС) набувають все більшої популярності, тому що вони надають користувачам можливість доступу до інформації з будь-якого місця та у будь-який час. Прикладом таких систем можуть бути соціальні мережі: Facebook (www.facebook.com), vkontakte (www.vkontakte.ru), twitter (www.twitter.com), системи керування завданнями користувачів: Due Today [1], Shuffle [2] та деякі інші.

МІС мають певні особливості, що відрізняють їх від desktop- та web-орієнтованих інформаційних систем. Такими особливостями є наступні: відносно невелика обчислювальна потужність мобільних пристроїв, вельми малий розмір дисплеїв їх терміналів, обмежений час роботи від акумулятору [3]. Тому для підвищення ефективності використання МІС необхідно розробляти відповідне програмне забезпечення (ПЗ) із вбудованим механізмом адаптації, який буде, в залежності від стану зовнішнього середовища, змінювати певні параметри функціонування компонентів ПЗ МІС з метою підтримки їх у працездатному стані.

Одним з найбільш поширених типів МІС є системи керування завданнями (СКЗ). До їх типової функціональності належать такі можливості як: формування та обробка списків задач користувача, синхронізація даних з віддаленим сервером, можливість додання різноманітних файлів до створених задач і деякі ін., які, наприклад, визначені в рекомендаціях групи розробників СКЗ: Getting Things Done (GTD) [4]. Таким чином, такі системи можуть накопичувати великі обсяги інформаційних ресурсів, і тому в умовах жорстко обмежених апаратних ресурсів МІС постає задача адаптивного управління СКЗ для підвищення продуктивності використання таких систем.

Проаналізувавши функціональність та архітектуру деяких існуючих СКЗ, можна зробити висновок, що ці системи надають всю необхідну функціональність з керування завдань, але жодна з них не використовує механізмів адаптивного управління. Тому метою досліджень, які представлені в цій статті, є розробка та використання механізму адаптивного управління ресурсами типової СКЗ, на прикладі сервісу Google Tasks (<http://mail.google.com/tasks>).

2. Аналіз типової архітектури мобільної СКЗ з використанням сервісу Google Tasks. Спираючись на результати, отримані при аналізі існуючих мобільних СКЗ, була проаналізована загальна архітектура клієнт-серверної мобільної СКЗ, що подана на рис. 1 у вигляді діаграми розгортання її компонентів в нотатції UML 2.0. Основними вузлами в цій системі є: сервер Google та мобільний пристрій (клієнтське застосування) на платформі мобільної операційної системи (МОС) Android [5]. При цьому на стороні мобільного пристрою наявні такі програмні компоненти:

1. «Менеджер облікових записів» – компонент МОС Android, що надає засоби для зберігання та доступу до облікових записів користувачів;
2. «Менеджер задач» – безпосередньо клієнтське застосування, що надає графічний інтерфейс користувача для роботи з задачами;
3. «База даних SQLite» – реляційна локальна база (БД) даних СКЗ;
4. «Сервіс синхронізації» – фоновий процес, що забезпечують синхронізацію локальної БД з віддаленим сервісом Google;
5. «Бібліотека Google Tasks API» – сукупність засобів для обміну даними з сервісом Google Tasks.

На серверній стороні МІС розгорнуті наступні компоненти:

1. «Google Tasks API» – компонент, що надає засоби для взаємодії з централізованою БД задач;
2. «Сервіс Google Task» – централізована БД задач СКЗ.

