

Лекция: Параллельные математические библиотеки

Василий Воронов
basrav@angel.ctc.msu.ru

Факультет ВМК МГУ имени М.В. Ломоносова

Летняя школа «Суперкомпьютерное
моделирование и визуализация данных»
04-14 июля 2010 г.

Что дают библиотеки пользователю?

- выбор: производительность \Leftrightarrow скорость разработки
- работа на уровне объектов предметной области
- разработка кода без знания тонкостей параллельного программирования
- стандартизация в hpc, путь к «взрослению» software development lifecycle в научных вычислениях

План лекции

- Базовые библиотеки линейной алгебры
 - BLAS, LAPACK, ScaLAPACK, ATLAS, PLASMA, Volkov LAPACK
- Библиотека быстрых преобразований Фурье FFTW
- Библиотеки разделения графов *Metis
 - Metis, hMetis, ParMetis
- Высокоуровневые научные библиотеки: Trilinos

Цель: дать обзор возможностей некоторых современных пакетов

Линейная алгебра и параллельные вычисления

- Одна из первых и основных областей приложения параллельных вычислений
- Стандартизация: библиотеки EISPACK(1974), LINPACK(1982), LAPACK(1987), ScaLAPACK(1992)
- Де-факто стандарты матрично-векторных операций: BLAS, LAPACK www.netlib.org/blas,
www.netlib.org/lapack

BLAS: Basic Linear Algebra Routines

- Интерфейс пользователя-3 уровня операций
 - Вектор-вектор $y = \alpha x + \beta$
 - Матрица-вектор $y = \alpha Ax + b$
 - Матрица-матрица $B = \alpha AB + \beta C$
- Реализации от OEM: Intel MKL(нитевая), IBM ESSL(нитевая), Cray LibSci(нитевая), AMD ACML. HP MLIB, NEC PDLIB...
- «Быстрые» сторонние реализации: OSKI, GotoBLAS (нитевая), ATLAS
- «Экзотика»: интерфейс для Scheme (SENSAWAVE)

Структура функции BLAS

- Тип данных S[ingle], D[ouble], C[omplex], Z[double complex]
- 2 буквы — формат матрицы (11 форматов)
- 1-2 буквы — тип операции (в зависимости от BLAS-уровня)

Примеры: DGEMM (TRANSA, TRANSB, M, N, K, ALPHA, A, LDA, B, LDB, BETA, C, LDC) для

$$C = \alpha AB + \beta C$$

Пример кода на BLAS

```
int main (void) {  
    double a[] = { 0.11, 0.12, 0.13,  
                  0.21, 0.22, 0.23 };  
    double b[] = { 1011, 1012,  
                  1021, 1022,  
                  1031, 1032 };  
    double c[] = { 0.00, 0.00,  
                  0.00, 0.00 };  
  
    gsl_matrix_view A = gsl_matrix_view_array(a, 2, 3);  
    gsl_matrix_view B = gsl_matrix_view_array(b, 3, 2);  
    gsl_matrix_view C = gsl_matrix_view_array(c, 2, 2);  
  
    /* Compute C = A B */  
  
    gsl_blas_dgemm (CblasNoTrans, CblasNoTrans,  
                  1.0, &A.matrix, &B.matrix, 0.0, &C.matrix);  
  
    printf (" [ %g, %g\n", c[0], c[1]);  
    printf (" %g, %g ]\n", c[2], c[3]);  
    return 0;  
}
```

