УДК 004.3

**Я.М. КЛЯТЧЕНКО, В.П. ТАРАСЕНКО, О.К. ТЕСЛЕНКО**

***Національний технічний університет України «КПІ», Україна***

# АНАЛІЗ ПАРАЛЕЛІЗМУ В АЛГОРИТМАХ ІЄРАРХІЧНОГО АДАПТИВНОГО ПОРІВНЯННЯ ІНФОРМАЦІЙНИХ ОБ'ЄКТІВ

*Узагальнюється метод нечіткого порівняння інформаційних об'єктів – метод адаптивного порівняння. Пропонується підхід до реалізації інформаційної технології ієрархічного адаптивного порівняння інформаційних об'єктів, що орієнтований на використання багатоядерних процесорів (багатопроцесорних систем). У цьому випадку переваги ієрархічного адаптивного порівняння проявляються найбільше. При цьому можна досягти суттєво кращих показників по завантаженню процесорів та по швидкодії порівняно з використанням доступних однопроцесорних систем.*

***Ключові слова:*** *адаптивне порівняння, багатопроцесорна система*

## Вступ

В забезпеченні інформаційної стійкості комп'ютерних систем та мереж існує малодосліджений напрямок, який полягає в протидії негативним явищам, які самі по собі не впливають на надійність надання сервісів (за винятком використання ресурсів), але суперечать існуючим в суспільстві моральними та законодавчими нормами.

До таких явищ можна віднести використання різних «троянських» технологій, несанкціоноване запозичення даних з комп'ютера та використання обчислювальних потужностей без узгодження із власником, використання комп'ютерних мереж для розповсюдження інформації, яка знаходиться поза законом або моральними нормами суспільства, використання комп'ютерних технологій з метою плагіату та порушення авторських прав і т.п. Серед цих явищ в освітніх інформаційних технологіях особливу стурбованість викликає використання запозичень. Це обумовлює необхідність створення технічних засобів для виявлення запозичень в інформаційних об'єктах. (Зауважимо, що рішення про плагіат та порушення авторських прав виконується виключно уповноваженими на це спеціалістами на основі існуючої нормативної бази).

## 1. Постановка задачі дослідження

Широке впровадженням багатоядерних процесорів та можливість реалізації багатопроцесорних систем в межах одного кристалу сучасних ПЛІС актуалізує задачу ґрунтовного дослідження методики розпаралелювання для автоматизованого визначення запозичень в інформаційних об'єктах.

## 2. Узагальнення методики адаптивного порівняння інформаційних об'єктів

В роботах [1 – 4] запропонований метод адаптивного порівняння інформаційних об'єктів, який за його основною концепцією можна віднести до методів нечіткого пошуку символьних послідовностей, не дивлячись на відмінності в постановці конкретних задач, що вирішуються його застосуванням .

Суть методики адаптивного порівняння полягає в наступному. Порівнюються дві послідовності символів - B - послідовність-зразок довжиною в $n_b$ символів (байт) та A - послідовність символів, яка перевіряється, довжиною $n_a$, (піддослідна послідовність). Встановлюється значення d – мінімальне значення довжини збігу двох будь-яких підпослідовностей із B та A, яка береться до уваги. Далі довжину збігу двох будь-яких підпослідовностей будемо називати їх спів- падінням. Значення d визначають, виходячи із типу та особливостей інформаційного об'єкту так, щоб забезпечити ігнорування статистично обумовлених співпадінь (наприклад, співпадіння окремих слів в текстах). Звідси виникає задача визначення всіх символів послідовності A, які входять в співпадаючі підпослідовності із B, довжиною не менше d. Відносна кількість - Loan(A,B) таких символів в послідовності A (в подальшому їх будемо називати такими, що позначені) вказує на рівень запозичень в A із B. Відносна кількість Orig(A,B) символів, що не є позначеними, в послідовності A вказує на рівень оригінальності A порівняно з B. Очевидно, що

$$Loan(A,B) + Orig(A,B) = 1.$$

В роботах [2 – 4] запропоновано ряд варіантів алгоритму швидкісного та детального адаптивного

порівняння інформаційних об'єктів. Якщо в послідовності A вибирати "робочі" підпослідовності довжиною c < d символів з кроком x, та порівнювати їх із всіма підпослідовностями довжиною c із B з кроком y, то при $x \le d - c + 1$ та y=1 в A будуть виявлені всі співпадаючі підпослідовності довжиною не меншою за d. Величина c визначається можливістю одночасного порівняння символів підпослідовностей. Для сучасних процесорів характерні команди порівняння даних довжиною від 1 до 8 байт, тому $1 \le c \le 8$ відповідно.