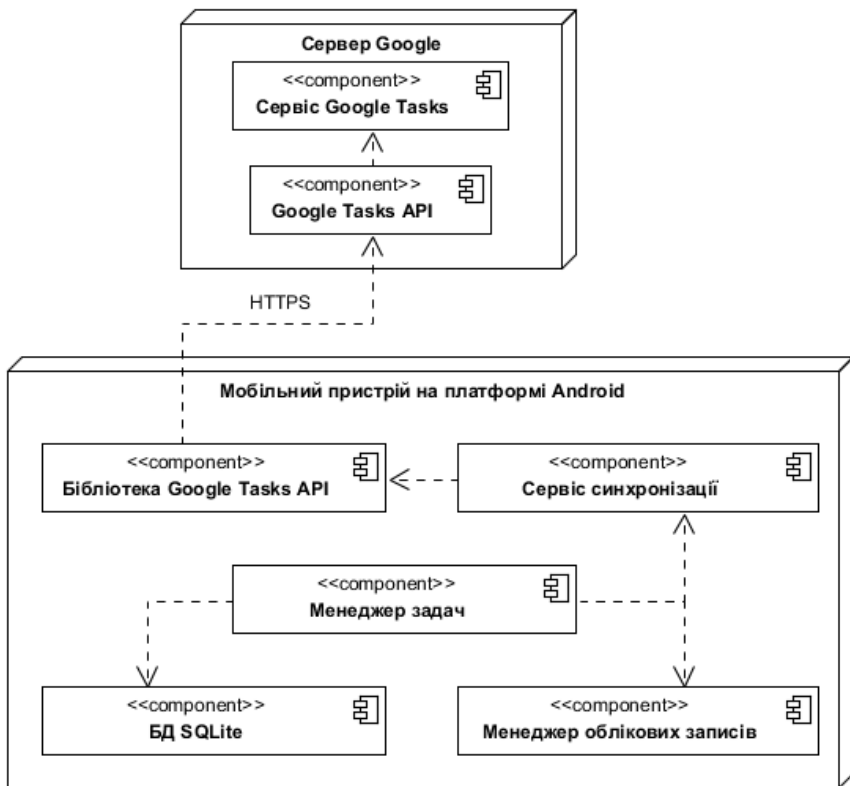


Рис. 1 – Загальна архітектура мобільної СКЗ із використанням MOC Android та сервісу Google Tasks

Оскільки Google Tasks є стороннім програмним сервісом, то будь-яка його модифікація з метою додання можливості адаптивного управління не є можливою. На стороні мобільного застосування можливо використати кеш (cache) запитів до локальної БД та засоби для адаптивного керування розміром кешу, що є актуальним у випадку великої кількості задач, створених користувачем мобільної СКЗ та в умовах обмеженості обчислювальних ресурсів мобільного пристрою.

3. Визначення та формальний опис ресурсів адаптації мобільної системи. Адаптація є важливою складовою процесу управління будь-якими складними об'єктами, що є насамперед необхідним в умовах постійних змін, що відбуваються в середовищі їх функціонування. Спираючись на загальновідому класифікацію підходів до побудови адаптивних систем управління [6–7], а також беручи до уваги деякі особливості саме адаптивних

програмних систем (див., наприклад, в [8]), в цьому дослідженні в подальшому буде розглянуто підхід до реалізації параметричної адаптації МІС при вирішенні задачі управління розміром кеш-пам'яті клієнтського пристрою мобільної СКЗ. В її архітектурі «вузьким місцем» є локальна БД, оскільки саме процес обробки SQL-запитів до неї, при великій кількості зберігаємих даних та при відносно низькій обчислювальній потужності мобільних пристроїв, може суттєво вплинути на продуктивність усього мобільного застосування.

Для вирішення цієї задачі пропонується побудова кешу запитів до БД, який буде зберігати записи, до яких найчастіше відбувалося звернення. Для забезпечення функціонування кешу обрано алгоритм LRU (Least Recently Used) [9]. Цей алгоритм передбачає обробку елементів з кешу за принципом побудови такої структури даних як стек (stack). Тобто, нові елементи додаються до вершини стеку, а коли наявний розмір стеку приймає граничне значення, то елементи даних, які більше не використовуються, витісняються із дна стеку.

На розмір кешу (C_s) впливають такі фактори, як: відсоток вільної оперативної пам'яті мобільного пристрою (M) від загального об'єму та кількість унікальних запитів до БД (R).

Для функціонування блоку адаптивного управління (БАУ) можливо використати такі підходи як: нейронні мережі, дерева рішень, вивід на основі прецедентів та деякі інші. Спираючись на раніше вказані особливості мобільних пристроїв, для розробки БАУ найбільш доцільно використовувати підхід із використанням виводу на основі аналізу прецедентів (Case Based Reasoning – CBR) [10]. Він не передбачає наявності потужних обчислювальних ресурсів та дозволяє зберігати необхідні дані для алгоритму управління безпосередньо в оперативній пам'яті мобільного пристрою. Використання CBR призводить до зменшення часу на розрахунок розміру кешу в залежності від поточного стану системи.

