Федеральное государственное учреждение
ФЕДЕРАЛЬНЫЙ НАУЧНЫЙ ЦЕНТР
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
СИСТЕМНЫХ ИССЛЕДОВАНИЙ
Российской академии наук

# Использование параллельных вычислений для моделирования многокомпонентной фильтрации при разработке месторождений нефти и газа

**В.А. Бахтин**, А.В. Королев, Н.В. Поддерюгина

bakhtin@keldysh.ru, alexandre.korolev@mail.ru, konov@keldysh.ru

**Сургут, 2016**

`

Федеральное государственное учреждение
ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ ИМ. М.В. КЕЛДЫША
Российской академии наук

# План доклада

- Автоматизация разработки параллельных программ
  - Система DVM
  - Система САПФОР
- Композиционная модель фильтрации
- Распараллеливание программы Композит

TOP 500 The List.

PRESENTED BY

ICL UT INNOVATIVE COMPUTING LABORATORY THE UNIVERSITY of TENNESSEE

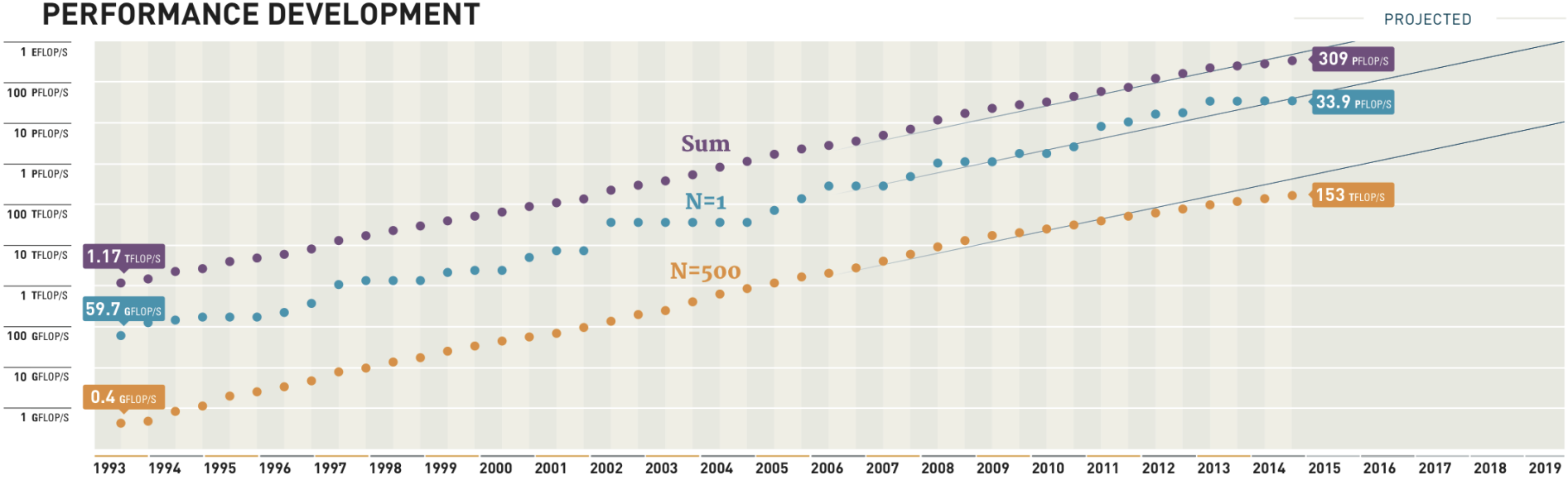
BERKELEY LAB Lawrence Berkeley National Laboratory

PROMETEUS PROFESSOR MEUER TECHNOLOGIEBERATUNG

FIND OUT MORE AT top500.org

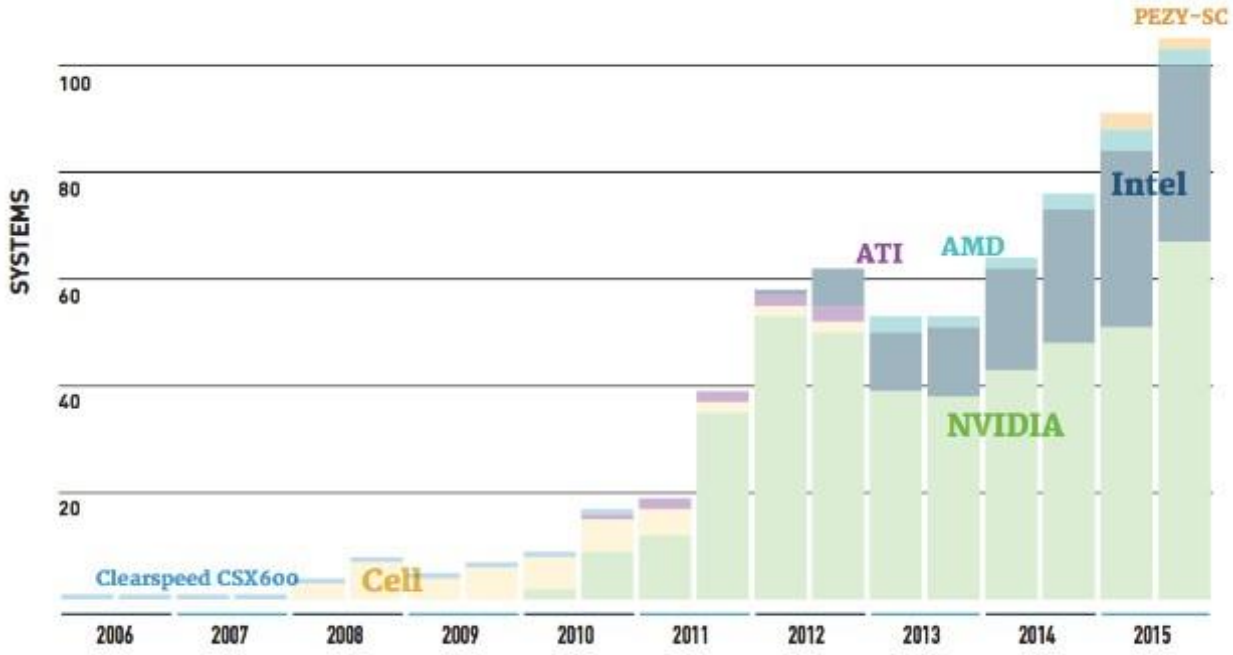
| | NAME | SPECS | SITE | COUNTRY | CORES | RMAX PFLOP/S | POWER MW |
|---|---|---|---|---|---|---|---|
| 1 | Tianhe-2 (Milkyway-2) | NUDT, Intel Ivy Bridge (12C, 2.2 GHz) & Xeon Phi (57C, 1.1 GHz), Custom interconnect | NSCC Guangzhou | China | 3,120,000 | 33.9 | 17.8 |
| 2 | Titan | Cray XK7, Opteron 6274 (16C 2.2 GHz) + Nvidia Kepler GPU, Custom interconnect | DOE/SC/ORNL | USA | 560,640 | 17.6 | 8.2 |
| 3 | Sequoia | IBM BlueGene/Q, Power BQC (16C 1.60 GHz), Custom interconnect | DOE/NNSA/LLNL | USA | 1,572,864 | 17.2 | 7.9 |
| 4 | K computer | Fujitsu SPARC64 VIIIfx (8C, 2.0GHz), Custom interconnect | RIKEN AICS | Japan | 705,024 | 10.5 | 12.7 |
| 5 | Mira | IBM BlueGene/Q, Power BQC (16C, 1.60 GHz), Custom interconnect | DOE/SC/ANL | USA | 786,432 | 8.59 | 3.95 |

