
Технология фрагментированного программирования

Перепелкин В.А., м.н.с. ИВМиМГ СО РАН
06.07.2015 г.

Проблема

— На решение какой проблемы направлена технология фрагментированного программирования?

Параллельная реализация крупномасштабных численных моделей для суперкомпьютеров — сложная задача системного параллельного программирования

Суперкомпьютеры

Суперкомпьютеры — это наиболее производительные вычислители планеты

Производительность: операции в секунду, объем памяти, скорость сети

Виды суперкомпьютеров: кластеры и грид-системы

Являются ли cloud-системы суперкомпьютерами?

Суперкомпьютеры



Суперкомпьютер ССКЦ



Списки суперкомпьютеров

<http://www.top500.org> — список мощнейших суперкомпьютеров планеты

<http://top50.supercomputers.ru/> — список мощнейших суперкомпьютеров СНГ

Применение суперкомпьютеров

“640 КБайт хватит для любых нужд”

- Моделирование ядерных реакций
 - Предсказание погоды
 - Поиск нефти
 - Разработка новых видов материалов и техники
 - Поиск новых лекарств
 - Задачи криптографии
 - “Добыча” криптовалют
 - ...
-

Крупномасштабные численные модели

Численные модели научного моделирования, характеризующиеся:

- Большим объемом данных (терабайты и более)
 - Большим объемом вычислений (терафлопсы и более)
-

Единицы измерения производительности

10^0	flops	byte	byte/sec
10^3	Kflops	Kbyte	Kbyte/sec
10^6	Mflops	Mbyte	Mbyte/sec
10^9	Gflops	Gbyte	Gbyte/sec
10^{12}	Tflops	Tbyte	Tbyte/sec
10^{15}	Pflops	Pbyte	Pbyte/sec
10^{18}	Eflops	Ebyte	Ebyte/sec

Программирование на суперкомпьютерах

- Параллельное, т.к. все суперкомпьютеры — параллельные
 - Системно сложное, т.к. требуется:
 - представить алгоритм решения задачи в параллельной форме
 - обеспечить эффективную работу суперкомпьютера (равномерная и полная загрузка полезными вычислениями)
-

Проблема

— На решение какой проблемы направлена технология фрагментированного программирования?

Параллельная реализация крупномасштабных численных моделей для суперкомпьютеров — сложная задача системного параллельного программирования

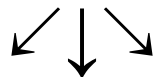
Цель технологии фрагментированного программирования (ТФП)

— В чем цель технологии фрагментированного программирования?

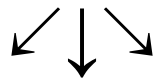
Автоматизация эффективной параллельной реализации численного алгоритма на суперкомпьютере

Автоматизация программирования

Формулировка задачи



Алгоритм решения



Параллельная программа

Формулировка задачи

- Функциональные требования, т.е. функция, которая должна быть вычислена
 - Пример: сортировка массива
 - Нефункциональные требования (эффективность)
-

Алгоритм решения задачи

- Алгоритм задаёт способ вычисления функции
 - Для вычисления одной и той же функции существует счётное множество алгоритмов
 - Разные алгоритмы обладают разными нефункциональными свойствами
-

Программа, реализующая алгоритм

Один и тот же алгоритм можно запрограммировать разными программами

Программа:

- реализует алгоритм
 - управляет ресурсами
 - содержит управление
 - менее переносима
-

Переносимость

- Переносимость на уровне исполняемых файлов
 - Переносимость на уровне исходных кодов
 - Переносимость с точки зрения сохранения нефункциональных свойств
-

Эффективность

- Экономия памяти
 - Экономия времени вычислений
 - Малая доля накладных расходов
-

Частичная автоматизация

Эффективная реализация численного алгоритма — алгоритмически труднорешаемая задача

Аналог: задача о рюкзаке

Вывод: автоматизация должна быть частичной, система — специализированной

Представление алгоритма

Разные представления алгоритма обладают разными свойствами:

- явность параллелизма
- гранулярность алгоритма
- автономность частей алгоритма
- управление ресурсами

Для разных целей удобны разные представления алгоритма

Масштабируемость исполнения алгоритма

Виды масштабируемости: сильная и слабая

Требования масштабируемости:

- Децентрализация по данным, вычислениям и коммуникациям
- Локальность коммуникаций

Отсутствует возможность иметь информацию сразу или всю

Цель технологии фрагментированного программирования (ТФП)

— В чем цель технологии фрагментированного программирования?