CBR – це метод прийняття рішень, в якому використовуються знання про попередні проблемні ситуації або випадки (прецеденти) [7, 10]. При розгляді нової проблеми (поточного випадку) знаходиться схожий прецедент як її аналог. Можна спробувати використовувати його рішення, можливо, адаптувавши щодо поточного випадку, замість того, щоб шукати вирішення кожен раз спочатку. Після того, як поточний випадок буде оброблений, він вноситься до бази прецедентів, разом із своїм рішенням, для можливого подальшого використання.

Прецедент – це опис проблемної ситуації із зазначення дій, що застосовуються в цій ситуації. Опис кожного прецеденту включає:

- опис проблемної ситуації;
- визначення рішення цієї проблеми.

Формально прецедент можна представити таким чином:

$$\vec{p} = \left\{ \vec{d}, \vec{s} \right\} \quad (1)$$

де $\vec{d} = \{d_1, d_2, \dots, d_m\}$ – опис проблемної ситуації;

$\vec{s} = \{s_1, s_2, \dots, s_k\}$ – опис рішення.

Стосовно до задачі, яка розглядається в даній статті, формальний опис прецеденту буде виглядати так:

$$\vec{d} = \{R, M\}, \quad (2)$$

$$\vec{s} = \{C_s\} \quad (3)$$

де R – кількість унікальних запитів до бази даних;

M – відсоток вільної пам'яті;

C_s – розмір кешу (кількість елементів у стеку).

Для адаптивного керування кешем пропонується алгоритм вказаний на рисунку 3. Цей алгоритм враховує необхідність нормування параметрів.

Як було зазначено вище, один із параметрів, що використовується – відсоток вільної оперативної пам'яті мобільного пристрою. Для запобігання частого перерахування об'єму кешу значення цього параметру округлюється до цілого числа.

Розрахунок розміру кешу відбувається за наступною формулою:

$$C_s = \frac{R * r_{avg}}{M} \quad (4)$$

де C_s – розмір кешу;

R – кількість унікальних запитів;

r_{avg} – середній розмір запису у кеші (байти);

M – коефіцієнт вільної оперативної пам'яті (%).

Геометрична інтерпретація цього формалізованого опису ресурсів адаптації мобільної СКЗ наведено на рис. 2.

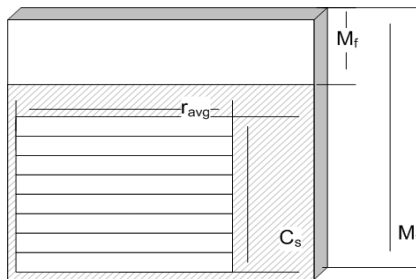


Рис. 2 – Формалізована модель ресурсів адаптації СКЗ

Беручи до уваги загальновідому процедуру CBR [7], а також враховуючи формулу (4), можливо запропонувати загальний алгоритм адаптивного управління розміром кешу, який у спрощеному вигляді наведено на рис. 3.