При швидкісному порівнянні використовується операція, подібна до сканування, тобто пошук чергової "робочої" підпослідовності із A в послідовності B. У разі успіху пошуку позначаються всі символи послідовності A в оточенні вибраної "робочої" підпослідовності. Одержана при цьому оцінка оригінальності $Orig_s(A,B) \le Orig(A,B)$, тобто може розглядатись як нижня границя значення оригінальності A порівняно із B. В сучасних процесорах реалізовані апаратні цикли для пошуку елемента в масиві (наприклад, команди ***Repne Scan***). Але вони не забезпечують виконання y=1 за умови, що кількість байт на символ не дорівнює значенню c. В [4] була запропонована команда ***Simple_Scan,*** яка це забезпечує при реалізації процесорного пристрою на базі ПЛІС. Для зменшення часу швидкісного порівняння в [4] також запропонована команда ***Double_Scan***, яка виконує сканування пари сусідніх "робочих" підпослідовностей із A в послідовності B з кроком x. Команда ***Double_Scan*** може також використовуватись для зменшення хибних позначень символів із A. В разі успішного знаходження співпадіння командою ***Simple_Scan*** команда ***Double_Scan*** визначає співпадіння пар допоміжних "робочих" підпослідовностей із ближнього оточення. Позначення символів "робочої" підпослідовності та символів оточення в послідовності A виконується лише при знаходженні співпадіння хоча б однієї із пар.

В [4] була запропонована команда ***Fast_Scan***, яка визначає входження чергової "робочої" підпослідовності із A в підпослідовність довжиною c* символів послідовності B, де $d \ge c^* > c$. Реалізація такої команди може бути виконана на матричному комбінаційному пристрої, що забезпечує максимальну швидкість. При використанні такого спеціалізованого пристрою значення кроку

$$y \le c^* - c + 1.$$

Якщо витримувати пропорційність значення d довжині $n_a$, то час обчислення $Orig_s(A,B)$ фактично не буде залежати від довжини A. Якщо забезпечувати пропорційність значення y довжині $n_b$, то час для обчислення $Orig_s(A,B)$ фактично не буде залежати від довжини B.

Якщо за певних умов нижня оцінка оригінальності, що одержана швидкісним алгоритмом не задовольняє, то використовується детальне адаптивне порівняння. Загальний алгоритм полягає в порівнянні підпослідовностей, починаючи із кожного символу послідовності A з підпослідовностями із B, також починаючи із кожного її символу. Якщо довжина знайденої спільної підпослідовності більше або дорівнює d , то відповідні символи послідовності A позначаються. Кількість позначених символів використовується для обчислення точної оцінки запозичень Loan(A,B). Час виконання такого алгоритму пропорційна $n_a$ в випадку повного співпадіння послідовностей A та B (Loan(A,B)=1), та пропорційна $n_a \times n_b$ у випадку відсутності спільних підпослідовностей довжиною, яка більша або дорівнює d (Orig(A,B)=1). Якщо в алгоритмі детального порівняння враховувати результати швидкісного порівняння, то витрати часу на детальне порівняння будуть пропорційні величині, не більшій за 2dk , де k –кількість позначених "робочих" підпослідовностей із A. Таким чином, у випадку Orig(A,B)=1 і, відповідно, k=0 виконання детального порівняння з найбільшими затратами часу, пропорційними $n_a \times n_b$ не потрібне. В [5] була запропонована апаратна реалізація алгоритму детального порівняння на ПЛІС з використанням регістра зсуву. В цьому випадку витрати часу на детальне порівняння пропорційні $n_a + n_b$, без використання результатів попереднього швидкісного порівняння.

## 3. Аналіз розпаралелювання алгоритмів адаптивного порівняння

Використання методики позначення символів піддослідної послідовності допускає просте розпаралелювання при порівнянні піддослідної послідовності з базою оригінальних послідовностей. При цьому кожний потік (процесор) реалізує алгоритм адаптивного порівняння однієї і тієї підослідної послідовності з окремою оригінальною послідовністю із бази. В спільній пам'яті багатопроцесорної системи може знаходитись лише масив для позначення символів піддослідної послідовності. При такій організації обчислень визначається оригінальність $Orig(A,B_1,\ldots, B_m)$ та може виявлятися факт компіляції піддослідної послідовності із декількох оригінальних.