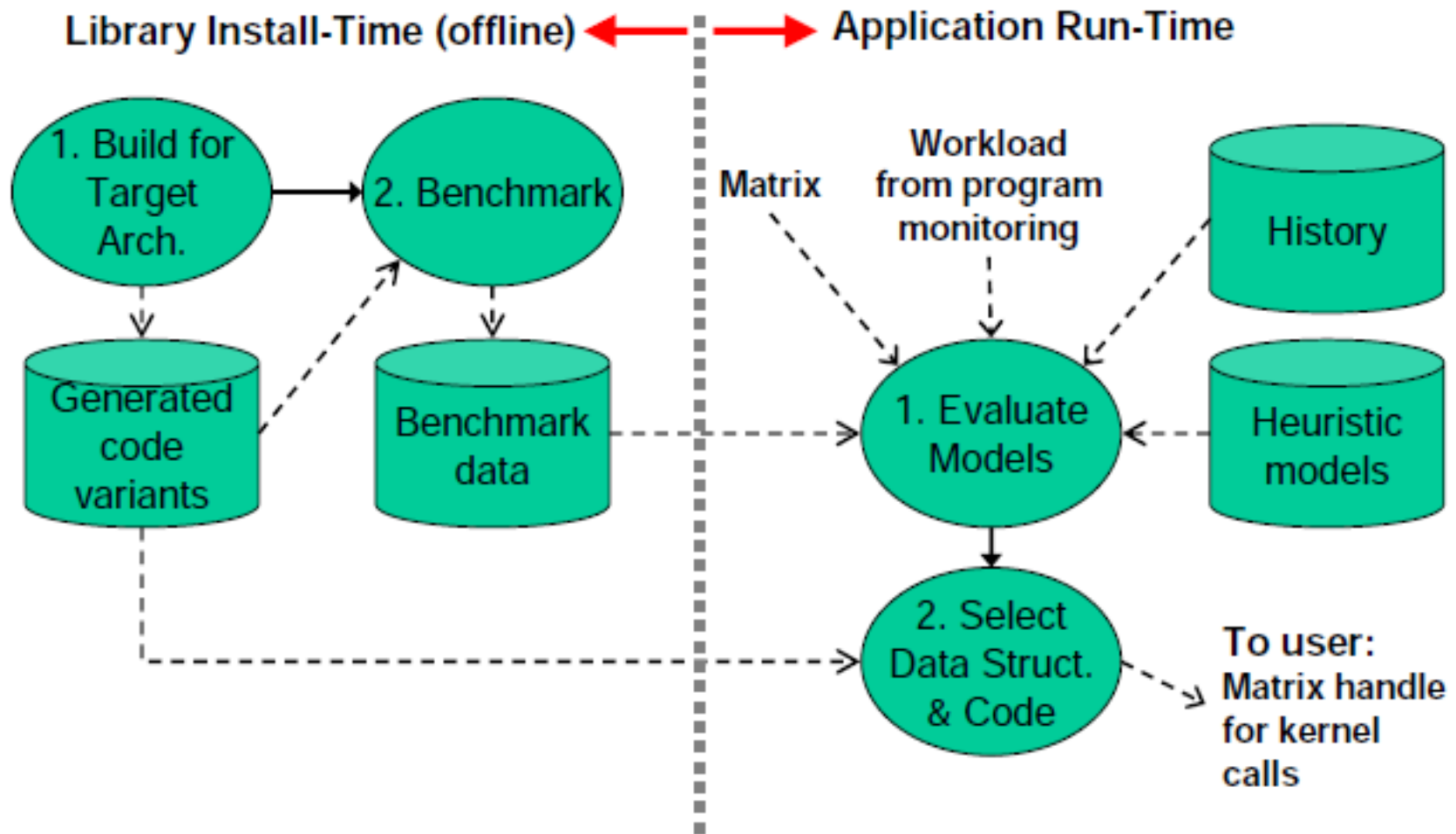
Вывод:

```
$ ./a.out  
[ 367.76, 368.12  
674.06, 674.72 ]
```

«Особенные» BLAS

- **GotoBLAS** www.tacc.utexas.edu/index.php?id=93:
 - оптимизированная реализация функций
- **ATLAS** <http://math-atlas.sourceforge.net>
 - Адаптивная настройка функций под архитектуру
 - Фаза «настройки» с линейным поиском параметров функций
- **OSKI** <http://bebop.cs.berkeley.edu/oski/>
 - Матрично-векторные операции + диагональные решатели
 - Настройка функций под особенности данных
 - Эвристики и поисковые алгоритмы

