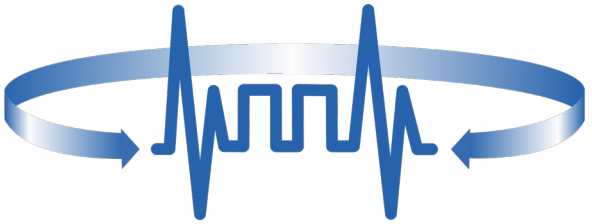
3-я международная конференция «Суперкомпьютерные дни в России»
(Москва, 25-26 сентября 2017)

Использование имитационного моделирования для улучшения планирования композитных приложений в гетерогенных вычислительных системах

Назаренко А.М., *Сухорослов О.В.*

Институт проблем передачи информации им. А.А. Харкевича
Российской академии наук



Введение

- Композитные приложения (workflows, КП)
  - Важный класс параллельных приложений, состоящих из множества задач с зависимостями между ними
  - Могут быть описаны в виде направленного ациклического графа (DAG)
- Гетерогенные вычислительные системы (ГВС)
  - Параллельные вычислительные системы, образованные из различных одиночных ресурсов, локальных или географически распределенных
  - Гетерогенность: производительность ресурсов, характеристики сетевых каналов
- Планирование выполнения композитных приложений в ГВС
  - Назначение выполнения отдельных задач КП ресурсам
  - Основная цель: минимизация времени выполнения КП (makespan)
  - Дополнительные ограничения: заданный срок, фиксированный бюджет и т.д.
  - NP-полная задача
  - Алгоритмы, основанные на различных эвристиках и метаэвристиках
  - **Требуются точные оценки времён выполнения задач и передачи данных**

Введение

- Использование в алгоритме простых моделей для оценки времени передачи данных в ГВС может приводить к существенному снижению качества планирования
  - Модель Хокни не учитывает топологию сети и разделение пропускной способности между потоками данных
- Для улучшения качества планирования предлагается использовать вместо аналитической модели более точную имитационную модель
  - В качестве примера рассматриваются модификации известного алгоритма HEFT

