

Московский государственный университет имени М.В.Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Суперкомпьютеров и Квантовой Информатики

Разработка метода прогнозирования слабой масштабируемости суперкомпьютерных приложений

Студент: Мокров К.С., 423 группа
Науч. руководитель: к.ф.-м.н., в.н.с. НИВЦ МГУ
Антонов Александр Сергеевич



Москва
2020 г.

Введение

- Масштабируемость – ключевая характеристика параллельных программ со сложным поведением.
- Существует проблема запуска задачи на большом количестве узлов: они недоступны или ожидание занимает слишком много времени.
- Актуальна задача прогнозирования масштабируемости приложения на большие конфигурации вычислительной системы.
- Предсказания можно строить, основываясь на данных, полученных из множественных запусков на малых конфигурациях.
- Разрабатываемый метод должен быть достаточно универсальным.

Постановка задачи

- Исследовать возможные подходы к предсказанию масштабируемости.
- Реализовать метод, предсказывающий слабую масштабируемость суперкомпьютерных приложений на основе экспериментальных данных.
- Проверить применимость метода на различных приложениях, собрав экспериментальную базу и оценив точность предсказаний.

Обзор существующих подходов к предсказанию масштабируемости

- Линейная регрессия.
 - Методы машинного обучения.
 - Симуляция исполнения программы.
 - Коллаборативная фильтрация.
- + Модель простая, легко искать неизвестные параметры.
 - + Для поиска параметров не требуется большое число запусков.
 - + Точность предсказаний либо сопоставима, либо даже лучше, чем у других методов
 - Далеко не всегда простая модель может описать поведение рассматриваемой величины из-за её нелинейного характера изменения

Обзор существующих подходов к предсказанию масштабируемости

- Линейная регрессия.
 - Методы машинного обучения.
 - Симуляция исполнения программы.
 - Коллаборативная фильтрация.
- + Способны уловить сложные аспекты взаимодействия между архитектурой суперкомпьютера и исследуемыми программами.
 - + Есть возможность работать при наличии неинформативных, зашумлённых входных данных.
 - Для обучения необходимо провести большое число запусков приложения.
 - Относительно высокая вычислительная стоимость процесса обучения как по времени, так и по памяти.
 - Чёрный ящик - невозможно объяснить ответ.

Обзор существующих подходов к предсказанию масштабируемости

- Линейная регрессия.
 - Методы машинного обучения.
 - Симуляция исполнения программы.
 - Коллаборативная фильтрация.
- + Сбор трасс исполнения приложения и его профилирование может быть произведено на одной машине для последующей симуляции исполнения на другой.
 - Часто требует для своей работы наличия информации о структуре программы и подробных технических характеристик используемой системы.
 - Сложно реализовать.

Обзор существующих подходов к предсказанию масштабируемости

- Линейная регрессия.
 - Методы машинного обучения.
 - Симуляция исполнения программы.
 - Коллаборативная фильтрация.
- + Давно используется в рекомендательных системах, где хорошо себя зарекомендовала.
 - Для построения точных прогнозов необходимо провести много тестирований программы и иметь доступ к большим наборам параметров запуска программы и динамических характеристик исполнения.

Описание разработанного метода

Экстраполирующая функция и оценка ошибок предсказаний

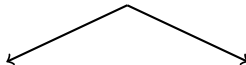
- С помощью линейной регрессии строятся предсказания значений динамических характеристик на p_{target} процессах, используя эмпирические данные, полученные из нескольких запусков на $\mathbb{Q} = \{q_1, \dots, q_n\}$, $q_1 < q_2 < \dots < q_n < p_{target}$ процессах.
- Предиктор - функция от параметров запуска:
$$DF = \hat{D}F + error = F(x_1, x_2, \dots, x_n, p) + error$$
- Параметры регрессии ищутся с помощью метода наименьших квадратов, он минимизирует:
$$|f(w, g(X)) - y|_2 = \sum_{i=1}^N (f(w, g_i(X)) - y_i)^2 \xrightarrow{w} min$$

Описание разработанного метода

Экстраполирующая функция и оценка ошибок предсказаний

- LSM \equiv минимизация абсолютной ошибки *error*.
- Качество предсказаний принято оценивать по относительным ошибкам.

$$RE = \frac{|DF - \hat{DF}|}{DF}$$



Описание разработанного метода

Экстраполирующая функция и оценка ошибок предсказаний

- LSM \equiv минимизация абсолютной ошибки *error*.
- Качество предсказаний принято оценивать по относительным ошибкам.

$$RE = \frac{|DF - \hat{DF}|}{DF}$$

$$DF = \hat{DF} + error$$

$$RE_{norm} = \frac{|error|}{DF}$$

Описание разработанного метода

Экстраполирующая функция и оценка ошибок предсказаний

- LSM \equiv минимизация абсолютной ошибки *error*.
- Качество предсказаний принято оценивать по относительным ошибкам.

