

О ПРИМЕНЕНИИ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ АВТОМАТИЧЕСКОЙ НАСТРОЙКИ ПРОИЗВОДИТЕЛЬНОСТИ РЕШАТЕЛЕЙ СЛАУ В ПАРАЛЛЕЛЬНОЙ БИБЛИОТЕКЕ НАУЧНЫХ ВЫЧИСЛЕНИЙ PETSC

В.Ю. Воронов, Н.Н. Попова

Один из современных подходов разработки масштабируемых научных приложений - использование высокоуровневых научных библиотек, предоставляющих пользователю готовые реализации параллельных численных методов и средства поддержки для работы с данными, отладки. Использование подобных библиотек позволяет получать приложения, переносимые на разные высокопроизводительные архитектуры, предоставляет пользователю широкий выбор эффективных решателей и максимально скрывает от него специфику программирования на параллельных платформах. Помимо этого, библиотеки позволяют создавать высокопроизводительное приложение, манипулируя объектами и структурами данных на уровне предметной области, что снижает трудоемкость разработки и отладки приложения в сравнении с низкоуровневыми средствами разработки.

В данной работе рассматривается библиотека PETSc [1], ориентированная на решение систем линейных алгебраических уравнений (СЛАУ) большой размерности с разреженной матрицей коэффициентов. Выбор для исследования решателя СЛАУ обусловлен тем, что решение СЛАУ лежит в основе численных методов большого класса научных приложений и на ее решение может приходиться до 85% операций с плавающей точкой всего приложения. Таким образом, настройка производительности научного приложения именно на этом уровне решения численной задачи позволяет получить наиболее существенный выигрыш. Для решения разреженных СЛАУ библиотека PETSc предоставляет пользователю параллельные итерационные методы крыловского типа, релаксационные методы, вычислительная сложность которых для разреженных матриц $O(mn^2)$, где m - число итераций, n - размерность матрицы; последовательные предобуславливатели, основанные на неполной факторизации (ILU, ICC) и параллельные предобуславливатели (на основе методов Якоби и Гаусса-Зейделя); последовательные реализации прямых методов (LU-декомпозиция, декомпозиция Холецкого, QR-декомпозиция), требующих $O(n^3)$ операций.

Одна из актуальных задач для развития библиотек - создание механизмов, автоматически повышающих производительность приложений (automatically tuned performance).

Общие подходы к настройке производительности решателей СЛАУ можно разделить на две группы: методы настройки на основе архитектурных особенностей платформы и на основе анализа входных данных. Наиболее известные представители первой группы - пакеты ATLAS [7] и OSKI [6]. Эти приложения реализуют идею автоматической настройки функций пакета базовых операций линейной алгебры BLAS, настройке подвергаются операции вида "матрица-вектор" и "матрица-матрица" с целью более эффективного использования кэш-памяти и других особенностей платформы. В пакете ATLAS настройка осуществляется двумя основными способами: варьирование параметров функций с выбором оптимальных настроек на этапе выполнения программы и создание множества реализаций функции, с выбором определенной реализации функции для данной архитектуры. К отличительным особенностям пакета OSKI следует отнести возможность использования априорной информации пользователя о структуре заполнения матрицы, о характере и потоке вычислений в приложении. В результате работы OSKI осуществляется более эффективное использование кэш-памяти, кроме того пакет выполняет преобразование исходной матрицы, причем пользователь имеет возможность получить это преобразование явно в виде кода на языке OSKI-Lua и затем использовать самостоятельно. Авторы замечают, что настройка умножения требует в среднем в 40 раз больше FLOPS, чем умножение "матрица-вектор" целесообразность применения настройки в конкретном случае должен определять пользователь исходя из специфики своего приложения. Образец подхода настройки на основе анализа матрицы коэффициентов СЛАУ - работа [8], авторы которой решают задачу прогнозирования числа обусловленности матрицы. Задача представляется в виде проблемы классификации, предлагается алгоритм прогнозирования в выбранном пространстве признаков. Результаты обучения и тестирования классификатора показывают ошибку прогнозирования в среднем 55%, однако такого качества прогнозирования достаточно для того, чтобы судить о характере обусловленности матрицы без трудоемкого вычисления обращения матрицы. Данный подход находит применение в системе IPRS, разрабатываемой авторами в University of Kentucky, Lexington для автоматического выбора предобуславливателя СЛАУ.

В представленной работе предлагается для настройки производительности библиотечных приложений осуществлять автоматический выбор решателя и значений его параметров. Наша предыдущая работа [9], а также результаты работы [3] показывают, какое существенное влияние на производительность оказывает выбор решателя и его настроек. Кроме того, по умолчанию библиотека настроена на наиболее консервативный решатель СЛАУ, с це-

лю обеспечить устойчивость для очень широкого класса приложений, что очевидно сказывается на производительности.