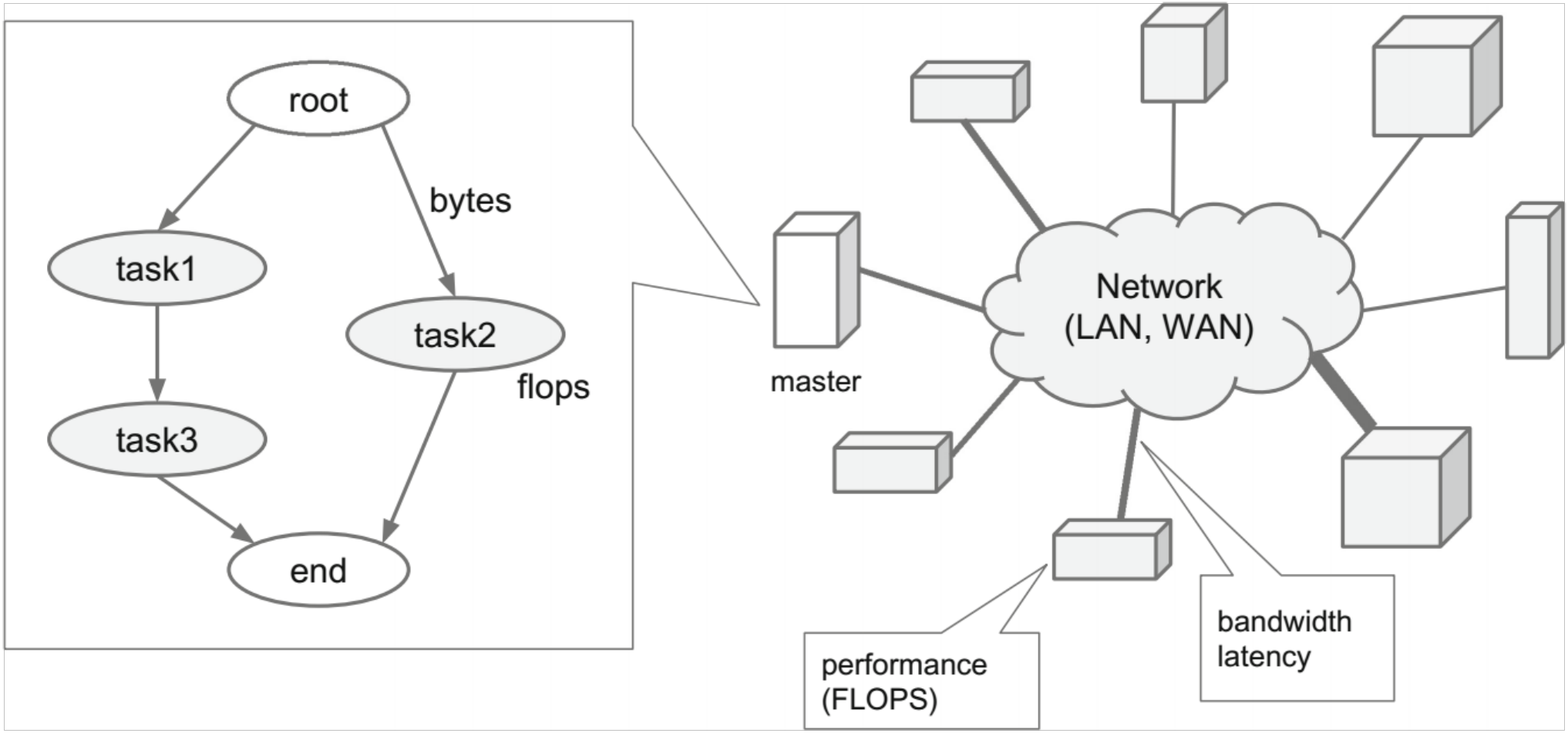
Имитационное моделирование

- Широко применяется для анализа алгоритмов планирования
  - Однако редко используется внутри самих алгоритмов
- Позволяет провести в короткие сроки большое число экспериментов для различных конфигураций систем и приложений
- Обеспечивает воспроизводимость полученных результатов
- Невысокие требования к аппаратным ресурсам
- Требует использования точных и верифицированных моделей
- WorkflowSim
  - Open source инструментарий для имитационного моделирования КП
  - Основан на симуляторе CloudSim (обнаружены ошибки в модели сети)
  - Алгоритмы: FCFS, MCT, MinMin, MaxMin, HEFT

Инструментарий pysimgrid

- Основан на SimGrid
  - Платформа для реализации симуляторов распределенных систем (грид, облака, peer-to-peer, MPI)
  - Зрелая, активно развиваемая разработка, верифицированные модели
- Библиотека на языке Python
  - Тонкий слой над SimGrid C API
  - Предоставляет удобный интерфейс для реализации и встраивания в симулятор алгоритмов планирования
- Дополнительные средства
  - Генерация синтетических систем и КП, пакетное выполнение имитационных экспериментов, анализ результатов экспериментов
- https://github.com/alexmnazarenko/pysimgrid

Модели приложения и системы



Реализованные алгоритмы

- Dynamic Level Scheduling (DLS, 1990)
- Opportunistic Load Balancing (OLB, 1998)
- Minimum Completion Time (MCT)
- Min-Min, Max-Min (1998)
- Sufferage (1999)
- Heterogeneous Earliest Finish Time (HEFT, 2002)
- Heterogeneous Critical Parent Trees (HCPT, 2003)
- Longest Dynamic Critical Path (LDCP, 2008)
- Lookahead (LA, 2010)
- Predict Earliest Finish Time (PEFT, 2014)
- Генетический алгоритм

Алгоритм HEFT

- Статический алгоритм приоритетного планирования
- Этап 1: ранжирование задач
  - Список задач сортируется в порядке убывания ранга

$$rank(T_i) = \overline{EET}(T_i) + \max_{T_j \in succ(T_i)} \left(\overline{ECOMT}(data_{ij}) + rank(T_j)\right)$$

- Этап 2: назначение задач на ресурсы
  - Задачи рассматриваются в порядке их следования в списке и назначаются на ресурс с минимальным ожидаемым временем завершения задачи

$$ECT(T_i, H_j) = EST(T_i, H_j) + EET(T_i, H_i)$$

$$EST(T_i, H_j) =$$
$$\max\{avail(H_j), \max_{T_k \in pred(T_i)} (ECT(T_k, H_k) + ECOMT(data_{ki}, H_k, H_i))\}$$

Вычисление времён передачи данных

$$\overline{ECOMT}(data_{ij}) = \overline{L} + \frac{data_{ij}}{\overline{B}}$$

$$ECOMT(data_{ij}, H_i, H_j) = L_{ij} + \frac{data_{ij}}{B_{ij}}$$

Модификации HEFT

- Цель: улучшить точность оценок *ECOMT* путём использования имитационной модели, с минимальными модификациями алгоритма
- Вариант SimHEFT
  - Этап ранжирования совпадает с HEFT
  - На этапе назначения для вычисления *ECT(T,H)* используется имитационная модель
    - Для рассматриваемой задачи T и каждого хоста H выполняется моделирование выполнения подграфа из уже назначенных задач и задачи T на хосте H
    - Выбирается хост, показавший минимальное время завершения задачи
- Вариант SimHEFT*
  - Отличается от SimHEFT критерием, используемым для выбора ресурса на этапе назначения
  - Выбирается хост с минимальным временем выполнения всего текущего подграфа из назначенных задач и задачи T
  - Цель: минимизировать деградацию makespan при планировании задач