## PERFORMANCE DEVELOPMENT



# Top 500. Использование ускорителей

| Accelerator | Count | System Share (%) | Rmax (GFlops) | Rpeak (GFlops) | Cores |
|---|---|---|---|---|---|
| Nvidia Kepler | 52 | 10.4 | 59,907,300 | 90,909,313 | 1,529,135 |
| Intel Xeon Phi | 29 | 5.8 | 57,080,027 | 89,469,693 | 4,903,752 |
| Nvidia Fermi | 14 | 2.8 | 8,891,651 | 18,165,047 | 691,712 |
| Hybrid | 4 | 0.8 | 8,142,240 | 12,403,928 | 681,400 |
| ATI Radeon | 3 | 0.6 | 1,148,600 | 2,280,349 | 94,304 |
| PEZY-SC | 2 | 0.4 | 812,196 | 1,908,851 | 1,444,736 |



# Алгоритм Якоби в модели DVMH

```
 PROGRAM    JACOBY_DVMH
            PARAMETER    (L=4096,  ITMAX=100)
            REAL     A(L,L), B(L,L)
!DVM$       DISTRIBUTE    ( BLOCK,   BLOCK)   ::   A
!DVM$       ALIGN  B(I,J)  WITH  A(I,J)
            PRINT *,  '**********  TEST_JACOBI   **********'
            DO IT  =  1,  ITMAX
!DVM$           REGION INOUT(A,B)
!DVM$               PARALLEL (J,I) ON  A(I,  J)
                    DO  J  =  2, L-1
                        DO  I  =  2, L-1
                            A(I, J)  =  B(I, J)
                        ENDDO
                    ENDDO
!DVM$               PARALLEL (J,I) ON B(I,  J), SHADOW_RENEW (A)
                    DO  J = 2,  L-1
                        DO  I = 2,  L-1
                            B(I, J) =  (A(I-1, J) + A(I, J-1) +  A(I+1, J) + A(I, J+1)) / 4
                        ENDDO
                    ENDDO
!DVM$           END REGION
                ENDDO
!DVM$       IO_MODE(LOCAL, ASYNCHRONOUS)
            OPEN(3, FILE='JAC.DAT', ACCESS='STREAM')
!DVM$       GET_ACTUAL(B)
            WRITE(3, *) B
            CLOSE(3)
            END
```

© ФГУ ФНЦ НИИСИ РАН
© ИПМ ИМ. М.В. КЕЛДЫША РАН

# Алгоритм Якоби в модели DVMH

```
FILE *f;
#pragma dvm array distribute[block][block] shadow[1:1][1:1]
double A[L][L];
#pragma dvm array align ([i][j] with A[i][j])
double B[L][L];
int main(int argc, char *argv[]) {
   for(int it = 0; it < ITMAX; it++) {
       #pragma dvm region inout (A,B)
      {
         #pragma dvm parallel([i][j] on A[i][j])
         for (int i = 1; i < L - 1; i++)
            for (int j = 1; j < L-1; j++)  A[i][j] = B[i][j];
         #pragma dvm parallel([i][j] on B[i][j]), shadow_renew(A)
         for (int i = 1; i < L - 1; i++)
            for (int j = 1; j < L - 1; j++)
               B[i][j] = (A[i - 1][j] + A[i + 1][j] + A[i][j - 1] + A[i][j + 1]) / 4.;
      }
   }
  f = fopen("jacobi.dat", "wbsl");
  #pragma dvm get_actual(B)
  fwrite(B, sizeof(double), L * L, f);
  fclose(f);
  return 0;
}
```

# Средства программирования

**C-DVMH = Язык Си + специальные прагмы**

**Fortran-DVMH = Язык Фортран 95 + специальные комментарии**

- Специальные комментарии и прагмы являются высокоуровневыми спецификациями параллелизма в терминах последовательной программы
- Отсутствуют низкоуровневые передачи данных и синхронизации
- Последовательный стиль программирования
- Спецификации параллелизма «невидимы» для стандартных компиляторов
- Существует единственный экземпляр программы для последовательного и параллельного счета

# Спецификации параллельного выполнения программы

- Распределение элементов массива между процессорами
- Распределение витков цикла между процессорами
- Спецификация параллельно выполняющихся секций программы (параллельных задач) и отображение их на процессоры
- Организация эффективного доступа к удаленным (расположенным на других процессорах/ускорителях) данным

# Спецификации параллельного выполнения программы

- Организация эффективного выполнения редукционных операций - глобальных операций с расположенными на различных процессорах/ускорителях данными (таких, как их суммирование или нахождение их максимального или минимального значения)
- Определение фрагментов программы (регионов) для возможного выполнения на ускорителях
- Управление перемещением данных между памятью ЦПУ и памятью ускорителей
- Управление параллельным вводом-выводом

# Состав DVM-системы

Система состоит из следующих компонент:

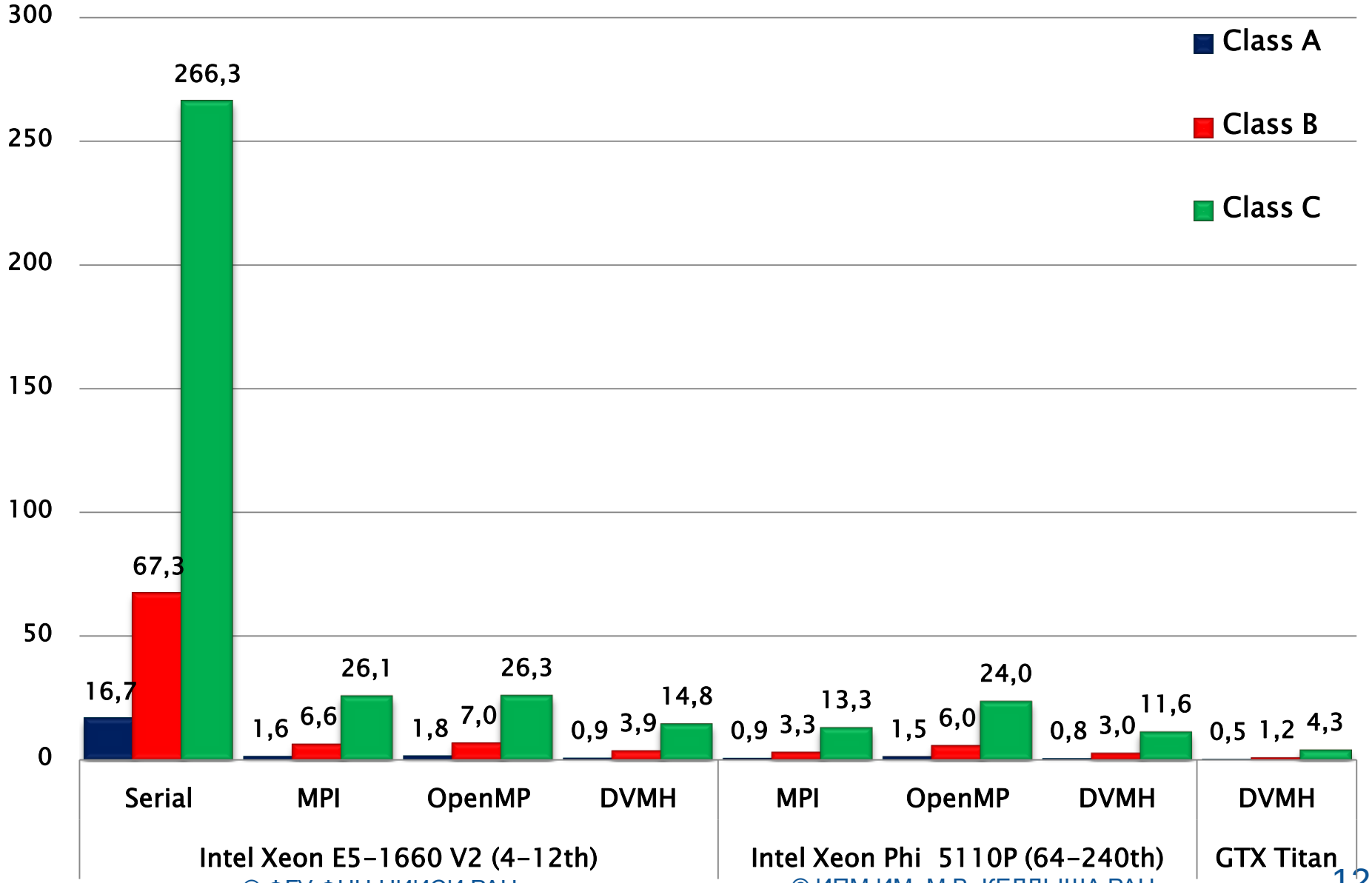
- Компилятор Fortran-DVMH
- Компилятор C-DVMH
- Библиотека поддержки LIB-DVMH
- DVMH-отладчик
- Анализатор производительности DVMH-программ