Предлагаемый алгоритм автоматической настройки. Интуитивно представляя задачей выбор наилучшего из доступных в пакете PETSc решателя и значений его набора параметров в зависимости от характеристик входных данных, представим формальную модель предметной области и дадим ее интерпретацию для процесса решения разреженной СЛАУ $Ax=b$. Для анализа производительности в предлагаемой модели использование библиотечного решателя СЛАУ для задачи i представляется функционалом качества Q :

$$Q:(F_i; S_j) \rightarrow P_{ij}$$

где $F_i=(f_1; f_2; \dots; f_n)^T$ – вектор входных признаков для задачи i , $S_j=(s_1; s_2; \dots; s_m)^T$ – вектор настроек решателя, $P_{ij}=(p)$ – вещественная метрика производительности выбранного решателя для заданной СЛАУ. Входные признаки – вещественные характеристики, получаемые в результате анализа матрицы коэффициентов A . Настройки решателя – это применяемые прямой метод (или итерационный метод с предобуславливателем) и значения их настроек. Метрика производительности – используется характеристика *time*memory*.

Задачу определения наилучших параметров решателя в предложенной модели можно свести к решению задачи поиска S_j , на котором достигается минимум функционала качества $\min_{S_j} Q$ и применять для ее решения методы оптимизации. В связи с тем, что вычисление функционала качества является чрезвычайно трудоемкой операцией, применение оптимизационных методов приводит к вычислительной нагрузке во время решения задачи. Для методов, основанных на обучении с учителем, вычислительная нагрузка приходится на этап обучения, процесс применения метода на СЛАУ не является столь сложным. По этой причине нами был предложен алгоритм автоматической настройки решателя СЛАУ, основанный на методах машинного обучения.

1. Определение обучающей выборки $L=F_i$ и формирование статистики $stat:(F_i; S_j; P_{ij})$

2. Для каждой матрицы F_i обучающей выборки определить класс $c_{ij}=(P_{ij}-P_{imin})/P_{imax}$. Параметр с очевидно определяет производительность данного решателя для задачи. Значение 0 соответствует наилучшей производительности, значение 1 – наихудшей производительности.

3. Создать и обучить на статистике (F_i, S_j, c_{ij}) классификатор $C:(F_i, S_j) \rightarrow c_{ij}$. Предлагаемый механизм классификации -- Support Vector Machine [4].

4. Для матриц тестовой выборки $L=F_i$ случайным образом инициализировать набор k гипотез параметров решателя $S=S_j$, применить построенный классификатор C . Полученные значения прогноза c_{ij} использовать для оценивания гипотез и выбора наилучшего решателя.

Предлагаемый метод настройки удобен для рассмотрения различных моделей решателя (изменение пространства признаков, пространства настроек решателя СЛАУ и т.д.). Вычислительный эксперимент по реализации предложенного метода проводится на платформе МВС-100К МЦС РАН, использована библиотека PETSc версии 2.3.3-p8, с дополнительно подключенными библиотеками BLAS и LAPACK версии 3.1.1. Для извлечения характеристик матриц коэффициентов и формирования признакового пространства используется пакет AnaMod, разработанный в University of Tennessee, Knoxville. Пакет позволяет получить для матрицы СЛАУ три группы характеристик: скалярные, структурные, спектральные; всего 63 основных признака. Работа со сформированной статистикой, создание классификаторов проводится на основе библиотеки операций машинного обучения Shogun [5]. Источник обучающих и проверяющих выборок СЛАУ -- класса данных University of Florida sparse matrix repository [2], в котором собраны СЛАУ, возникающие из ряд научных и промышленных проблем; отобраны задачи с числом ненулевых элементов матрицы от 1 до 50 миллионов; пример признакового пространства для одной из выборок матриц представлен на рис. 1.

