

Мета роботи: опанувати техніку розроблення паралельних програм у мультикомп'ютерному середовищі.

Теоретичні відомості

Багато обчислювальних задач у галузі фізики зводяться до операцій оброблення даних для кожної пари об'єктів наявної фізичної системи. Такою задачею є, зокрема, проблема, широко відома в літературі як гравітаційна задача N тіл (або просто задача N тіл) [5]. У найзагальнішому вигляді задача може бути описана в такий спосіб.

Нехай дана велика кількість тіл (планет, зірок тощо), для кожного з яких відома маса, початкове положення й швидкість. Під дією гравітації положення тіл змінюється, і необхідне розв'язання задачі полягає в моделюванні динаміки зміни системи N тіл протягом деякого заданого інтервалу часу. Для проведення такого моделювання заданий інтервал часу зазвичай розбивають на часові відрізки невеликої тривалості й в подальшому на кожному кроці моделювання обчислюють сили, що діють на кожне тіло, а потім оновлюють значення швидкості й положення для всіх тіл.

Очевидний алгоритм розв'язання задачі N тіл полягає в розгляді на кожному кроці моделювання всіх пар об'єктів фізичної системи й виконанні для кожної отримуваної пари всіх необхідних розрахунків. Як результат, за такого підходу час виконання однієї ітерації моделювання становитиме

$$T_1 = \tau \cdot N \cdot (N - 1) / 2,$$

де τ – час обчислення параметрів для однієї пари тіл.

Як впливає з наведеного опису, обчислювальна схема розглянутого алгоритму є порівняно простою, що дозволяє використовувати задачу N тіл як ще одну наочну демонстрацію застосування методики розроблення паралельних алгоритмів.

Слід зазначити, що для розв'язання задачі N тіл існують й ефективніші послідовні алгоритми, однак їх вивчення вимагає достатньо великих зусиль. З урахуванням цієї обставини для подальшого розгляду був вибраний саме цей "очевидний" (але не найшвидший) метод, хоча, у загальному випадку, безумовно, для розпаралелювання слід вибирати найкращі схеми виконання розрахунків.

Поділ обчислень на незалежні частини

Вибір способу поділу обчислень не викликає будь-яких утруднень; очевидний підхід – вважати базовою підзадачею весь обсяг обчислень, пов'язаних з обробленням даних для одного довільного тіла фізичної системи.

Виділення інформаційних залежностей

Виконання обчислень, пов'язаних з кожною підзадачею, стає можливим тільки у випадку, коли в підзадачах є дані (положення й швидкості пересування) усіх тіл наявної фізичної системи. Як результат, перед початком кожної ітерації моделювання кожна підзадача повинна отримати всі необхідні відомості від усіх інших підзадач системи. Таку процедуру передачі даних, як відзначалося в лекції 3, називають операцією збирання даних (single-node gather). У розглянутому алгоритмі вказана операція повинна бути виконана для кожної підзадачі – такий варіант передачі даних зазвичай називають операцією узагальненого збирання даних (multi-node gather or all gather).

Визначення вимог до необхідних результатів інформаційного обміну не приводить до однозначного встановлення потрібного інформаційного обміну між

підзадачами – досягнення необхідних результатів може бути забезпечене за допомогою різних алгоритмів виконання операції узагальненого збирання даних.

Найпростіший спосіб виконання необхідного інформаційного обміну полягає в реалізації послідовності кроків, на кожному з яких усі наявні підзадачі розбивають на пари й обмін даними здійснюється між підзадачами щойно створених пар. За умови належної організації попарного поділу підзадач $(N - 1)$ -разове повторення описаних дій приведе до повної реалізації необхідної операції збирання даних.

Розглянутий вище метод організації інформаційного обміну є досить трудомістким – для збирання всіх необхідних даних потрібно виконати $N - 1$ ітерацій, на кожній з яких виконується одночасно $N/2$ операцій передачі даних. Для скорочення необхідної кількості ітерацій можна звернути увагу на той факт, що після виконання першого кроку операції збирання даних підзадачі будуть уже містити не тільки свої дані, але й дані підзадач, з якими вони утворювали пари. Як результат, на другій ітерації збирання даних можна буде утворювати пари підзадач для обміну даними відразу про два тіла фізичної системи – тим самим, після завершення другої ітерації кожна підзадача буде містити відомості про чотири тіла системи тощо. Як можна помітити, цей спосіб реалізації обмінів дозволяє завершити необхідну процедуру за $\log_2 N$ ітерацій. Слід зазначити, що при цьому обсяг даних, що пересилаються, у кожній операції обміну подвоюється від ітерації до ітерації – на першій ітерації між підзадачами пересилаються дані про одне тіло системи, на другій ітерації – про два тіла тощо.

Використання розглянутого способу реалізації операції узагальненого збирання даних природно приводить до визначення доцільної структури інформаційних зв'язків між підзадачами у вигляді N -вимірного гіперкуба.

Масштабування набору підзадач та їх розподіл між процесорами

Як правило, кількість тіл у фізичній системі N значно перевищує кількість процесорів p . Як результат, розглянуті раніше підзадачі слід укрупнити, об'єднавши в рамках однієї підзадачі обчислення для групи N/p тіл. Після проведення подібної агрегації кількість підзадач і кількість процесорів буде збігатися, і під час розподілу підзадач між процесорами залишиться лише забезпечити наявність прямих комунікаційних ліній між процесорами з підзадачами, між якими будуть відбуватися інформаційні обміни під час виконання операції збирання даних.

Контрольні запитання

1. Що слід розуміти під паралельною програмою?
2. Що розуміють в MPI під комунікатором?
3. Як можна організувати приймання повідомлень від конкретних процесів?
4. Як визначити час виконання MPI-програми?
5. У чому відмінність парних і колективних операцій передачі даних?
6. Яка функція MPI забезпечує передачу даних від одного процесу всім процесам?
7. Що розуміють під операцією редукції?
8. У яких ситуаціях слід застосовувати бар'єрну синхронізацію?
9. Які режими передачі даних підтримуються в MPI?
10. Як організує неблокуючий обмін даними в MPI?
11. Які колективні операції передачі даних передбачені в MPI?

Індивідуальні завдання

Розробити програмну реалізацію задачі N тіл в середовищі MPI.