# Эффективность выполнения тестов NAS

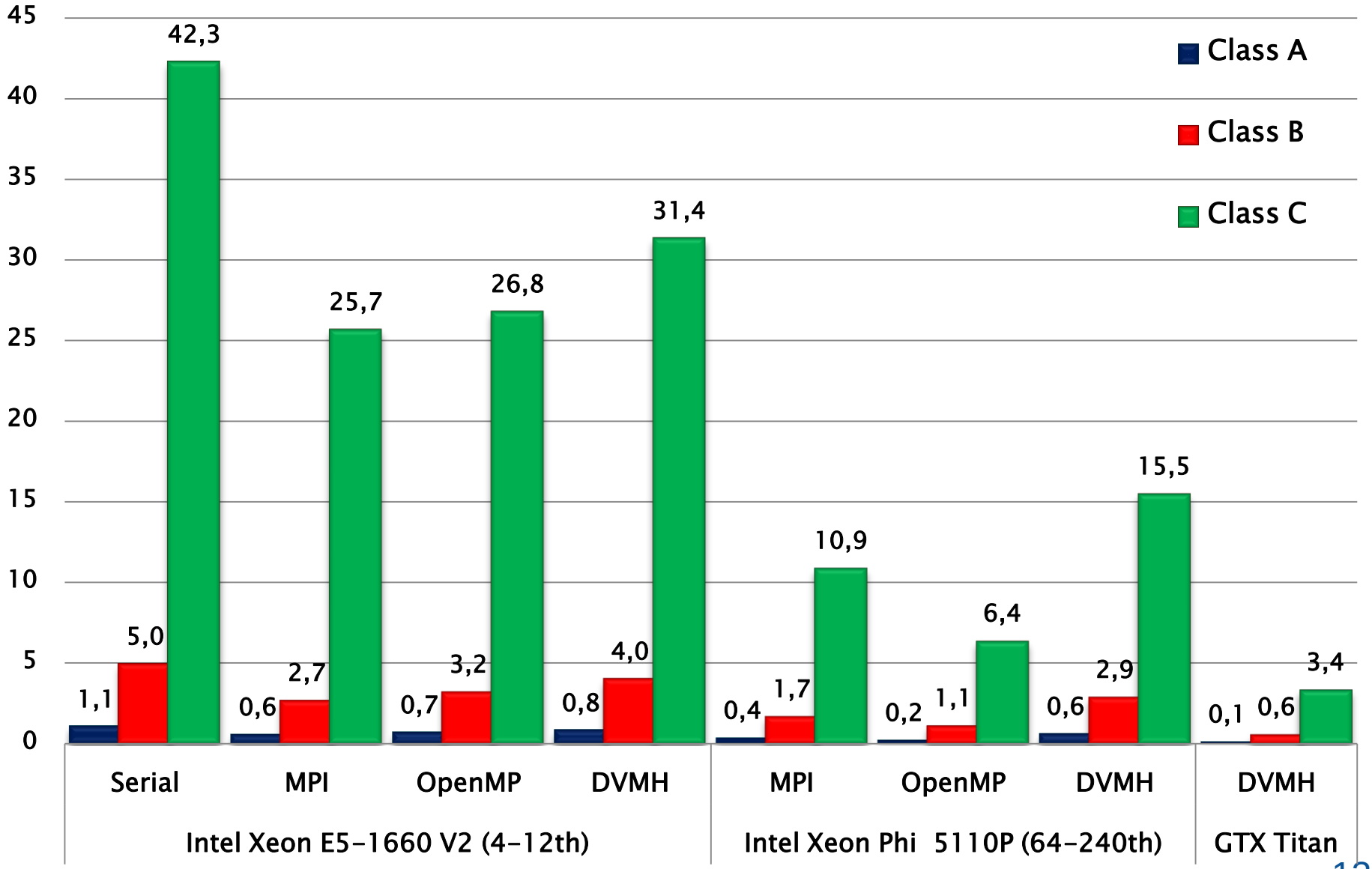
- **EP** - генерация пар случайных чисел Гаусса
- **MG** - приближенное решение трехмерного уравнения Пуассона. Метод MultiGrid
- **BT** - 3D Навье-Стокс, блочная трехдиагональная схема. Метод переменных направлений
- **LU** - 3D Навье-Стокс. Метод последовательной верхней релаксации
- **SP** - 3D Навье-Стокс. Скалярная пятидиагональная схема. Beam-Warning approximate factorization

Подробные результаты исследования эффективности доступны на сайте http://www.dvm-system.org

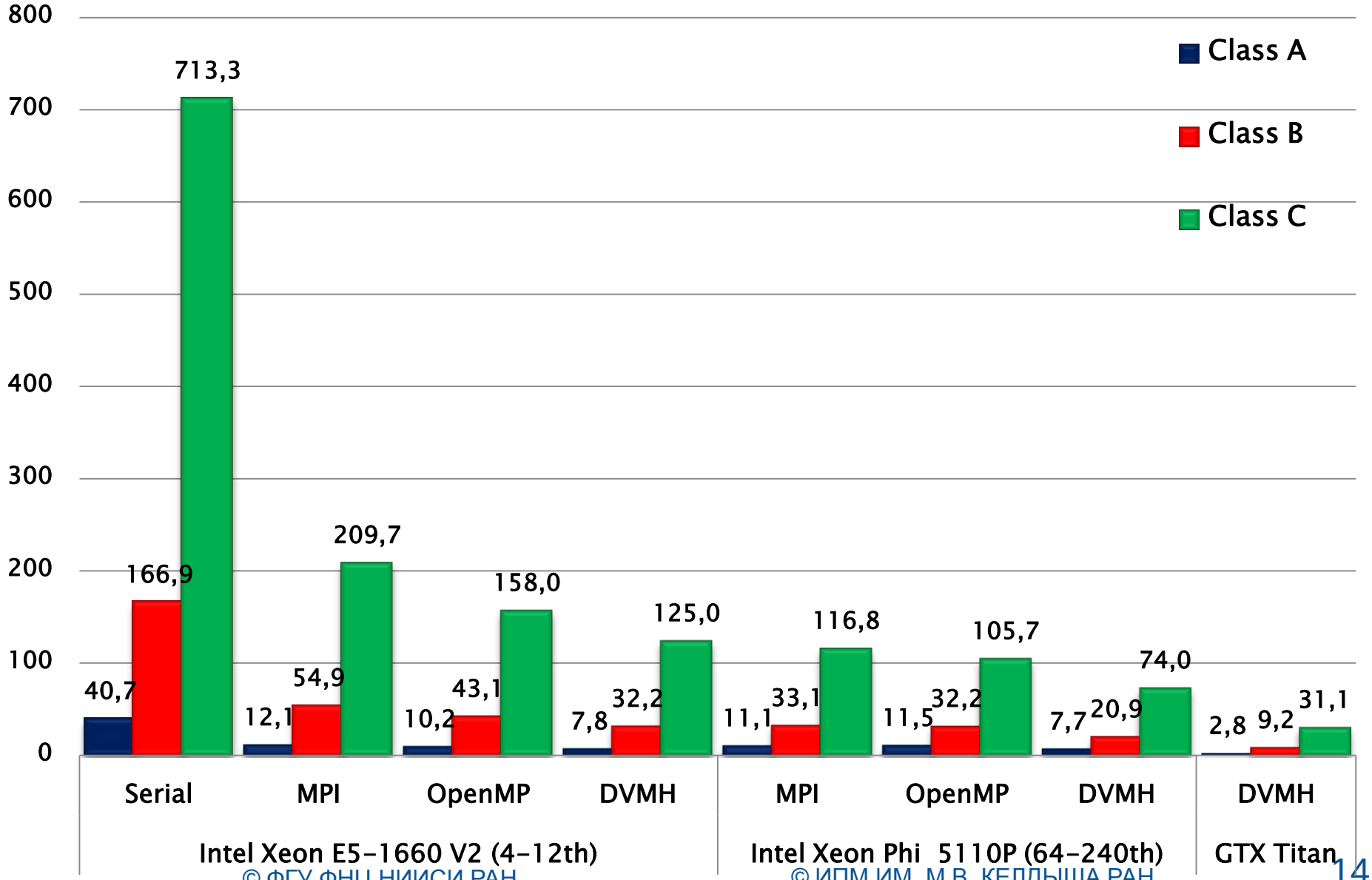
# Времена выполнения теста EP, в секундах