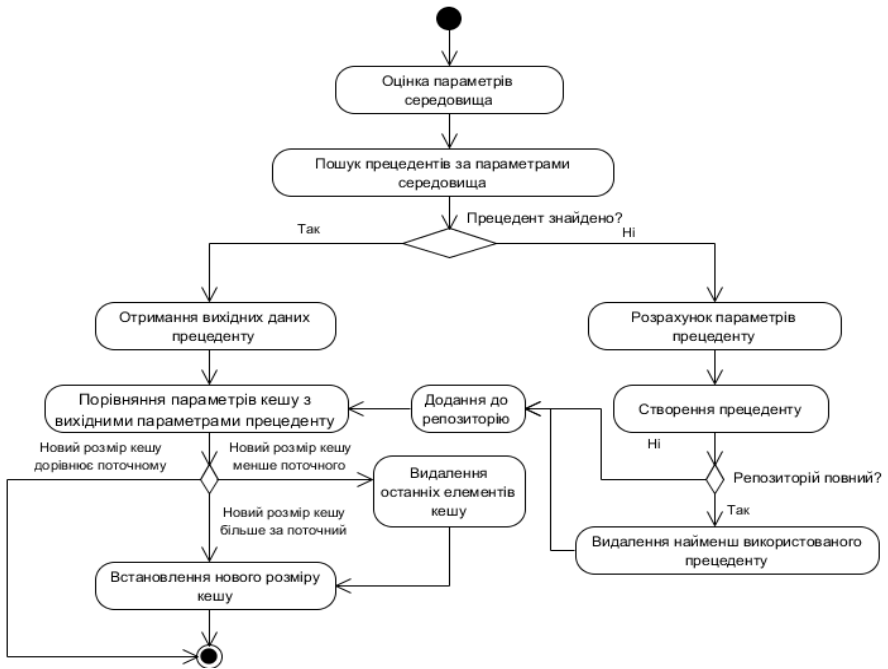


Рис. 3 – Алгоритм адаптивного управління кешем

При цьому коефіцієнт вільної оперативної пам'яті розраховується за наступною формулою:

$$M = \frac{M_f}{M_t} \cdot 100\% \quad (5)$$

де M_f – об'єм вільної оперативної пам'яті (Мб);
 M_t – загальний об'єм оперативної пам'яті (Мб).

Такий підхід дозволяє змінювати розмір кеш-пам'яті у мобільній СКЗ в залежності від стану середовища.

4. Архітектурна модель СКЗ з вбудованим БАУ. Наведена у розділі 2 архітектурна модель мобільної СКЗ була розширена компонентами, що є необхідними для реалізації адаптивного управління кеш-пам'яттю мобільного пристрою. Відповідна діаграма розміщення компонентів наведена на рис. 4, а саме:

1. «Блок адаптивного управління», що реалізує моніторинг характеристик мобільного пристрою (відсотку вільної оперативної пам'яті та кількості запитів до БД), розрахунок та зміну розміру кешу запитів до БД.
2. «Кеш запитів», що реалізує збереження в оперативній пам'яті мобільного пристрою запитів до БД та результатів їх виконання.

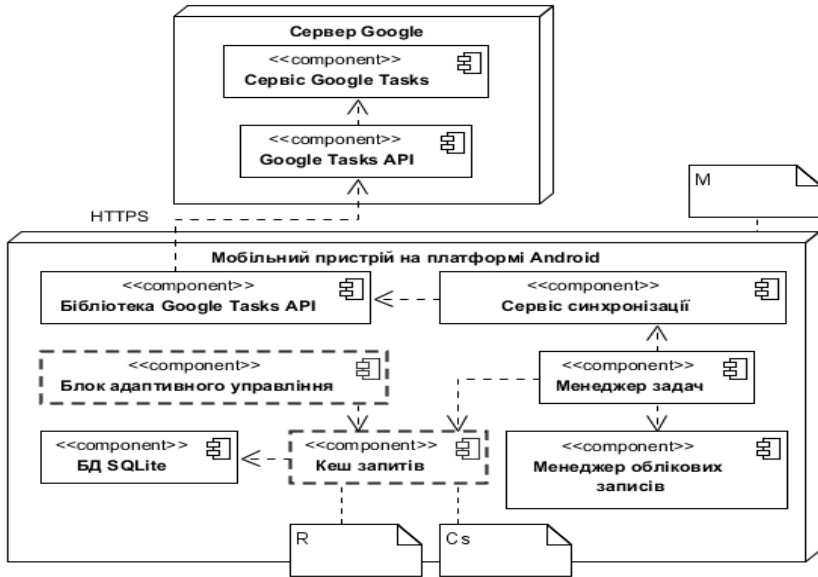


Рис. 4 – Загальна архітектура мобільної СКЗ з компонентами адаптивного управління

У якості цільової платформи для розробки прототипу такої системи обрано МОС Google Android, у якості сервісу для збереження задач обрано зовнішній сервіс Google Tasks.