OSKI: структура



Механизм настройки в OSKI

- Язык OSKI-Lua для текстового описания преобразований матриц
- Сохранить трансформацию данных и использовать ее в других задачах

```
/* In "my_app.c" */  
fp = fopen("my_xform.txt", "rt");  
fgets(buffer, BUFSIZE, fp);  
  
oski_ApplyMatTransform(A_tunable,  
    buffer);  
oski_MatMult(A_tunable, ...);
```

```
# In file, "my_xform.txt"  
# Compute  $A_{fast} = P * A * P^T$  using  
    Pinar's reordering algorithm  
A_fast, P =  
    reorder_TSP(InputMat);  
# Split  $A_{fast} = A_1 + A_2$ , where  $A_1$  in 2x2  
    block format,  $A_2$  in CSR  
A1, A2 =  
    A_fast.extract_blocks(2, 2);  
  
return transpose(P) * (A1+A2) * P;
```

LAPACK: алгоритмы линейной алгебры

- Основная функциональность
 - Факторизация матриц (LU, QR, ...)
 - Решение систем линейных уравнений
 - Решение задач на собственные значения
 - Решение задач наименьших квадратов
- Последовательная реализация, работа с разными типами данных, 30+ типов матриц, 100+ функций

Пример кода на LAPACK

```
#include <stdio.h>
#include <atlas_enum.h>
#include "clapack.h"

double m[] = {
    3, 1, 3,
    1, 5, 9,
    2, 6, 5
};

double x[] = {
    -1, 3, -3
};

int main() {
    int ipiv[3];
    int i, j;
    int info;
    for (i=0; i<3; ++i) {
        for (j=0; j<3; ++j) printf("%5.1f", m[i*3+j]);
        putchar('\n');
    }
    info = clapack_dgesv(CblasRowMajor, 3, 1, m, 3, ipiv,
        x, 3);
    if (info != 0) fprintf(stderr, "failure with error %d\n",
        info);
    for (i=0; i<3; ++i) printf("%5.1f %3d\n", x[i], ipiv[i]);
    return 0;
}
```

ScaLAPACK www.netlib.org/scalapack

- Реализует подмножество LAPACK для вычислителей с распределенной памятью
- Использует собственную библиотеку коммуникаций BLACS
- Требует явного распределения данных по процессам
- Де-факто стандарт операций линейной алгебры 15+ лет

Current trends: PLASMA, MAGMA, Volkov LAPACK

- PLASMA — параллельный LAPACK для архитектуры cluster of multicores
www.icl.cs.utk.edu/plasma
- MAGMA — параллельная линейная алгебра на гибридных и неоднородных архитектурах <http://icl.cs.utk.edu/magma>
- Volkov LAPACK — подмножество LAPACK для GPU <http://www.cs.berkeley.edu/~volkov/>
- CUBLAS — быстрая NVIDIA-реализация для GPU
http://developer.download.nvidia.com/compute/cuda/2_0/docs/CUBLAS_Library_2.0.pdf

FFTW: пакет быстрых преобразований Фурье

- Одна из самых быстрых реализаций FFT-преобразований www.fftw.org/
- Алгоритмы Cooley-Tuskey на основе декомпозиции данных
- Многонитевая, отдельно версия для MPI
- 30+ функций для FFT, DCT, DST преобразований FFT, DCT, DST
- Адаптивная настройка под платформу (wisdom, guru)
- Аналог: OpenCL FFT под GPU
http://developer.apple.com/mac/library/samplecode/OpenCL_FFT/index.html

