

РОЗПОДІЛЕНА РЕАЛІЗАЦІЯ МЕТОДА ЛІТТЛА З ВИКОРИСТАННЯМ WINDOWS COMMUNICATION FOUNDATION

Задача побудови у зв'язному графі $H = (V, U)$ найкоротшого замкненого маршруту, який з'єднує усі вершини і проходить у точності один раз по кожній вершині графу, називається задачею комівояжера (ЗК). Для отримання точних розв'язків задачі найефективнішим є метод гілок та меж (МГМ). Але для розв'язання задач великих розмірностей МГМ вимагає виконання великої кількості операцій та збереження великих обсягів даних, що викликає проблеми застосування МГМ у реальному масштабі часу. Тому актуальними є шляхи підвищення ефективності МГМ. Одним із таких шляхів є розробка та реалізація паралельних версій алгоритмів з подальшим їх виконанням на багатопроцесорних системах.

Застосування програмного фреймворку Windows Communication Foundation, який входить до складу .NET Framework, дає можливість організувати клієнт-серверну взаємодію та обмін даними між програмними продуктами, написаними на основі .NET Framework. Це, у свою чергу, дає можливість реалізувати клієнт-серверний додаток, який для розв'язання задач буде використовувати кілька процесорів та комп'ютери, об'єднані у мережу.

Для роботи клієнт-серверного додатку потрібна наявність одного керуючого процесора, на якому буде запущена головна (клієнтська) частина, і робочих процесорів, ресурси яких будуть використовуватися серверними процесорами.

Розподілене розв'язання задачі починається з того, що у клієнтській частині задається умова задачі, яку можна ввести вручну або згенерувати випадковим чином. Після цього клієнтський процес починає розв'язання задачі за допомогою МГМ, виконує деяке число його ітерацій. Це необхідно для формування початкової бази активних завдань, які пізніше будуть розподілені по робочих процесорах. Після формування початкової бази активних вершин, клієнтський процес здійснює трансляцію UDP-повідомлень зі службовою інформацією за допомогою класу *ServiceDiscoveryBehavior*. Вільні робочі процеси, які в даний момент часу простоюють, за допомогою класів *DiscoveryClient*, *FindCriteria*, та *FindResponse* отримують UDP-повідомлення та відгукуються на нього, надсилаючи у відповідь інформацію про апаратні характеристики обладнання (кількість процесорів, їх характеристики, кількість оперативної пам'яті і т.п.).

Головний процес, на основі отриманих даних, формує список робочих процесів для вирішення задачі. Початкова база активних вершин дерева розгалуження розподіляється між цими робочими процесорами пропорційно доступній оперативній пам'яті. Головний процес зберігає інформацію про поточний розв'язок-претендент на оптимум і глобальну верхню оцінку. На початку процесу розв'язання, глобальна оцінка дорівнює нескінченності, а розв'язок-претендент не визначений (*null*).

Кожний робочий процес, отримавши множину завдань, починає їх розв'язання. У випадку, якщо знайдено допустимий розв'язок, вага якого менша, ніж глобальна оцінка, робочий процес передає отриманий розв'язок головному процесу. Головний процес оновлює оцінку і повідомляє всім робочим процесам її нове значення. Кожен робочий процес, отримавши нове значення глобальної оцінки, переглядає множину своїх активних вершин і видаляє з черги усі вершини, оцінки яких більше або дорівнюють цій оцінці.

Якщо виявилось, що у робочого процесу множина завдань для вирішення вичерпана, він звертається до головного процесу для поповнення своєї бази завдань. Головний процес, отримавши такий запит, звертається до всіх робочих процесів і запитує у них певну кількість завдань для передачі процесу, який простоює.

Розв'язання завершується у випадку, коли всі робочі процеси вичерпують базу активних вершин. Глобальна оцінка у головного процесу являє собою вартість оптимального розв'язку, а розв'язок, збережений разом з цією оцінкою, є оптимальним розв'язком задачі.

Головний і кожний робочий процеси містять розв'язуючий модуль і модуль взаємодії. Розв'язуючий модуль представляє собою оболонку для класу *TSPSolver*, який дозволяє управ-

ляти розв'язанням на рівні одного процесу (виконувати вилучення і додавання активних вершин у дерево розгалуження, встановлювати глобальну оцінку і т.п.). Модуль взаємодії представляє собою клас, який здійснює обмін даними між головним і робочими процесами. За допомогою *Windows Communication Foundation* досягається уніфікація процесу обміну даними. Зокрема, встановлюється дуплексний зв'язок між головним і робочими процесами, що дає можливість головному процесу викликати методи об'єктів серверних процесів і навпаки.

Таким чином, робочі процеси можуть функціонувати як на одній багатопроекторній ЕОМ, так і на віддалених ЕОМ, об'єднаних локальною або глобальною мережею. При цьому звертання до локального або віддаленого процесу виконується за допомогою одного і того ж програмного коду.

Програмна реалізація складається з трьох частин. Перша частина – це проект *TSPClassLibrary*, у якому реалізовані усі класи, пов'язані з розв'язанням задачі. *TSPClassLibrary* скомпільований у вигляді DLL-бібліотеки, що дозволяє використовувати класи проекту при розробці інших програмних продуктів.

Друга частина – це серверний процес *TSPServer*. У ньому за допомогою UDP Discovery реалізована розсилка UDP-повідомлень з інформацією про серверний процес та його адресу у мережі. Реалізовано механізм прийому даних від клієнтського процесу та розв'язання отриманих підзадач, відправку повідомлення клієнтському процесу з інформацією про поточний розв'язок-претендент на оптимум.

Третя частина – клієнтський процес. У ньому реалізовано механізм керування розв'язанням задачі, розділення задачі на підзадачі та їх розподіл між серверними процесами.

Оскільки у *Windows Communication Foundation* по замовчуванню встановлені обмеження на передачу великих обсягів даних, було модифіковано файл *App.config* і реалізовано серіалізацію даних у потоки при передачі і десеріалізацію при прийомі.

Висновок

Реалізація розподіленого клієнт-серверного додатку дає змогу зменшити час отримання точних розв'язків ЗК та допомогою МГМ за умови, якщо швидкість передачі даних по мережі значно перевищує швидкість розв'язання підзадач на відповідних серверних процесах.

МОРОЗОВ Андрій Васильович, кандидат технічних наук, завідувач кафедри інформатики та комп'ютерного моделювання Житомирського державного технологічного університету. Наукові інтереси: задачі маршрутизації на графах та мережах, паралельні та розподілені системи, сучасні Інтернет-технології.

ЛЄТОШКО Анна Ігорівна, магістрант групи ПІ-37м кафедри програмного забезпечення обчислювальної техніки Житомирського державного технологічного університету. Наукові інтереси: задачі маршрутизації на графах та мережах, сучасні Інтернет-технології.