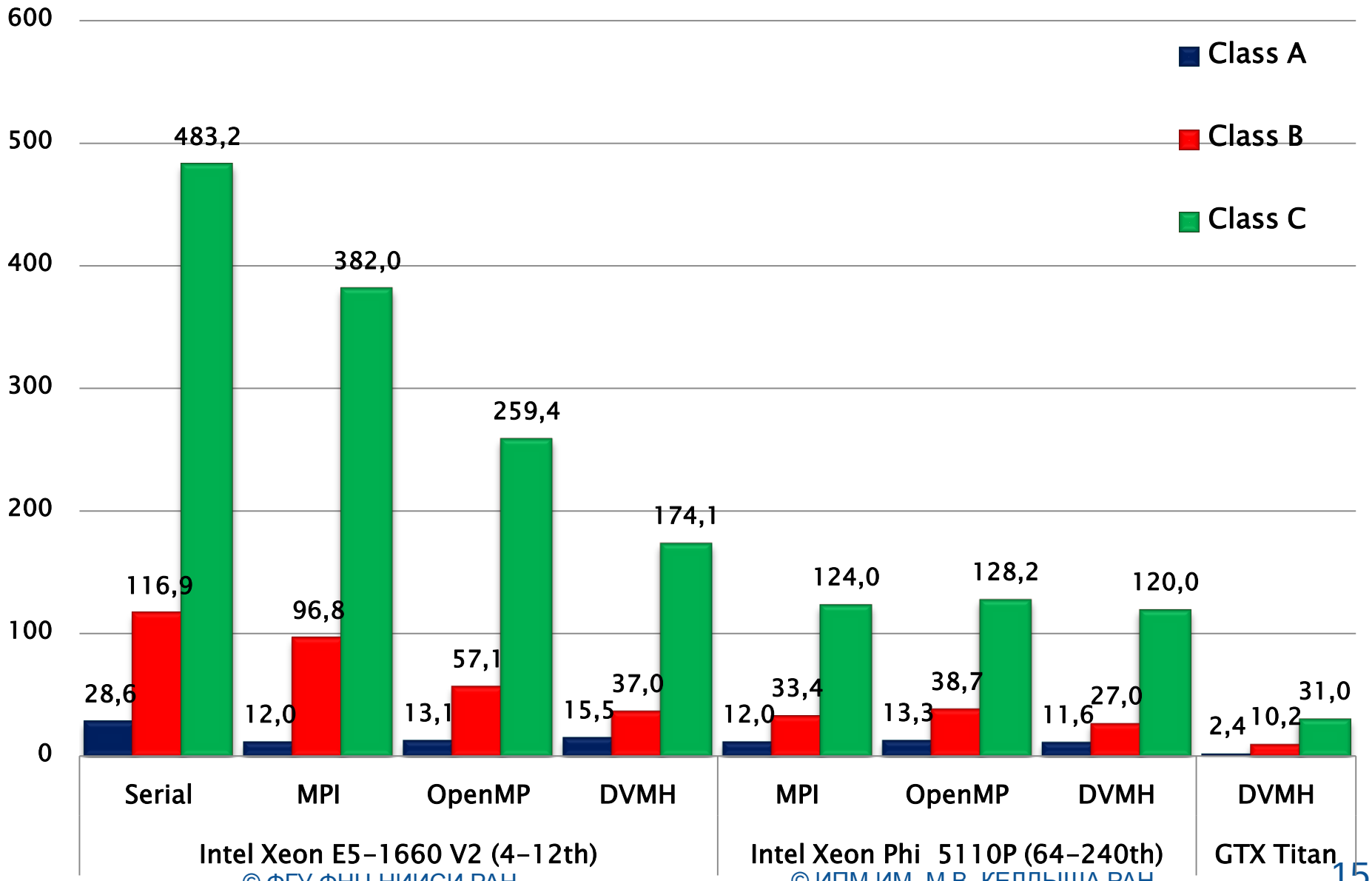
# Времена выполнения теста MG, в секундах



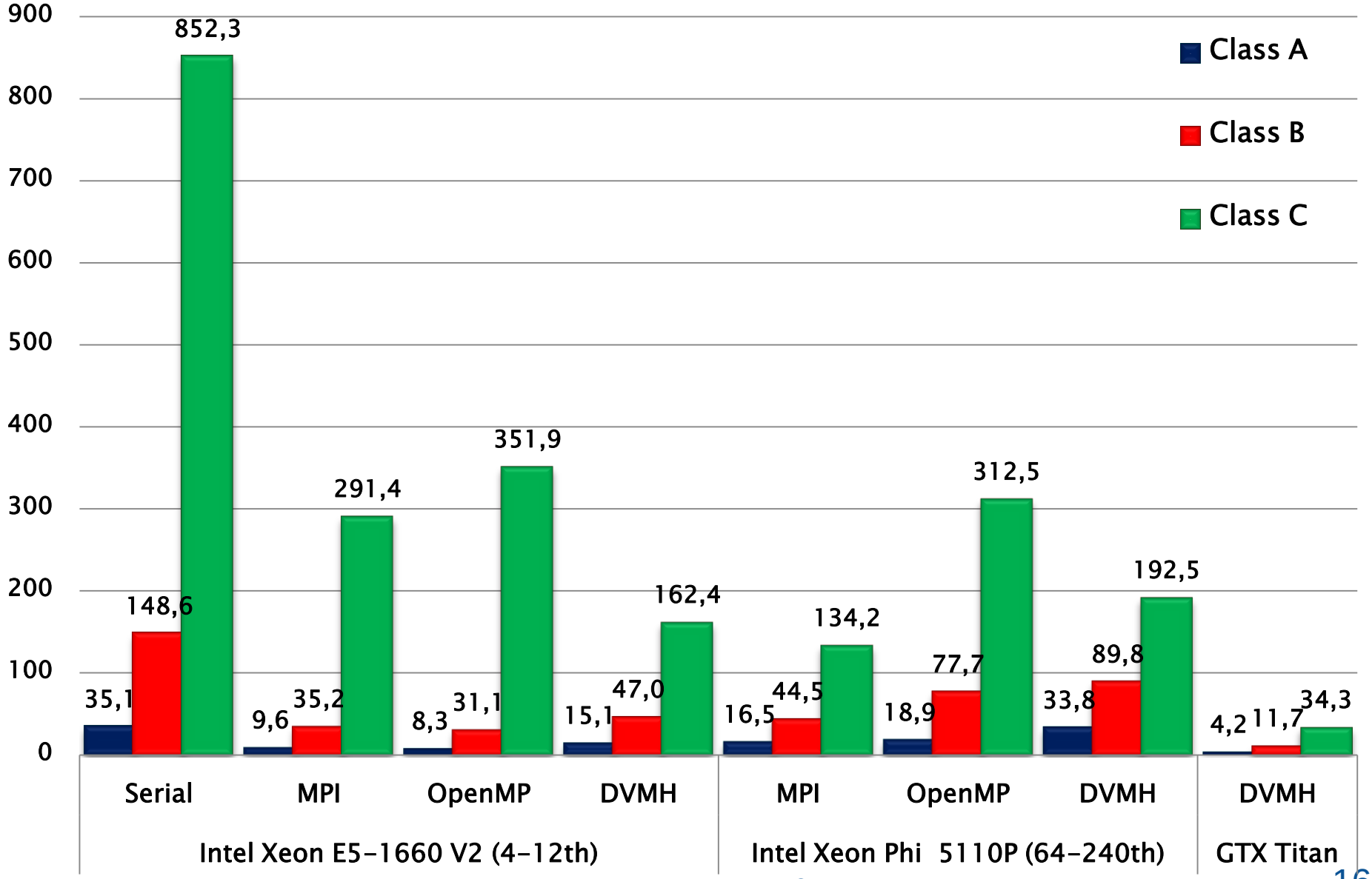
# Времена выполнения теста BT, в секундах