Автоматизация эффективной параллельной реализации численного алгоритма на суперкомпьютере

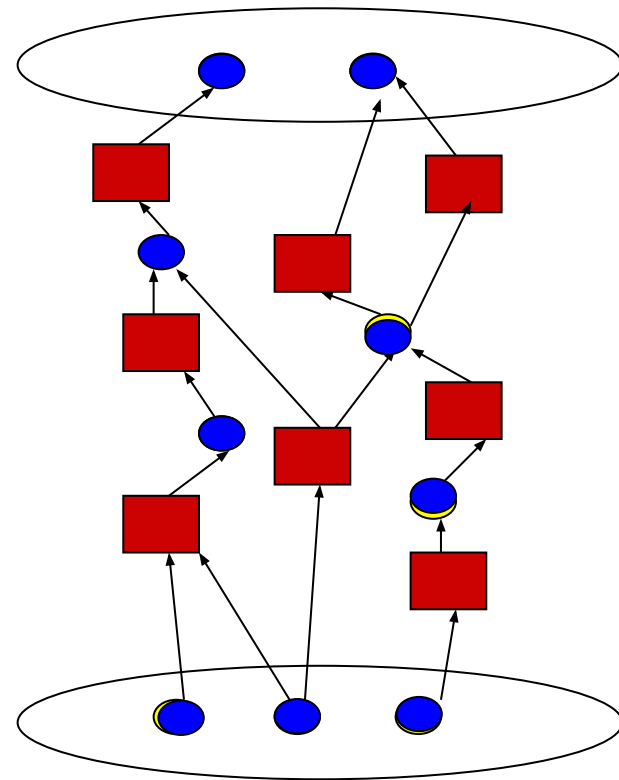
Фрагментированный алгоритм

— Каково представление алгоритма в ТФП и каковы его особенности?

Алгоритм представляется в явно-параллельной форме, ориентированной на автоматизацию обеспечения нефункциональных свойств

Фрагментированный алгоритм (ФА)

ФА — это набор
фрагментов данных (ФД),
фрагментов вычислений
(ФВ), и отношения in/out
Выходные ФД
вычисляются из входных
пока все ФВ не окажутся
исполненными

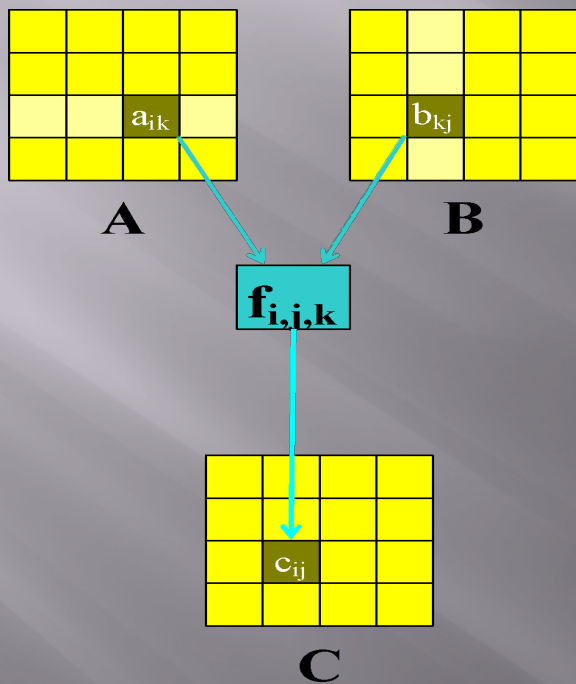


Особенности ФА

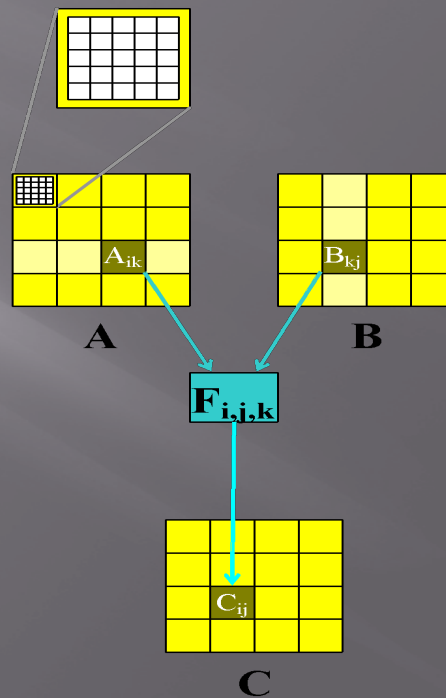
- Сериализуемые ограниченные по размеру ФД
 - Ограниченные по времени фрагменты вычислений без побочных эффектов
 - Единственность присваивания ФД
 - Крупная зернистость ФА
-