Эксперименты

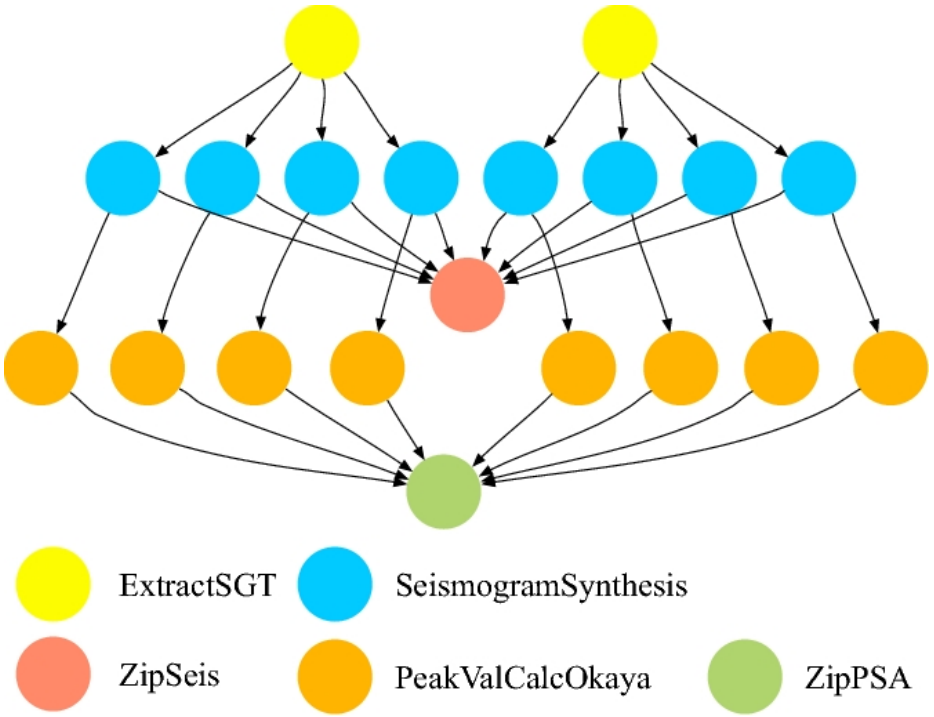
- Имитационные эксперименты, сравнивающие эффективность предложенных модификаций с HEFT и другими алгоритмами на различных примерах КП и конфигурациях ГВС
- Алгоритмы
  - HEFT
  - Opportunistic Load Balancing (OLB)
  - Minimum Completion Time (MCT)
  - SimHEFT, SimHEFT*
- Метрика эффективности – среднее нормированное время выполнения КП
  - Время выполнения КП с заданным алгоритмом нормируется на время алгоритма OLB
  - Значение усредняется по всем экспериментам