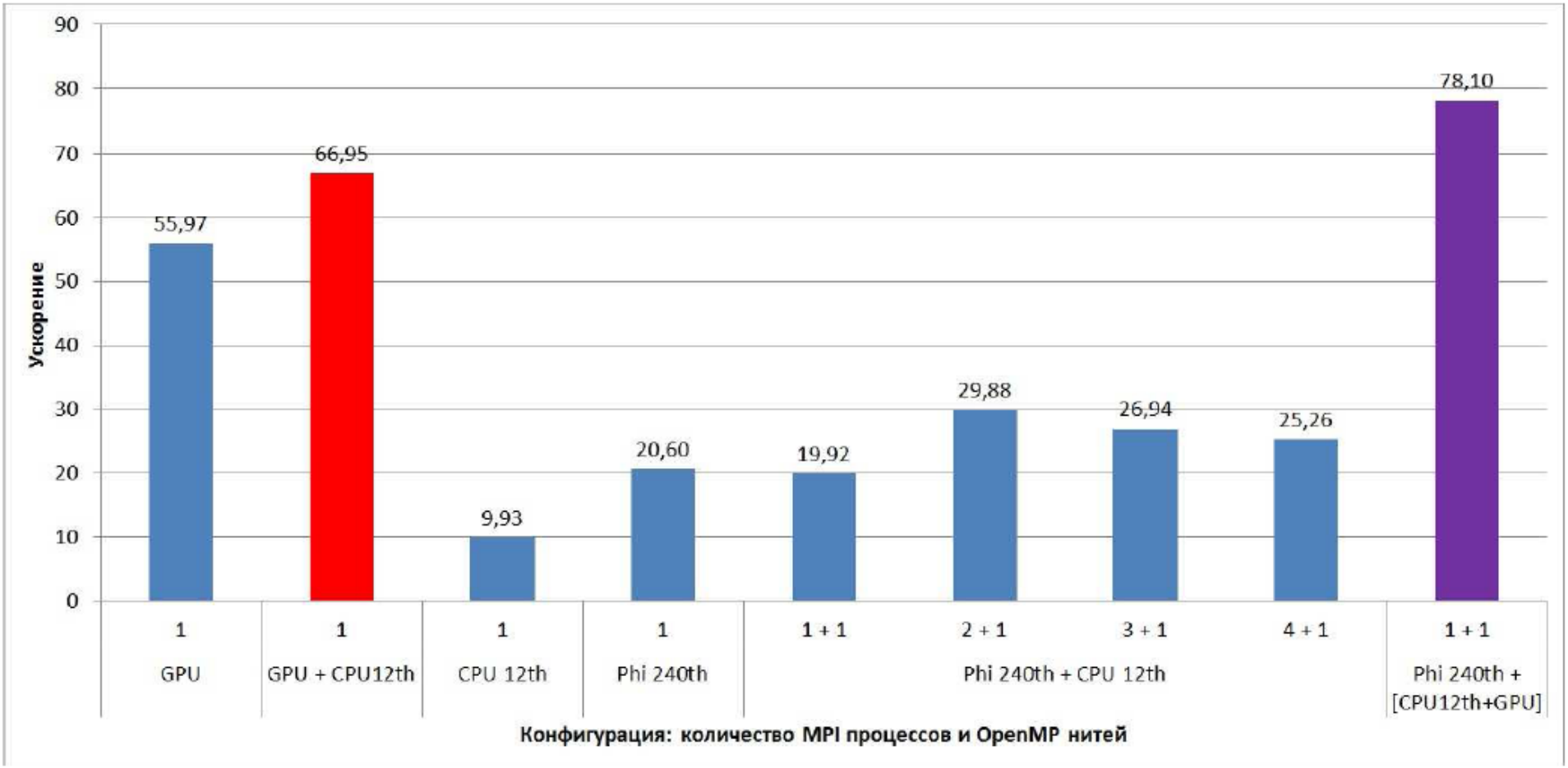
# Времена выполнения теста SP, в секундах



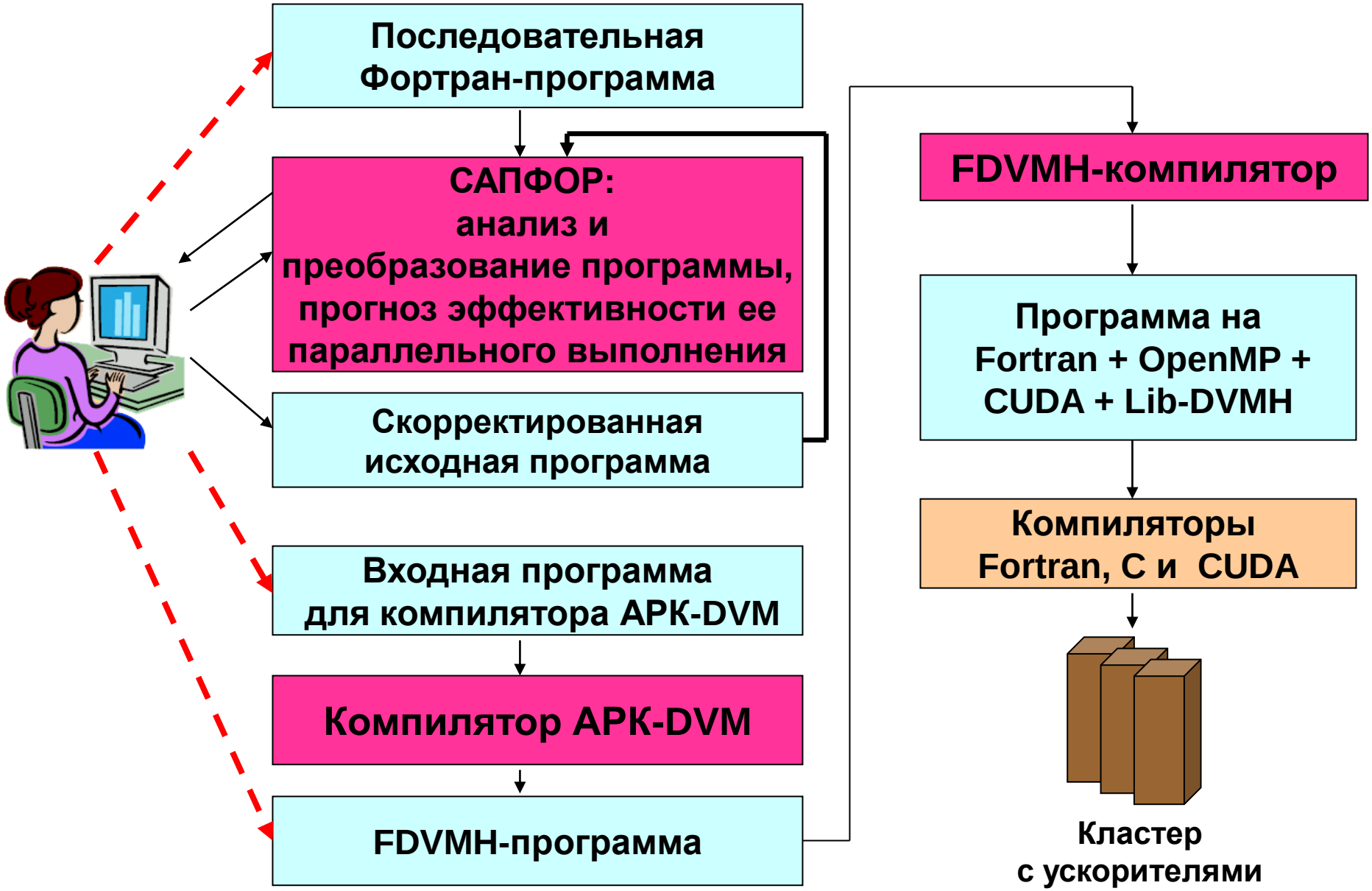
# Времена выполнения теста LU, в секундах