FFTW: пример прямого преобразования DFT в 1D

```
#include <fftw3.h>

...

{
    fftw_complex *in, *out;

    fftw_plan p;

    ...

    in = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * N);
    out = (fftw_complex*) fftw_malloc(sizeof(fftw_complex) * N);
    p = fftw_plan_dft_1d(N, in, out, FFTW_FORWARD, FFTW_ESTIMATE);

    ...

    fftw_execute(p); /* repeat as needed */

    ...

    fftw_destroy_plan(p);

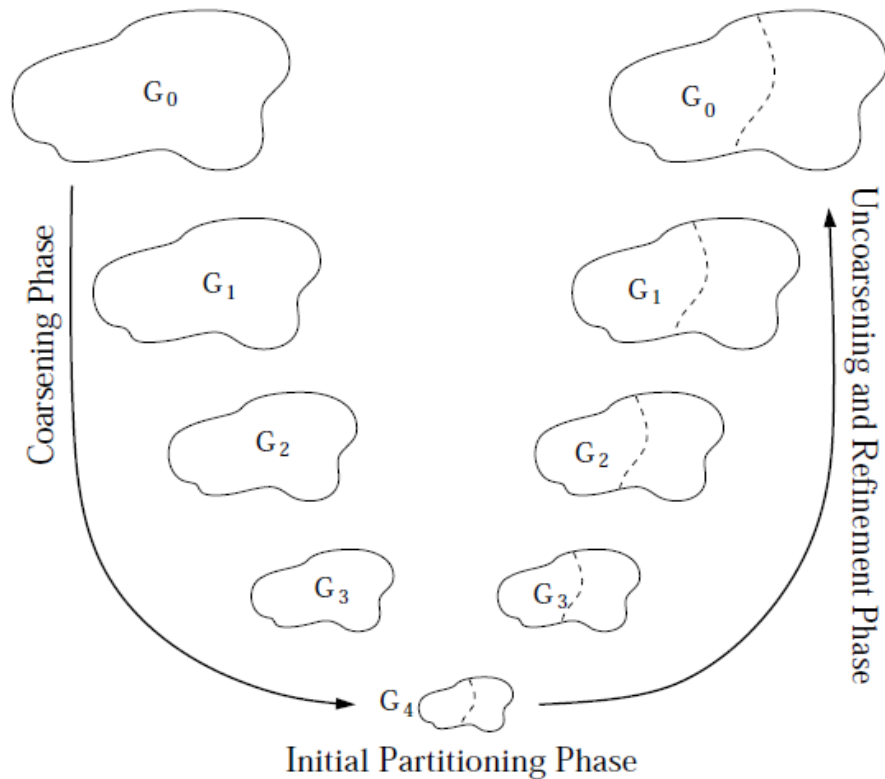
    fftw_free(in); fftw_free(out);
}

```


*METIS: пакеты разбиения графов на области

- Библиотеки Metis, hMetis, ParMetis
www.cs.umn.edu/~metis
- Metis: разбиение графов, сеток
- hMetis: разбиение гиперграфов (VLSI, data mining,...)
- ParMetis: параллельное разбиение графов, сеток
- Задачи разбиения графов NP, решаются эвристически (алгоритмы бисекции,...)

METIS: k-уровневое разбиение графов



- Граф $G = (V, E)$
- Разбить граф на области так, чтобы минимизировать число ребер разных областей
- Многоуровневое огрубление графа, разбиение и уточнение до исходного
- Сложность $O(|E|)$, в отличие от $O(|E| \log k)$

Особенности *METIS

- hMetis реализует разбиение гиперграфов $H = (E, V)$
- *METIS работают с несколькими форматами представления графов
- ParMetis — параллельное k-разбиение внутри каждой области
- В ParMetis **известны ошибки**, но поддерживаются Metis и hMetis
- Аналог: пакет Zoltan <http://www.cs.sandia.gov/Zoltan/>

Современные библиотеки: Trilinos

- Trilinos <http://trilinos.sandia.gov> - набор научных компонентов, связанных едиными интерфейсами и структурами данных
- Более 20 компонент для решения задач из разных областей научных вычислений
- Инфраструктурные компоненты Epetra, TriUtils
- Интерфейсы к нескольким языкам, средства ввода-вывода данных, анализа производительности,...

Что почитать?

- Курс James Demmel CS 267'2008
<http://www.cs.berkeley.edu/~demmel>
- Курсы Yosef Saad CSCI 5451'2010, CSCI 8314'2009 <http://www-users.cs.umn.edu/~saad/>
- Курс Jack Dongarra CS594'2010
<http://www.netlib.org/utk/people/JackDongarra/>
- Parallel Computing, J. on Supercomputing, SIAM J. on Scientific and Statistical Computing, J. of the ACM, J. of Parallel and Distributed Computing...