Складнішою є організація розпаралелювання безпосередньо в адаптивному порівнянні. При використанні багатопроцесорних систем практичного значення набуває інформаційна технологія ієрархічного адаптивного порівняння. Така технологія ґрунтується на тому, що більшість інформаційних

об'єктів мають ієрархічну структуру, де елементи верхнього рівня ієрархії складаються із елементів нижнього рівня. Будемо вважати, що інформаційні об'єкти на будь-якому рівні ієрархії складаються з послідовності символів з деякого скінченого алфавіту. В той же час будь-який інформаційний об'єкт являє собою послідовність бітів (байтів), що зберігається для символів будь-якого рівня ієрархії. Це об'єкт забезпечує простоту розбиття символів верхнього рівня ієрархії на поточному рівні.

Особливо показовим прикладом ієрархічної структурованості є текстові дані. Для текстових даних в загальному випадку можна вказати на наступні рівні ієрархії – біти (алфавіт складається із 0 та 1), коди (алфавіт складається із букв, цифр і т.д. та для подання символів алфавіту використовуються байти або слова), речення (алфавіт складається із всіх можливих речень, тобто символами є речення), абзаци (символ алфавіту – абзац), та інші текстові структурні складові - підпункти, пункти, підрозділи, розділи, книги, сховища даних. Для подання символів верхніх рівнів ієрархії використовуються масиви, файли, підкаталоги, каталоги і т.п.

Розглянемо модифікацію адаптивного порівняння у відповідності до вимог технології ієрархічного порівняння. Позначимо через і поточний рівень ієрархії, починаючи з верхнього рівня (i=1,2,…,m), де m – кількість рівнів ієрархії (m ≥ 2). Найнижчий m-й рівень ієрархії характеризується безпосереднім використанням апаратних можливостей, тобто можливостей системи команд універсального або спеціалізованого процесора, або ж відповідного спеціалізованого пристрою на ПЛІС. Відповідно встановлюється суть символу (m-1)-го рівня ієрархії. Наприклад, використання спеціалізованого пристрою на ПЛІС [5] або при реалізації в спеціалізованому процесорі команд **Fast_Scan** довжина символу (m-1)-го рівня ієрархії повинна відповідати реченню або абзацу в текстових даних.

Вхідними даними для адаптивного порівняння і-го рівня є два символи із (i-1)–го рівня (наприклад два розділи). Ці два символи алфавіту верхнього рівня є рядками символів із алфавіту даного рівня (наприклад – речення). Для порівняння символів алфавіту і-го рівня використовується адаптивне порівняння (i+1)-го рівня. Як видно із наведеного, задачею адаптивного порівняння в технології ієрархічного порівняння є визначення співпадіння символів верхнього рівня ієрархії. Результатом адаптивного порівняння в даному випадку є одне із двох можливих значень – співпадають чи не співпадають два задані символи верхнього рівня (1-0 або ***true-false***). У випадку швидкісного порівняння на і-му рівні ієрархії значення 0 (***false***) установлюється за умови, що $Orig_s(A,B) \geq O(i)$. В іншому випадку встановлюється значення 1 (***true***). При детальному порівнянні значення 1 (***true***) встановлюється, якщо $Loan(A,B) \geq L(i)$. Граничні значення відносного рівня оригінальності O(i) та відносного рівня запозичень L(i) встановлюються на кожному рівні ієрархії в залежності від характеру інформаційного об'єкту та існуючих нормативних документів, моральних засад або здорового глузду. Оптимальні значення O(i) та L(i) також можуть бути результатом спеціального дослідження, зміст якого визначається характером і особливостями інформаційного об'єкту. Критеріями оптимізації тут є мінімальні (або відсутні) похибки в виявлені запозичень, а саме: ідентифікація не існуючих запозичень або не виявлення існуючих.

Оскільки суть адаптивного порівняння не змінюється з переходом з рівня на рівень в ієрархії, то можна утворити абстракцію – клас адаптивного порівняння. До властивостей класу слід віднести значення d, O(i) та L(i). Дані класу – два символи верхнього рівня ієрархії. Метод класу (надалі – метод порівняння) – визначення співпадіння символів верхнього рівня ієрархії, який полягає в розбитті символів верхнього рівня ієрархії на символи поточного рівня, виконанні швидкісного порівняння та детального порівняння. В конкретних випадках - швидкісне (при d=1) або детальне (при Orig(A,B)=1) порівняння не виконується. Визначення x, c, c*, у інкапсулюється у внутрішніх процедурах класу.