# Выполнение теста EP при одновременном использовании ЦПУ, ГПУ и сопроцессора Xeon PHI



# Схема использования DVMH-модели



# Система САПОР.
# Причины отказа в распараллеливании

Visualizer - [cav_1_sep.for]

File Project View Tools Panels Window Help

Files and PU
cav_1_sep.for
cavity

```
do j = 1,ny
 hy4 = hy(j+1)+2.d0*hy(j)+hy(j-1)
 do i = 0,nx
 t = gm*p(i,j)/ro(i,j)
        tp = gm*p(i+1,j)/ro(i+1,j)
        tm = (t+tp)*0.5d0
        rm = (ro(i,j) + ro(i+1,j))*0.5d0
        pm = (p(i,j) + p(i+1,j))*0.5d0
        uxm = (ux(i,j) + ux(i+1,j))*0.5d0
```

| Loo... | Progr... | File name | dvm_xp_0 Decisions | Writ |
|---|---|---|---|---|
| =,, | cavity | cav_1_sep.for (58) | наличие операторов ввода-вывода | |
| i=1,nx | cavity | cav_1_sep.for (71) | данные для записи витка ro(i,0) и ro(i,ny1) находятся на разных процессорах | |
| j=1,ny | cavity | cav_1_sep.for (85) | данные для записи витка ro(0,j) и ro(nx1,j) находятся на разных процессорах | |
| j=1,ny | cavity | cav_1_sep.for (147) | данные для записи витка tmp1(:,j) находятся на разных процессорах | |
| i=1,nx | cavity | cav_1_sep.for (206) | данные для записи витка tmp1(i,:) находятся на разных процессорах | |
| j=1,ny | cavity | cav_1_sep.for (291) | данные для записи витка tmp1(:,j) находятся на разных процессорах | |
| i=1,nx | cavity | cav_1_sep.for (344) | данные для записи витка tmp1(i,:) находятся на разных процессорах | |
| i=1,nx | cavity | cav_1_sep.for (454) | данные для записи витка ro(i,0) и ro(i,ny1) находятся на разных процессорах | |
| j=1,ny | cavity | cav_1_sep.for (468) | данные для записи витка ro(0,j) и ro(nx1,j) находятся на разных процессорах | |
| j=1,ny1 | cavity | cav_1_sep.for (527) | наличие операторов ввода-вывода | |
| i=1,nx1 | cavity | cav_1_sep.for (529) | наличие операторов ввода-вывода | |

Output | Decisions | Variables | Characteristics

Ready | Ln 147

# Система САПОР. Прогнозирование параллельного выполнения

Visualizer - [cav_1_sep.for]

File Project View Tools Panels Window Help

Files and PU
- cav_1_sep.for
  - cavity

Pro Cal PU

```
do j = 1,ny
 hy4 = hy(j+1)+2.d0*hy(j)+hy(j-1)
 do i = 0,nx
 t = gm*p(i,j)/ro(i,j)
        tp = gm*p(i+1,j)/ro(i+1,j)
        tm = (t+tp)*0.5d0
        rm = (ro(i,j) + ro(i+1,j))*0.5d0
        pm = (p(i,j) + p(i+1,j))*0.5d0
        uxm = (ux(i,j) + ux(i+1,j))*0.5d0
```

| Loops ... | Program ... | File name | Level ... | Sequential time ... | Parallel time ... | Speed up... | Communication... | Parallel ... |
|---|---|---|---|---|---|---|---|---|
| =,, | cavity | cav_1_sep.for (58) | 1.4 | 600.643616 | 609.380737 | 0.985662 | 2967.039551 | no |
| j=1,ny | cavity | cav_1_sep.for (147) | 1.4.5 | 116.019997 | 195.772003 | 0.592628 | 1065.375977 | no |
| i=0,nx | cavity | cav_1_sep.for (149) | 1.4.5.1 | 116.000000 | 195.751999 | 0.592587 | 1065.375977 | yes |
| i=1,nx | cavity | cav_1_sep.for (130) | 1.4.4 | 20.000000 | 187.844864 | 0.106471 | 755.202515 | yes |
| j=1,ny | cavity | cav_1_sep.for (291) | 1.4.11 | 108.020004 | 151.647995 | 0.712307 | 514.943970 | no |
| i=0,nx | cavity | cav_1_sep.for (293) | 1.4.11.1 | 108.000008 | 151.628006 | 0.712270 | 514.943970 | yes |
| i=0,nx1 | cavity | cav_1_sep.for (60) | 1.4.1 | 16.160400 | 102.586884 | 0.157529 | 432.970886 | yes |
| i=1,nx | cavity | cav_1_sep.for (206) | 1.4.7 | 116.019997 | 61.447998 | 1.888101 | 184.384003 | no |
| j=0,ny | cavity | cav_1_sep.for (208) | 1.4.7.1 | 116.000000 | 61.427998 | 1.888390 | 184.384003 | yes |
| i=1,nx | cavity | cav_1_sep.for (344) | 1.4.13 | 108.020004 | 31.560001 | 3.422687 | 0.000000 | no |
| j=0,ny | cavity | cav_1_sep.for (346) | 1.4.13.1 | 108.000008 | 31.540001 | 3.424223 | 0.000000 | yes |
| i=1,nx | cavity | cav_1_sep.for (410) | 1.4.16 | 44.000000 | 6.790580 | 6.479564 | 9.687840 | yes |
| i=1,nx | cavity | cav_1_sep.for (275) | 1.4.10 | 24.000000 | 3.110600 | 7.715553 | 0.000000 | yes |

Output | Decisions | Variables | Characteristics

Ready | Ln 149

# Система САПОР. Инкрементальное распараллеливание

- Возможность распараллелить не всю программу, а ее времяемкие фрагменты. Позволяет найти лучшие схемы распараллеливания времяемких фрагментов
- Постепенное добавление распараллеливаемых фрагментов с сохранением уже найденных решений или без сохранения этих решений
- Возможность ручного распараллеливания некоторых фрагментов и учета принятых программистом решений при распараллеливании других фрагментов

# Композиционная модель

- Разработка в режиме истощения залежей летучей нефти и газоконденсатных залежей, где состав и свойства фаз сильно меняются с изменением давления
- Закачка неравновесного газа (сухого или обогащенного) для мобилизации нефти за счет испарения в более подвижную газовую фазу или за счет конденсации в жидкую фазу промежуточных компонентов и достижения смесимости
- Закачка в нефтяную залежь углекислого газа для мобилизации нефти за счет испарения легких фракций, уменьшения вязкости и набухания нефти