Эксперименты

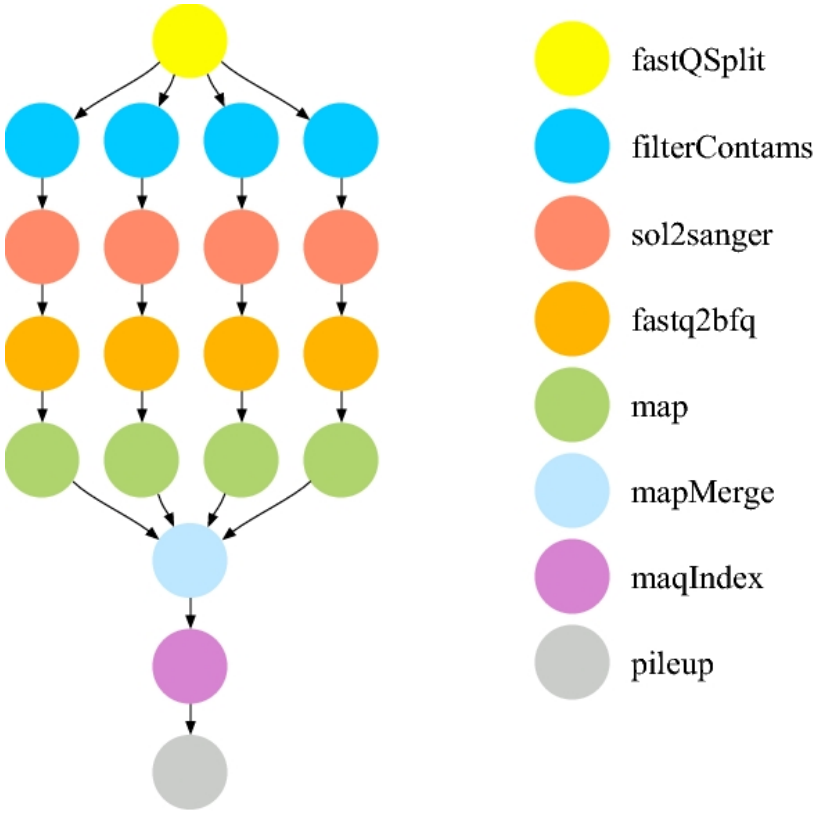
- Композитные приложения
  - Примеры на основе реальных приложений
  - LIGO Inspiral, Epigenomics, Montage, CyberShake
- Конфигурации ГВС
  - 5, 10, 20 хостов
  - Производительность варьируется от 1 до 4 ГФлопс
  - Сетевые каналы имеют одинаковые характеристики
    - Пропускная способность: 100 МБ/с
    - Латентность: 100 мкс
  - Для каждого числа хостов сгенерировано 100 систем