При виконанні швидкісного або детального порівняння генеруються об'єкти класу адаптивного порівняння нижнього рівня ієрархії та забезпечується можливість одночасного (паралельного) виконання їх методів порівняння. Після цього виконання методу порівняння об'єкту поточного рівня призупиняється до одержання результатів із нижнього рівня ієрархії.

Використання розглянутого класу адаптивного порівняння забезпечує суттєве розпаралелювання процесу нечіткого порівняння інформаційних об'єктів. Очевидно, що порівняння будь-яких пар символів поточного рівня (один із піддослідного об'єкта, другий – із об'єкта зразка) може виконуватись одночасно (паралельно). При переході з рівня на рівень ієрархії кількість паралельних потоків може збільшуватися в геометричній прогресії, і тому, оптимізація завантаження доступного пула процесорів є окремою складною проблемою. Тут же доречно проаналізувати ефективність основних засад адаптивного порівняння з точки зору оптимального завантаження пула процесорів.

Очевидно, що на рівнях ієрархії, крім останнього (m-го), значення c (кількість символів поточного рівня ієрархії для одночасного порівняння) доречно прийняти за 1. Дійсно, ймовірність співпа-

діння підпослідовностей довжиною c >1 різко зменшується з ростом c, тобто неспівпадіння визначається на кількості символів, меншій за c. Тоді перевірка співпадіння решти символів не потрібна. Аналогічна проблема характерна і для детального порівняння в цілому, враховуючи послідовний характер посимвольного порівняння двох послідовностей. Можливим вирішенням є використання методики, запропонованої в [4,5], шляхом попарного паралельного порівняння окремих символів з використанням програмної реалізації відносного зсуву символів та виявлення і позначення всіх збігів довжиною, не меншою за d.

Більш ефективне використання пула процесорів забезпечує швидкісне порівняння. Дійсно, програмна реалізація алгоритму команди ***Fast_Scan*** при умові c=1 не пов'язана з даремним запуском потоків. З іншого боку, по аналогії з алгоритмами команд ***Simple_Scan*** та ***Double_Scan*** на програмному рівні легко реалізувати порівняння трьох пар символів для зменшення хибних позначень "робочих" підпослідовностей (в даному випадку – символів).

## 4. Ієрархічне адаптивне порівняння для неструктурованих інформаційних об'єктів

У випадку, коли в структурі інформаційного об'єкту відсутня ієрархічність, то можливе її штучне введення з метою подальшого розпаралелювання нечіткого порівняння. Для цього на першому рівні ієрархії інформаційний об'єкт розбивається на частини, не обов'язково однакові по довжині. Кожна з цих частин в свою чергу може бути розділена на частини і т.д. до одержання послідовностей, які можуть бути оброблені за допомогою спеціалізованого пристрою на ПЛІС. На відміну від природного розподілу на частини при штучному розподілі підпослідовності, що співпадають та які необхідно виявляти, можуть бути розбиті на частини з довжиною меншою за d. Тому при швидкісному порівнянні необхідно в сукупність "робочих" підпослідовностей обов'язково вводити підпослідовності, які розташовані на початку і в кінці послідовності. При використанні детального порівняння розбиття інформаційного об'єкту на ієрархічні частини необхідно виконувати з перекриттям з обох боків на довжину d-1 символів нижнього рівня ієрархії. Враховуючи, що позначення символів поточного рівня ієрархії фактично виконується за допомогою операції **or**, то значення Loan(A,B) і, відповідно, результат порівняння символів верхнього рівня ієрархії не зміниться при відповідному незначному редагуванню L(i).

## Висновки

Доповнення адаптивного порівняння максимально допустимим відносним значенням запозичення (детальне порівняння) та мінімально допустимим відносним значенням оригінальності (швидкісне порівняння) забезпечує його використання, як операції порівняння окремих символів, що дає додаткові перспективи в розвитку методів нечітких порівнянь, зокрема, в розвитку інформаційної технології ієрархічного адаптивного порівняння.

Ієрархічне адаптивне порівняння може мати переваги і у випадку використання однопроцесорних систем, наприклад, за рахунок створення класу адаптивного порівняння, що забезпечує зменшення затрат на проектування відповідних програмних комплексів.

Найбільше переваги ієрархічного адаптивного порівняння проявляються при використанні багатопроцесорних систем.