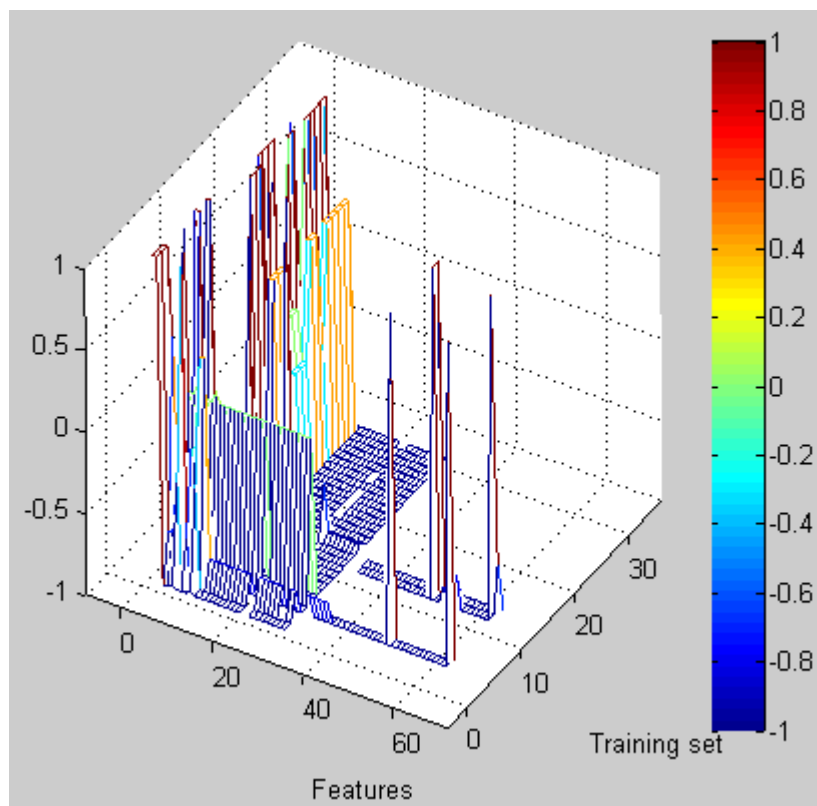


Рис. 1. Распределение характеристик обучающей и тестовой выборки после нормировки к интервалу [-1,1]. Всего 31 разреженная матрица.

При определении решателя СЛАУ рассматривались итерационные решатели крыловского типа с предобуславливателями, прямые решатели; основные характеристики представлены в таб. 1.

Метод решения	Предобуславливатель	Метод перенумерации (ordering)	Значения дополнительных параметров
Метод сопряженных градиентов(CG)	ICholesky(k)	Nested Dissections, Reverse Cuthill-McKee, One-way Dissections, Quotient Minimum Degree	Level of fill: 0, 1, 2 Fill Factor: 3, 5, 8, 100 Pivot threshold: 0.0, 0.1
Метод обобщенных минимальных невязок (GMRES)	ILU(k)		GMRES Restart: 30, 65, 100
LU-декомпозиция	-		Drop Tolerance: 10^{-1} , 10^{-3} , 10^{-7} , 10^{-9}
Декомпозиция Холецкого	-		Level of fill: 0, 1, 2 Fill Factor: 3, 5, 8, 10 Pivot threshold: 0.0, 0.1
			Level of fill: 0, 1, 2 Fill Factor: 3, 5, 8, 10 Pivot threshold: 0.0, 0.1

Таб. 1. Основные характеристики решателя СЛАУ в предлагаемой модели.

В докладе будет представлен анализ предложенного метода в разрезе применения к отдельным классам задач, анализ качества получаемого прогноза и применяемые способы повышения качества результата, анализ применимости алгоритма в различных моделях решателя, вопросы создания признакового пространства и особенностей статистики.

Предложенный метод может быть обобщен на более широкую задачу выбора не только решателя СЛАУ и его параметров в библиотеке, но и выбора решателя среди нескольких библиотек и даже среди библиотек нескольких вычислительных платформ. Для этого планируется провести исследование на платформах IBM BlueGene/P ВМК МГУ и «СКИФ-Чебышев» МГУ. Представленный метод автоматического выбора решателя СЛАУ в настоя-

щее время находит реализацию в инструментальной системе, разрабатываемой на факультете ВМК МГУ. Данная инструментальная система даст пользователям массивно-параллельных вычислительных платформ по представленным входным данным автоматически выбирать наиболее производительный решатель и значения параметров. Работы проводятся при поддержке гранта РФФИ 08-07-00445-а.

ЛИТЕРАТУРА:

1. S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>. URL <http://www.mcs.anl.gov/petsc-2>.
2. T. Davis. University of Florida sparse matrix collection. *NA Digest*, 97(23):7, 1997.
3. A. Gupta, T. George, and V. Sarin. An Experimental Evaluation of Iterative Solvers for Large SPD Systems of Linear Equations. Technical Report RC 24479, IBM T. J. Watson Research Center, Yorktown Heights, NY, 2008.
4. A.J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.
5. S. Sonnenburg, G. Ratsch, C. Schölkopf, and B. Schölkopf. Large Scale Multiple Kernel Learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.
6. R. Vuduc, J.W. Demmel, and K.A. Yelick. OSKI: A library of automatically tuned sparse matrix kernels. *Journal of Physics: Conference Series*, 16(1):521–530, 2005.
7. R.C. Whaley, A. Petitet, and J.J. Dongarra. Automated empirical optimizations of software and the ATLAS project. *Parallel Computing*, 27(1-2):3–35, 2001.
8. S. Xu and J. Zhang. A New Data Mining Approach to Predicting Matrix Condition Numbers. *Commun. Inform. Systems*, 4(4):325–340, 2004.
9. Н. Н. Попова, В. Ю. Воронов, О. В. Джосан, М. А. Медведев. Сравнительный анализ эффективности параллельных вычислений с использованием современных параллельных математических библиотек на примере решения систем линейных уравнений. Труды Всероссийской научной конференции Научный Сервис в Сети Интернет-2004, 2004.