$$RE = \frac{|DF - \hat{DF}|}{DF}$$

$$DF = \hat{DF} + error$$

$$RE_{norm} = \frac{|error|}{DF}$$

$$\log(DF) = \log(\hat{DF}) + error$$

$$RE_{log} = 2^{|error|} - 1$$

Описание разработанного метода

Экстраполирующая функция и оценка ошибок предсказаний

- LSM \equiv минимизация абсолютной ошибки *error*.
- Качество предсказаний принято оценивать по относительным ошибкам.

$$RE = \frac{|DF - \hat{DF}|}{DF}$$

$$DF = \hat{DF} + error$$

$$RE_{norm} = \frac{|error|}{DF}$$

$$\log(DF) = \log(\hat{DF}) + error$$

$$RE_{log} = 2^{|error|} - 1$$

- Ключевой шаг - параметризация функции
 $\log(\hat{DF}) = w_1 \cdot \log(p) + w_2 \cdot \log(N) + w_3 \cdot \log(p) \cdot \log(N)$

Описание разработанного метода

Общая схема работы метода

- 1) Определение набора тестовых конфигураций.
- 2) Проведение запусков с этими конфигурациями.
- 3) Извлечение из результатов запусков необходимых для поиска неизвестных коэффициентов модели данных, для идентичных конфигураций выбор минимума времени / максимума производительности исполнения.
- 4) Использование метода наименьших квадратов для подбора коэффициентов линейной регрессии.
- 5) Построение предсказаний значений динамических характеристик для заданного множества целевых запусков с помощью построенной модели.

Экспериментальная проверка применимости метода

- Система для тестирований: «Ломоносов-2»
- Приложения для тестирований: HPL, HPCG, алгоритмы матричного умножения DNS и SUMMA, Graph500.
- Тестирования некоторых приложений проводились с несколькими наборами конфигураций, каждый из которых определяет своя константа из отношения $T_A(N) / p = \text{const}$ (HPL - 3 набора; Graph500, DNS - 2; HPCG, SUMMA - 1).
- Количество используемых процессов на целевых конфигурациях превосходит самую большую тестовую конфигурацию в **1,2 - 6,6** раз.

Экспериментальная проверка применимости метода HPL

- HPL — тест производительности вычислительной системы. Суть теста - решение плотных систем линейных алгебраических уравнений, используя LU факторизацию.
- Сложность алгоритма $\mathcal{O}(N^3)$
- Количество операций чтения/записи $\mathcal{O}(N^2)$.
- Большое количество вычислений над плотными структурами данных.

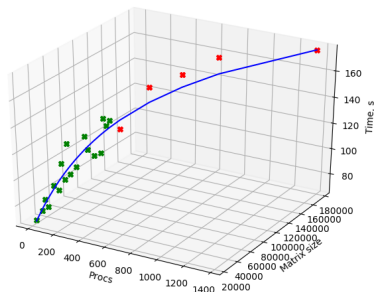


Рис.: Аппроксимирующая функция предсказаний времени, HPL

	Min	Max	Mean	Median
Time	0,02	11,35	4,12	3,82
Perf	0,07	16,35	5,23	5,69

Таблица: Относительные ошибки предсказаний, усреднённые по динамическим характеристикам, HPL

PN	225	400	576	784	1369
Mean	5,09	7,17	3,74	4,43	2,95

Таблица: Относительные ошибки предсказаний, усреднённые по количеству процессов, HPL

Экспериментальная проверка применимости метода HPCG

- HPCG — альтернативный HPL тест производительности.
- Сильно выделяется на фоне остальных приложений, так как сложность алгоритма и количество операций чтения/записи $\mathcal{O}(N)$.
- Преобладают нерегулярный доступ к памяти и рекурсивные вычисления.

PN	PS	RE_time	RE_perf
280	$PN \cdot 104^3$	0,02	0,37
560		1,56	0,80
700		1,89	13,07
980		2,85	19,54
1400		7,05	8,99

Таблица: Относительные ошибки предсказаний времени и производительности, HPCG

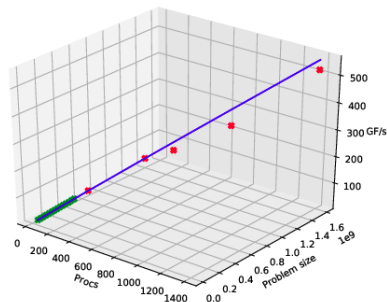


Рис.: Аппроксимирующая функция предсказаний производительности, HPCG

Экспериментальная проверка применимости метода

Алгоритмы матричного умножения, SUMMA

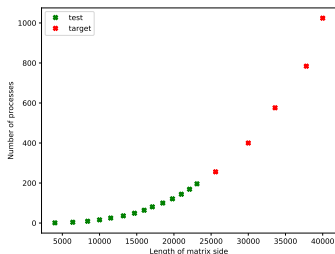


Рис.: Конфигурации запусков матричного умножения по алгоритму SUMMA

- SUMMA — алгоритм матричного умножения.
- Используется в ScaLAPACK и PBLAS.
- Сложность алгоритма $\mathcal{O}(N^3)$
- $\Rightarrow p = N^3 / \text{const}$