При цьому можна досягти суттєво кращих показників по завантаженню процесорів та по швидкодії порівняно з використанням доступних однопроцесорних систем. Теоретично, при достатньо великій кількості процесорів швидкість нечіткого порівняння, в основному, буде визначатись швидкістю порівняння фрагменту на нижньому рівні ієрархії, яка, в свою чергу, може не залежати від величини фрагменту.

## Література

*1. Автоматизація оцінки оригінальності інформації [Текст] / В.П. Тарасенко, А.Ю. Михайлюк, О.К. Тесленко, О.С.Осипов // Наукові записки українського науково-дослідного інституту зв'язку. – 2007.– № 1. – С. 95 – 100.*

*2. Тарасенко, В.П. Ефективність ПЛІС - реалізацій адаптивного порівняння послідовностей символів [Текст] / В.П. Тарасенко, О.К. Тесленко, Я.М. Клятченко // Науковий вісник Чернівецького університету. Серія „Комп'ютерні системи та компоненти": Зб. наук. праць. – № 446. – С. 23 – 29.*

*3. Тарасенко, В.П. Структури для ПЛІС реалізацій детального адаптивного порівняння послідовностей символів [Текст] / В.П. Тарасенко, О.К. Тесленко, Я.М. Клятченко // Міжвузівський збірник наукових праць "Наукові нотатки", Луцьк, – 2010. – № 27. – С. 306 – 314*

*4. Команди спеціалізованого процесора на ПЛІС для адаптивного порівняння інформаційних об'єктів [Текст] / В.П. Тарасенко, Я.М. Клятченко, О.К. Тесленко, А.Ю. Михайлюк. // Радіоелектронні і комп'ютерні системи. – 2010. – № 7. – С. 220 – 225.*

*5. Пат. № 61653 G06F 7/38. Пристрій для детального адаптивного порівняння символьних послідовностей [Текст] / В.П. Тарасенко, О.К. Тесленко, Я.М. Клятченко; Заявитель и патентообладатель НТУУ КПІ; заявл. 29.22.2010; опубл. 25.07.2011 бюл. №14.*

*Надійшла до редакції 16.02.2012*

**Рецензент:** д-р техн. наук, проф., проф. кафедри Є.Т. Володарський, Національний технічний університет України «Київський політехнічний інститут», Київ.

## АНАЛИЗ ПАРАЛЛЕЛИЗМА В АЛГОРИТМЕ ИЕРАРХИЧЕСКОГО АДАПТИВНОГО СРАВНЕНИЯ ИНФОРМАЦИОННЫХ ОБЪЕКТОВ

***Я.М. Клятченко, В.П. Тарасенко, А.К. Тесленко***

Обобщается метод нечеткого сравнения информационных объектов – метод адаптивного сравнения. Предлагается подход к реализации информационной технологии иерархического адаптивного сравнения информационных объектов, ориентированный на использование многоядерных процессоров (многопроцессорных систем). В этом случае преимущества иерархического адаптивного сравнения проявляются сильнее всего. При этом можно достичь существенно лучших показателей по загрузке процессоров и по быстродействию по сравнению с использованием доступных однопроцессорных систем.

**Ключевые слова:** адаптивное сравнение, многопроцессорная система.

## ANALYSIS OF PARALLEL PROCESSES FOR ALGORITHM OF HIERARCHICAL ADAPTIVE COMPARISON OF INFORMATION OBJECTS

***Y.M. Klyatchenko, V.P. Tarasenko, O.K. Teslenko***

The method of variable comparison of information objects is generalized as the method of adaptive comparison. An approach of information technology of adaptive hierarchical comparison of information objects is proposed, that is focused on the use of multi-core processors (multiprocessor systems). In this case, the benefits of adaptive hierarchical comparison are shown the most. It is possible to achieve significantly better performance in processors' workload and performance in comparison with available single processor systems.

**Key words:** adaptive comparison, multiprocessor system.

**Клятченко Ярослав Михайлович** – ст.викладач кафедри спеціалізованих комп'ютерних систем НТУУ «КПІ», e-mail: k_yaroslav@scs.ntu-kpi.kiev.ua.

**Тарасенко Володимир Петрович –** д-р техн. наук, професор, завідувач кафедри спеціалізованих комп'ютерних систем НТУУ «КПІ», e-mail: vtarasen@scs.ntu-kpi.kiev.ua.

**Тесленко Олександр Кирилович** – канд. техн. наук, доцент кафедри спеціалізованих комп'ютерних систем НТУУ «КПІ», e-mail: teslenko@scs.ntu-kpi.kiev.ua.