УДК 003.26.09; 004.032.24-004.272.3

А.М. Луцків канд.техн.наук, доц., І.В. Вербицький
Тернопільський національний технічний університет імені Івана Пулюя, Україна

СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ПРИ РОЗВ'ЯЗАННІ ЗАДАЧ КРИПТОАНАЛІЗУ

A.M. Lutskiv Ph.D., Assoc. Prof., I.V. Verbytskyi

HIGH-PERFORMANCE COMPUTING SYSTEMS SOFTWARE DEVELOPMENT FOR THE CRYPTANALYSIS TASKS

Для розв'язання задачі криптоаналізу необхідною є наявність математичного та алгоритмічного забезпечення, яке дає змогу показати ненадійність тієї чи іншої криптосистеми й неможливість забезпечити нею належного рівня конфіденційності та/або цілісності даних. Проектування криптоалгоритмів, як правило, передбачає врахування ключових особливостей сучасних обчислювальних систем до векторизації та розпаралелення. Здійснюється декомпозиція криптоаналітичної обчислювальної задачі, яка поділяється на частини, які потім паралельно опрацьовуються.

Водночас, створення високоефективних криптоаналітичних обчислювальних систем передбачає обґрунтований вибір як програмних так і апаратних засобів з урахуванням цілої низки факторів. Здебільшого, вибір апаратного забезпечення зумовлений факторами доступності компонентів та їх універсальністю, з точки зору застосування, для різних криптографічних задач. Вибір апаратного забезпечення на основі якісних [1] та кількісних [2] показників дає змогу отримати прогнозований результат виконання обчислювальної задачі. На основі [1] випливає, що за критерієм швидкодії та ціни, найефективнішим підходом буде використання кластерних систем, які обладнані відеокартами.

Водночас ефективність роботи обчислювальної системи в значній мірі залежить від програмного забезпечення: операційної системи, утиліт та програм, що реалізують криптоаналітичні задачі. Від коректності реалізації криптоалгоритмів та урахування факторів, що дають змогу підвищити ефективність їх виконання на обчислювальних системах залежить час виконання обчислювальної задачі.

Перспективним підходом для реалізації криптоаналітичного програмного забезпечення, на думку авторів, є використання мов програмування високого рівня, які забезпечують кросплатформовість. У даному аспекті цілу низку переваг надає технологія Java: надійність, апробованість, безкоштовність, документованість, велика кількість готових напрацювань у вигляді бібліотек. З виходом 8-ї версії Java, підтримка багатопоточності була вдосконалена й додано цілу низку конструкцій мови, які дають змогу спростити процес створення багатопотокового програмного забезпечення [3], а також відлагоджувати його.

Наведемо апаратні компоненти та способи їх ефективного використання при створенні криптоаналітичного програмного забезпечення на мові Java:

Багатоядерні/багатопроцесорні системи зі спільною пам'яттю (SMP-системи) можуть бути ефективно задіяні шляхом використання типових багатопотокових Java-конструкцій. Зокрема, доцільним є використання Concurrent-колекцій, а не синхронізованих колекцій, а також Executor-фреймворку, а не типових Thread-конструкцій.

Водночас використання можливостей GPU-обчислювачів є можливим завдяки використанню спеціалізованих бібліотек JCUDA [4] та JOCL [5]. Це програмні

інтерфейси-оболонки (wrappers) над бібліотеками CUDA та OpenCL, відповідно. Бібліотеками підтримуються 32- та 64-бітні платформи Linux і Windows. JCUDA-бібліотеки мають реалізації усіх програмних інтерфейсів на мові Java. Написання GPU-залежного коду при використанні OpenCL здійснюється типовим способом: створюються CL-ядра, які в процесі виконання будуть скомпільовані JIT-компілятором для цільової платформи.

В аспекті створення програм для систем із розподіленою пам'яттю технологія Java підтримує цілу низку технологій для мережевої взаємодії, зокрема RMI та мережевих сокетів. Проте вони є менш ефективні при створенні паралельних програм ніж інтерфейс обміну повідомленнями - MPI. Варто відзначити, що з ростом популярності технології Java стали наявні й програмні інтерфейси для взаємодії з MPI. Зокрема, наявна реалізація стандарту MPI — пакет OpenMPI, шляхом надання API-інтерфейсу забезпечує підтримку технології Java на рівні MPI-3.1[6]. Очевидно, що в порівнянні з виконанням програмного забезпечення, яке створене на мовах C та FORTRAN є деякі відмінності, проте час виконання програм є цілком прийнятним. Таким чином, кластерні системи також можуть бути ефективно задіяні.

Отже, запропонований підхід дасть змогу:

1) спростити процес створення програмного забезпечення;

2) програмне забезпечення стане гнучкішим, за рахунок кросплатформовості;

3) будуть задіяні обчислювальні можливості графічних та центральних процесорів, а також буде забезпечена мережева взаємодія.

До недоліків запропонованого підходу можна віднести той факт, що значна частина бібліотек використовує JNI-функції, що в ряді випадків може супроводжуватись непрогнозованими результатами. Проте, значна частина із наведених бібліотек є достатньо апробовані й цілком можуть бути використані для розв'язання задач криптоаналізу.

Література

1. Загородна Н. В., Лупенко С. А. Луцків А. М. Обґрунтування вибору доступних програмно-апаратних засобів високопродуктивних обчислювальних систем для задач криптоаналізу. // Електроніка та системи управління. 2011. №1(27). - К.: НАУ, 2011. - с.42-50.

2. David A. Patterson and John L. Hennessy. 2011. Computer Architecture, Fifth Edition: A Quantitative Approach 5th . Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

3. Sosnoski D. Java 8 concurrency basics /Dennis Sosnoski // [Електронний ресурс] Режим доступу: URL: http://www.ibm.com/developerworks/library/j-jvmc2/index.html

4. Java bindings for CUDA [Електронний ресурс] Режим доступу: URL: http://jcuda.org

5. Java bindings for OpenCL [Електронний ресурс] Режим доступу: URL: http://jocl.org

6. Vega-Gisbert O. Design and implementation of Java bindings in Open MPI / O. Vega-Gisbert, J. E. Roman, and J. M. Squyres. // [Електронний ресурс] Режим доступу: URL: http://users.dsic.upv.es/~jroman/preprints/ompi-java.pdf