© ФГУ ФНЦ НИИСИ РАН
© ИПМ ИМ. М.В. КЕЛДЫША РАН

# Формулировка композиционной модели с использованием уравнения состояния

$$\frac{\partial}{\partial t}\left[\phi\left(\xi_o S_o x_i + \xi_g S_g y_i\right)\right] + div\left(\xi_o x_i \vec{W}_o + \xi_g y_i \vec{W}_g\right) - q_i = 0, \quad i = 1,2,\ldots,N_c$$

$$\frac{\partial}{\partial t}\left[\phi \xi_w S_w\right] + div\left(\xi_w \vec{W}_w\right) - q_w = 0$$

$$\vec{W}_\alpha = -k k_{r\alpha} \mu_\alpha^{-1}\left(gradP_\alpha + \rho_\alpha g\ gradD\right), \qquad \alpha = o, g, w$$

$$f_i^o - f_i^g = 0, \qquad i = 1,2,\ldots,N_c$$

$$\sum_{i=1}^{N_c} x_i = 1, \qquad \sum_{i=1}^{N_c} y_i = 1$$

$$S_o + S_g + S_w = 1,$$

$$P_o - P_w = P_{cwo}, \quad P_g - P_o = P_{cog}, \quad P_g - P_w = P_{cwo} + P_{cog}$$

# Обозначения

$N_c$ - общее число компонентов в нефтяной и газовой фазах,

$k$ и $\phi$ - соответственно абсолютная проницаемость и пористость пласта,

$t$ - время, $P_\alpha$ - давление в фазе α, $S_\alpha$ - насыщенность фазой α,

$x_i$ и $y_i$ - мольная концентрация $i$-ого компонента соответственно в нефтяной и газовой фазе,

$\xi_\alpha$, $\mu_\alpha$, $\rho_\alpha$ - мольная плотность, динамическая вязкость, массовая плотность фазы $\alpha$,

$P_{cwo}$, $P_{cog}$ - капиллярные давления в системе нефть-вода и в системе нефть-газ соответственно,

$k_{r\alpha}$ - относительная проницаемость для фазы α,

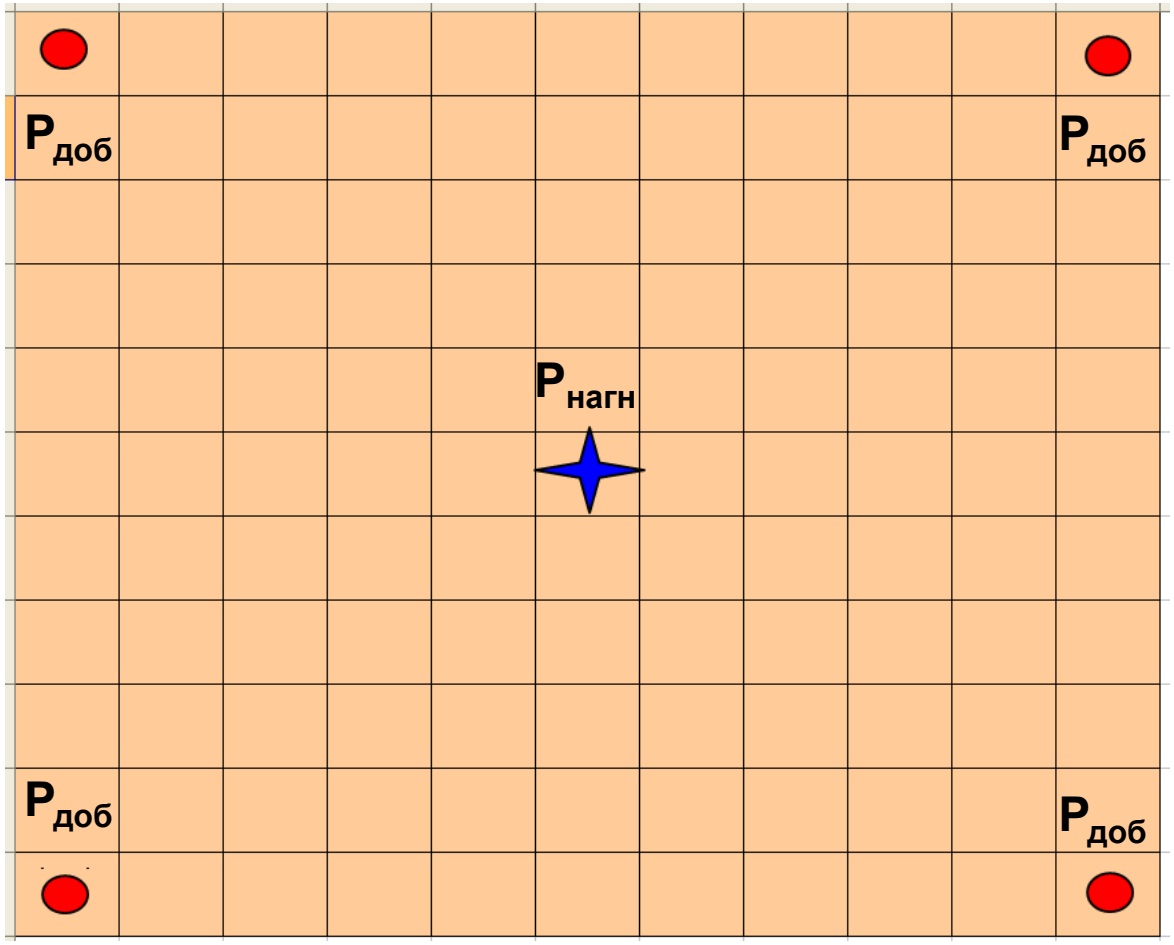
$q_i$, $q_w$ - плотность источников и стоков для компонентов и воды,

$D(x,y,z)$ - превышение точки пласта над некоторой горизонтальной плоскостью,

$g$ - ускорение силы тяжести,

$f_i^o$ и $f_i^g$ - летучести компонентов в нефтяной и газовой фазах

# Примеры с закачкой в пласт сухого и жирного газа (элемент 5-точечной системы разработки)




$P_{нач}$ = 244 атм

$P_{доб}$ = 244 атм

$P_{нагн}$ = 400 атм

**Схема неявная по давлению и явная по концентрации компонентов**

# Распараллеливание программы Композит

- Перенос программы в ОС Linux. Отладка программы
- Определение времяемких фрагментов программы
- Распараллеливание времяемких фрагментов программы
- Добавление распараллеливаемых фрагментов с сохранением уже найденных решений