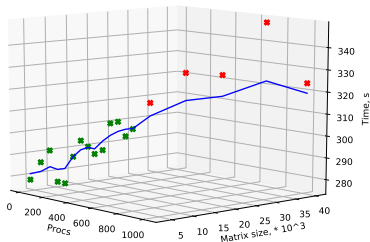


Рис.: Аппроксимирующая функция предсказаний времени, SUMMA

PN	PS	RE_time
225	25,6	1,95
400	30	3,94
576	33,6	3,01
784	37,8	7,59
1024	40	1,39

Таблица: Относительные ошибки предсказаний времени, SUMMA

Экспериментальная проверка применимости метода

Алгоритмы матричного умножения, DNS

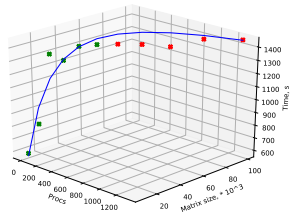
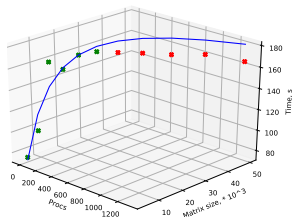


Рис.: Аппроксимирующие функции предсказаний времени для двух наборов конфигураций, DNS

- DNS — алгоритм матричного умножения.
- Сложность алгоритма $O(N^3)$
- Всего 6 тестовых конфигураций

PN	C_1		C_2	
	PS	RE_time	PS	RE_time
343	31,5	6,55	63	5,66
512	36	8,42	72	6,84
729	40,5	9,35	81	7,85
1000	45	7,94	90	1,94
1331	49,5	9,63	99	0,19

Таблица: Относительные ошибки предсказаний времени, DNS

Экспериментальная проверка применимости метода Graph500

- Graph500: BFS и SSSP.
- Сложность алгоритмом определяется через количество вершин и рёбер графа $\mathcal{O}(V + E)$.
- Выражая сложность через параметры запуска `scalefactor` и `edgfactor`, получим:
$$V + E = 2^{SC} + EF \cdot 2^{SC} = 2^{SC} \cdot (1 + EF)$$
- Сильнее всего проявляется проблема сбора тестовых конфигураций для слабой масштабируемости: они выбираются так, чтобы удовлетворять $T_A(N) / p = \text{const}$, но не всегда возможно обеспечить строгое равенство.
- $SC, EF \in \mathbb{Z}_+$. Округление сильно изменяет сложность.
- Средняя относительная ошибка предсказания времени - 13,28%, производительности - 13,22%.

Результаты

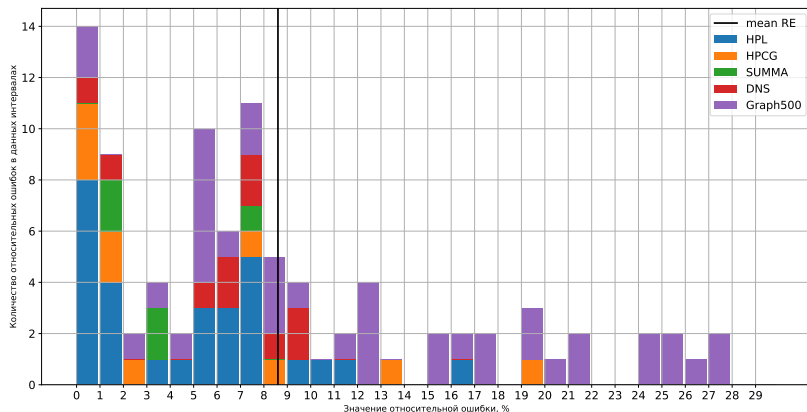


Рис.: Относительные ошибки предсказаний на всех конфигурациях всех рассматриваемых приложений

- Среднее значение относительных ошибок - 8,6% (HPL - 4,9%, HPCG - 5,6%, SUMMA - 3,6%, DNS - 6,4%, Graph500 - 13,2%)

Результаты

- Разработан метод, предсказывающий слабую масштабируемость суперкомпьютерных приложений на основе экспериментальных данных со средней относительной ошибкой по всем рассмотренным приложениям равной 8,6%.
- Выполнена проверка применимости метода на различных приложениях (HPL, HPCG, матричных алгоритмов умножения SUMMA и DNS, Graph500) на суперкомпьютере «Ломоносов-2».