На основі архітектури, що наведена на рис. 4, було програмно реалізовано прототип мобільної СКЗ та модель кешу запитів. Цей прототип пройшов випробування на декількох тестових наборах параметрів розміру кешу та кількості записів до локальної БД (деякі результати цих тестів наведені на рис. 5). Вони показали, що за рахунок збільшення розміру кешу підвищується швидкість виконання запитів, і це, в цілому, прискорює відображення даних на екрані мобільного пристрою. Але для певного розміру кешу існують деякі «порогові значення» мінімального та максимального його розміру, що пов'язано із загальною кількістю записів у БД. При досягненні цих значень швидкість виконання запитів може значно зменшуватися, тому постає питання визначення алгоритму пошуку ефективного значення розміру кешу.

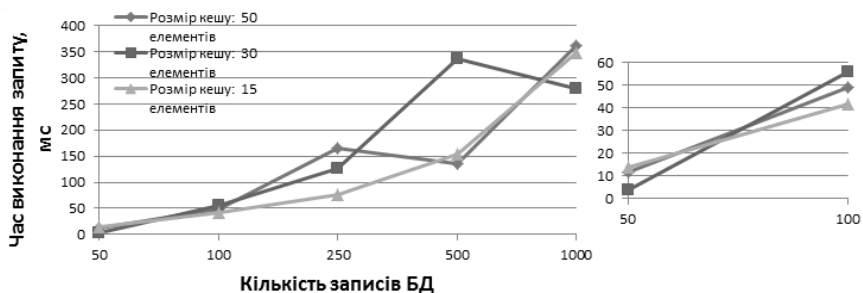


Рис. 5 – Результати тестування прототипу програмного рішення

Проведені дослідження свідчать про те, що розроблений підхід є працездатним, але деякі значення часу виконання запиту свідчать про необхідність більш детального тестування.

5. Висновки та напрямки подальших досліджень. В даній науковій статті: 1) показано актуальність розробки адаптивного програмного забезпечення для мобільних інформаційних систем; 2) проаналізована типова архітектура існуючих мобільних систем та визначені її можливі ресурси адаптації; 3) розроблено підхід до адаптивного управління розміром кеш-пам'яті мобільного пристрою, що базується на використанні методу логічного виводу на основі аналізу прецедентів; 4) запропоновано перспективну архітектуру клієнтського застосування у мобільній системі керування завданнями (СКЗ) з вбудованим блоком адаптивного управління (БАУ).

Отримані результати будуть використані при розробці прототипу мобільної СКЗ з інтегрованим БАУ на платформі Google Android та для проведення серії експериментів з оцінки ефективності запропонованого підходу. Також алгоритм керування розміром кешу буде модифіковано з метою виявлення «порогових значень» розміру кешу.

Список літератури: 1. Офіційна сторінка мобільного проекту Due Today // <http://bit.ly/s9ZLJ2> // 27.10.2010. 2. Офіційна сторінка мобільного проекту Shuffle // <http://bit.ly/iXcadm> // 27.10.2010. 3. *Vietanh N.* Mobile Computing & Disconnected Operation: A Survey of Recent Advance // <http://bit.ly/sTqXyo> // 27.10.2010. 4. Офіційний ресурс розробника методології GTD // <http://www.davidco.com> // 01.11.2011. 5. *Komatineni S.* Pro Android 3 / *S. Komatineni, D. MacLean, S. Y. Hashimi* – NY.: Apress, 2011. – 1200 с. 6. *Расстригин Л. А.* Адаптация сложных систем / *Л. А. Расстригин* – Рига: «Зинатне», 1981. – 375 с. 7. *Карпов Л. Е., Юдин В. Н.* Адаптивное управление по прецедентам, основанное на классификации состояний управляемых объектов // <http://bit.ly/vA7GF1> // 17.03.2011. 8. *Ткачук Н.В.* Структурная адаптация программных систем: анализ состояния проблемы и некоторые подходы к ее решению / *Н. В. Ткачук, К. А. Нагорный* // Східно-Європейський журнал передових технологій – 2009. – № 6 (42). – С. 33–36. 9. *Jelenkovi'c P.* Least-Recently-Used Caching with Dependent Requests / *Jelenkovi'c P., Radovanovi'c A.* // Theoretical Computer Science, №326. – 2004. – С. 293–327. 10. *Павлов А. И., Юрин А. Ю.* Программный модуль правдоподобного вывода по прецедентам // <http://bit.ly/uPgQ36> // 07.10.2010.

Надійшла до редколегії 07.12.2011