Приложения

CyberShake



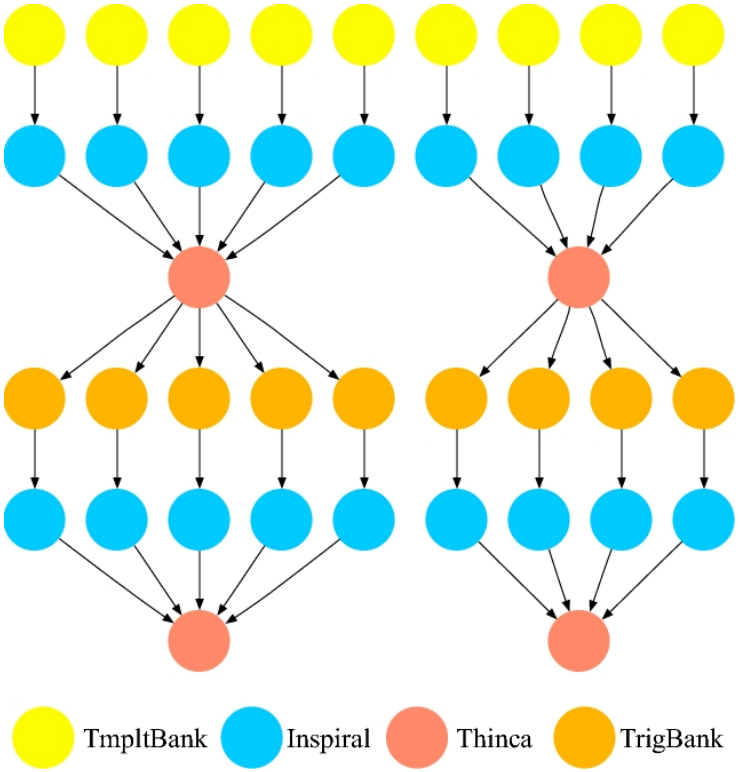
Epigenomics



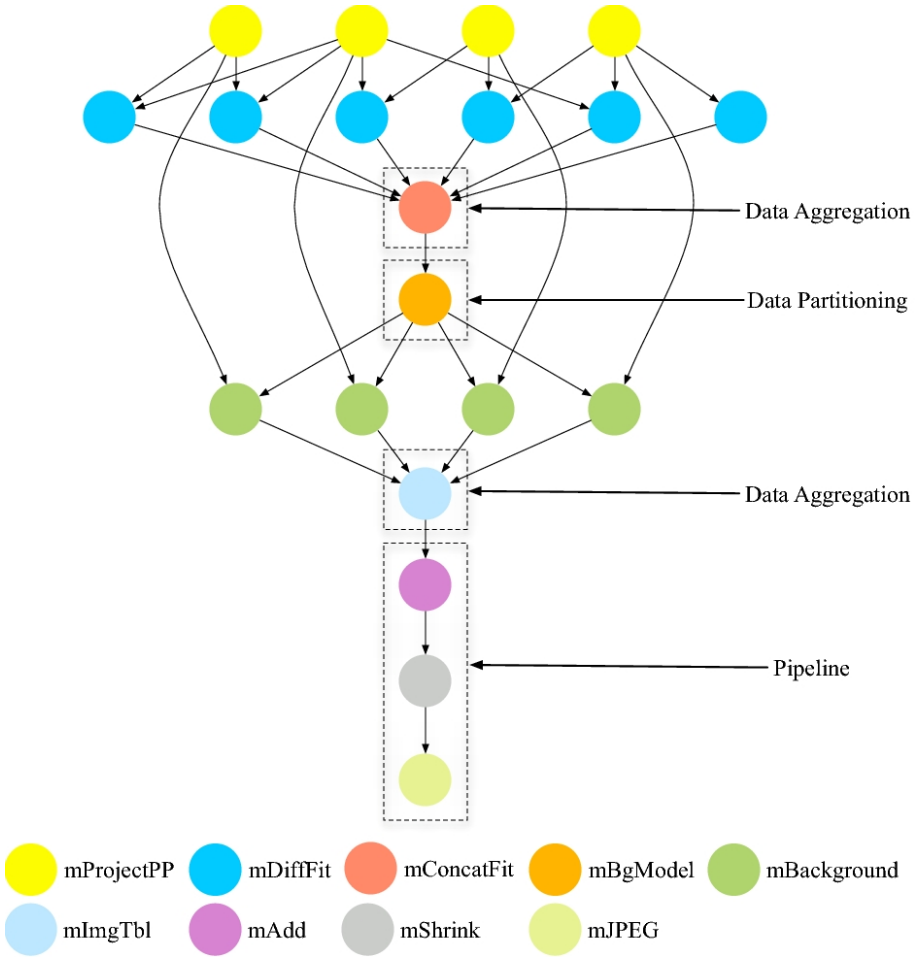
https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

Приложения

LIGO Inspiral



Montage



https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

Результаты экспериментов

Hosts count	OLB	MCT	HEFT	SimHEFT	SimHEFT*
LIGO Inspiral, 100 tasks					
5	1.0000	0.9839	0.9651 (0.9608)	0.9652	1.0229
10	1.0000	0.9182	0.8792 (0.8602)	0.8791	1.0338
20	1.0000	0.7885	0.6898 (0.6865)	0.6898	0.9384
Epigenomics, 100 tasks					
5	1.0000	0.9753	0.9376 (0.9311)	0.9376	0.9368
10	1.0000	0.9014	0.8459 (0.8405)	0.8458	0.8437
20	1.0000	0.7942	0.7093 (0.6740)	0.7099	0.7067
Montage, 100 tasks					
5	1.0000	0.9791	0.9769 (0.9683)	0.9766	0.9766
10	1.0000	0.9639	0.9635 (0.9478)	0.9629	0.9636
20	1.0000	0.9109	0.9165 (0.9023)	0.9156	0.9172
CyberShake, 100 tasks					
5	1.0000	1.0104	1.0616 (0.5074)	1.0760	1.0395
10	1.0000	0.9972	1.0846 (0.3789)	1.1354	1.0325
20	1.0000	0.9845	1.1038 (0.2958)	1.3803	1.0244

Результаты экспериментов

- Аналитическая модель в HEFT может приводить к большим ошибкам в оценке времен передачи данных и деградации качества планирования
  - Наблюдается для КП с высокими степенью параллелизма и интенсивностью обменов (CyberShake)
- Модификация SimHEFT не улучшает качество планирования для КП CyberShake
  - При оптимизации времен выполнения отдельных задач можно ухудшить времена выполнения уже назначенных задач за счет конкуренции за сетевые каналы
- Модификация SimHEFT* улучшает время выполнения КП CyberShake на 2-7% в сравнении с HEFT
  - По-прежнему отстаёт от динамического алгоритма MCT
  - Для КП с низкой интенсивностью обменов (LIGO) заметно проигрывает HEFT
- Использование имитационного моделирования значительно (до двух порядков) увеличило время работы алгоритмов

Заключение

- Исследована возможность использования имитационного моделирования вместо аналитических моделей для улучшения планирования композитных приложений в гетерогенных вычислительных системах
  - Предложены две модификации алгоритма HEFT, использующие симулятор на этапе назначения ресурсов
  - Результаты экспериментов показали, что простая замена аналитической модели на симулятор (вариант SimHEFT) не дает желаемого эффекта
  - Изменение критерия выбора ресурсов (вариант SimHEFT)* позволяет улучшить качество планирования для КП с высокими степенью параллелизма и интенсивностью обменов
- Направления развития предложенного подхода
  - Модификации фазы ранжирования
  - Адаптивное поведение алгоритма в зависимости от характеристик КП
  - Уменьшение времени планирования путем параллельного запуска имитационных моделей