Пример ФА: умножение матриц

Исходный алгоритм



Фрагментированный алгоритм



Полезные свойства ФА

- Высокая переносимость
 - Ориентация на автоматизацию исполнения
 - миграция фрагментов
 - сохранение контрольных точек
 - различное управление и распределение ресурсов
 - контролируемая гранулярность
 - Масштабируемость
 - Явный параллелизм
-

Слабые стороны ФА

- Отсутствие привязки к конкретным ресурсам
 - Отсутствие императивного управления
 - Единственность присваивания ФД
-

Фрагментированный алгоритм

— Каково представление алгоритма в ТФП и каковы его особенности?

Алгоритм представляется в явно-параллельной форме, ориентированной на автоматизацию обеспечения нефункциональных свойств

Рекомендации

— Зачем нужны рекомендации при реализации ФА?

Рекомендации предназначены для оптимизации реализации ФА путем частичного решения труднорешаемых подзадач на высоком уровне

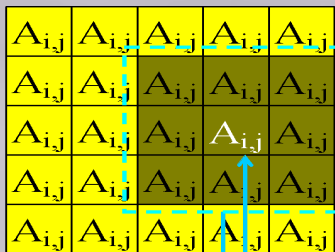
Рекомендации

Частичные решения по способу реализации
ФА — порядку выполнения ФВ,
переиспользования памяти, отображения
фрагментов на вычислительные узлы, и т.п.

Примеры рекомендаций

- Приоритет выполнения ФВ
 - Соседство ФД и ФВ
 - Группы ФВ, образующие “гамма-вычислений”
 - Совмещение буферов ФД
 - ...
-

Отношение соседства



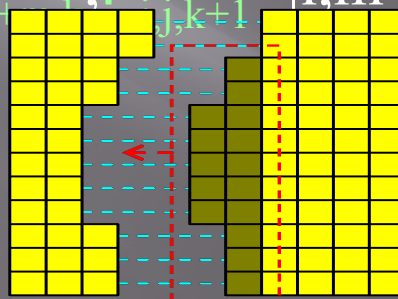
$A'_{i,j}$

$$DF = \{A_{i,j}\}$$

$$CF = \{A_{i,j} \rightarrow F_{i,j,k} \rightarrow A_{i,j}\},$$

$$\text{where } A_{i,j} = \{A_{i+l,j+m} | l,m \in [-1;1]\}$$

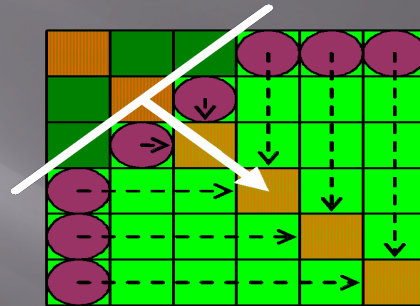
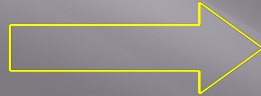
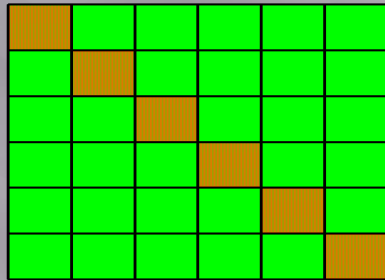
$$\rho = \{\langle F_{i+l,j+m}, F_{i,j,k+1} \rangle | l,m \in [-1;1]\}$$



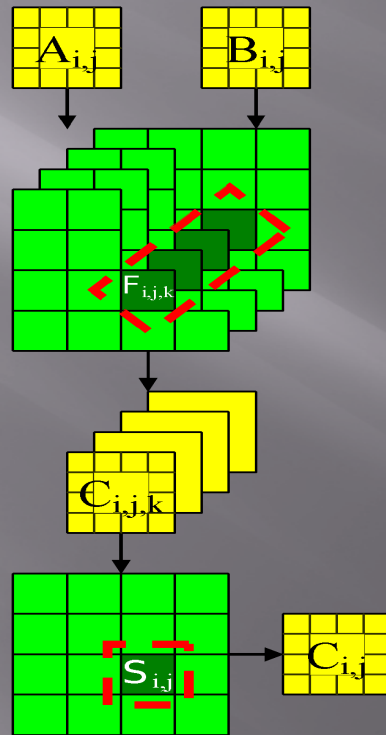
PE
0

PE
1

Приоритет ФВ

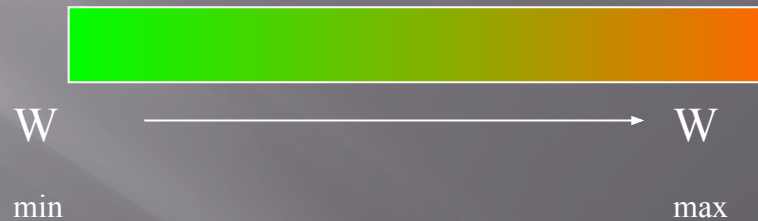


Группы ФВ



“Вес” ФВ

D_i	$U_{i,j}$	$U_{i,j}$	$U_{i,j}$
$L_{i,j}$	D_i	$U_{i,j}$	$U_{i,j}$
$L_{i,j}$	$L_{i,j}$	D_i	$U_{i,j}$
$L_{i,j}$	$L_{i,j}$	$L_{i,j}$	D_i



Рекомендации

ФА содержит все возможные способы его реализации (порядки выполнения ФВ и варианты отображения фрагментов на вычислительные узлы и т.п.)

Рекомендации ограничивают это множество реализаций до небольшого, содержащего преимущественно эффективные способы

Рекомендации — это высокоуровневое средство, их не требуется программировать, а лишь описать

Рекомендации

— Зачем нужны рекомендации при реализации ФА?

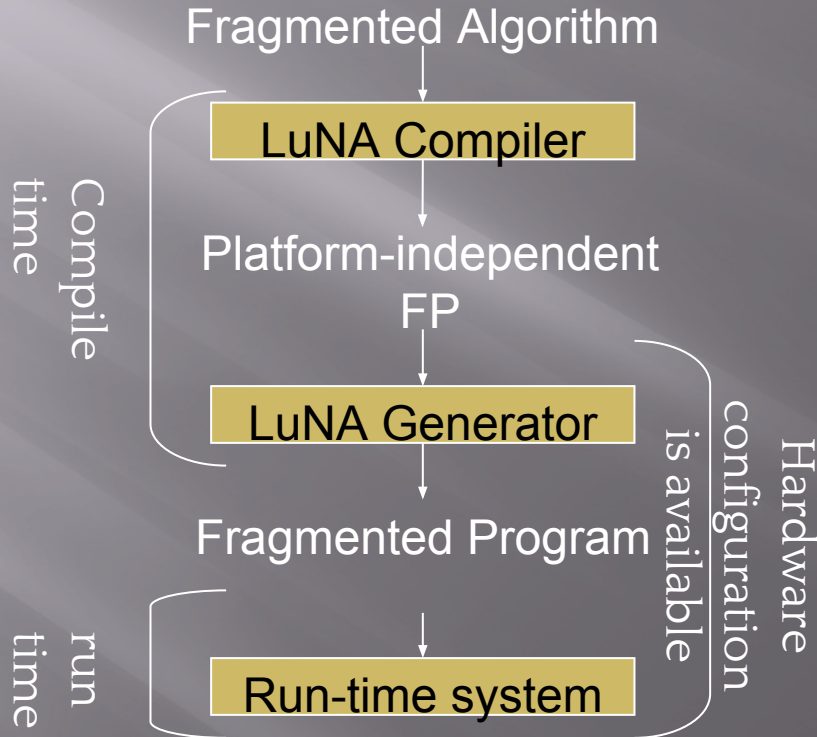
Рекомендации предназначены для оптимизации реализации ФА путем частичного решения труднорешаемых подзадач на высоком уровне

Система фрагментированного программирования LuNA

LuNA — Language for Numerical Algorithms

— Как устроена система LuNA и в чем
смысл такой организации?

Структура системы LuNA



Контрольные вопросы

- На решение какой проблемы направлена технология фрагментированного программирования?
 - В чем цель технологии фрагментированного программирования?
 - Каково представление алгоритма в ТФП и каковы его особенности?
 - Зачем нужны рекомендации при реализации ФА?
 - Как устроена система LuNA и в чем смысл такой организации?
-

**Спасибо за
внимание!**