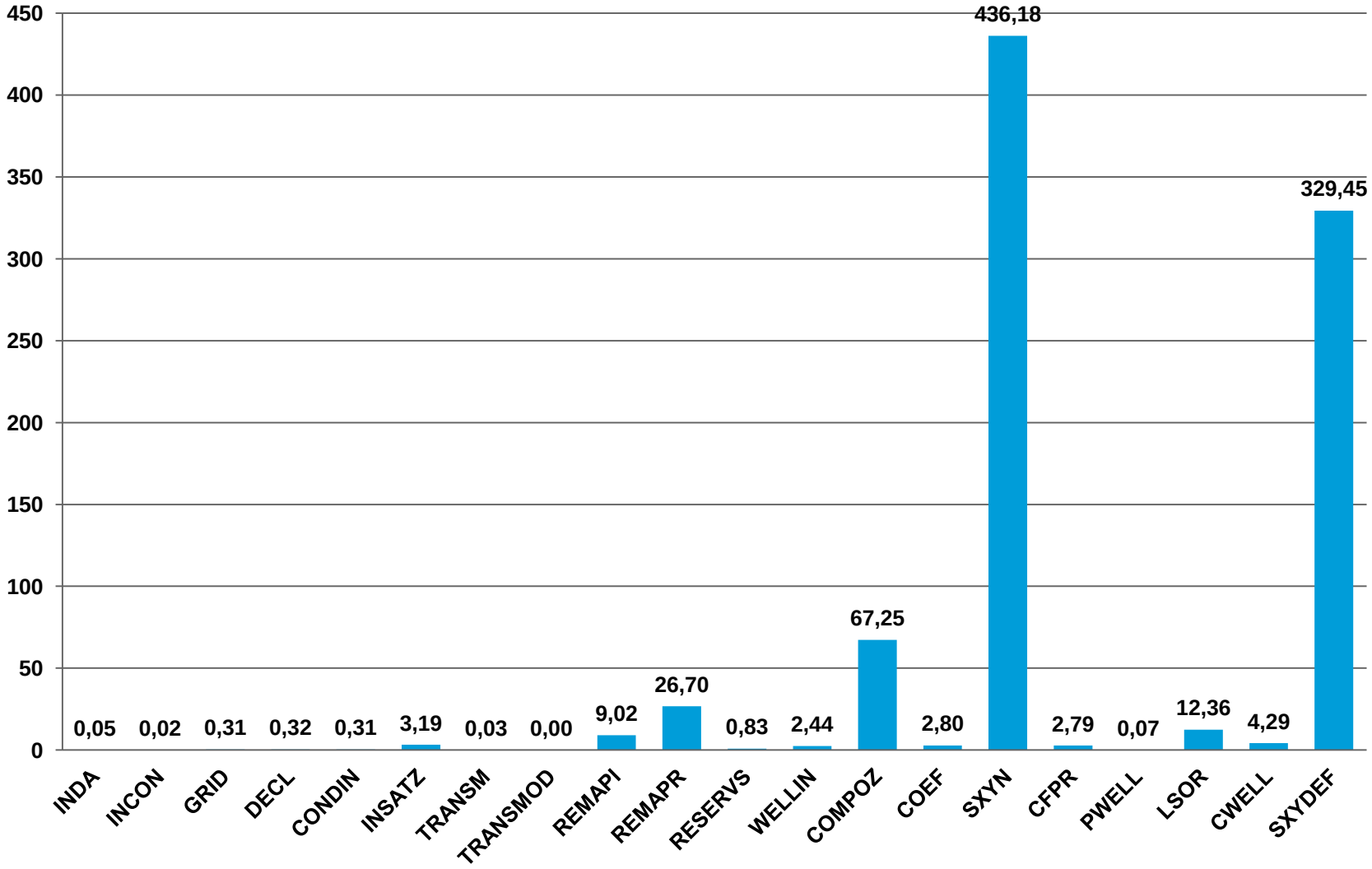
# Сравнительная отладка программы

- Для поиска ошибок используется метод накопления и сравнения результатов вычислений, который позволяет определить место в программе и момент, когда появляются расхождения
- При трассировке вычислений выполняется сбор информации обо всех чтениях и модификациях переменных, о начале выполнения каждого витка цикла
- Степенью подробности и объемом трассировки можно управлять при конвертации программы опциями DVMH-конверторов, а также при выполнении программы
- Существует возможность управлять точностью при сравнении результатов
- Генерируются протоколы с различиями, обнаруженными в процессе сравнения и скачками в значениях переменных и элементах массивов

# Определение времяемких фрагментов программы

```
INTERVAL ( NLINE=463 SOURCE=COMPOZ.FOR ) LEVEL=3 USER EXE_COUNT=564
-- The main characteristics ---
Parallelization efficiency      1.0000
Execution time                  67.2498
Processors                      1
Total time                      67.2498
Productive time                 67.2498 ( CPU= 67.2063  Sys= 0.0435 I/O= 0.0000 )
--- The comparative characteristics ---
                        Tmin N proc      Tmax N proc      Tmid
Execution time          67.2498    1     67.2498    1     67.2498
User CPU time           67.2063    1     67.2063    1     67.2063
Sys. CPU time           0.0435     1     0.0435     1     0.0435
Processors              1    1         1    1         1
--- The execution characteristics ---
                        1
Execution time          67.2498
User CPU time           67.2063
Sys. CPU time           0.0435
Processors              1
```

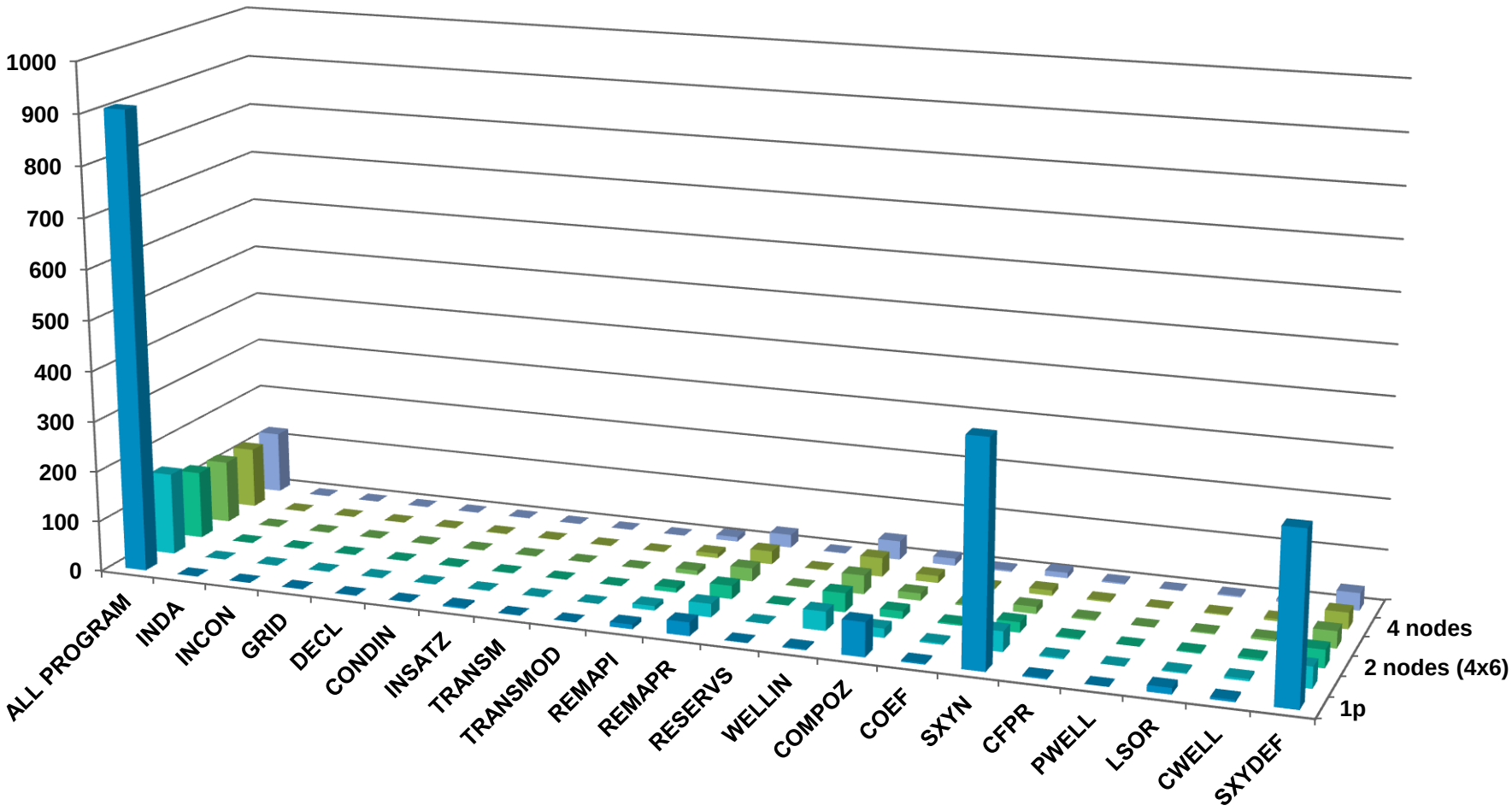
# Времена выполнения различных процедур программы Композит



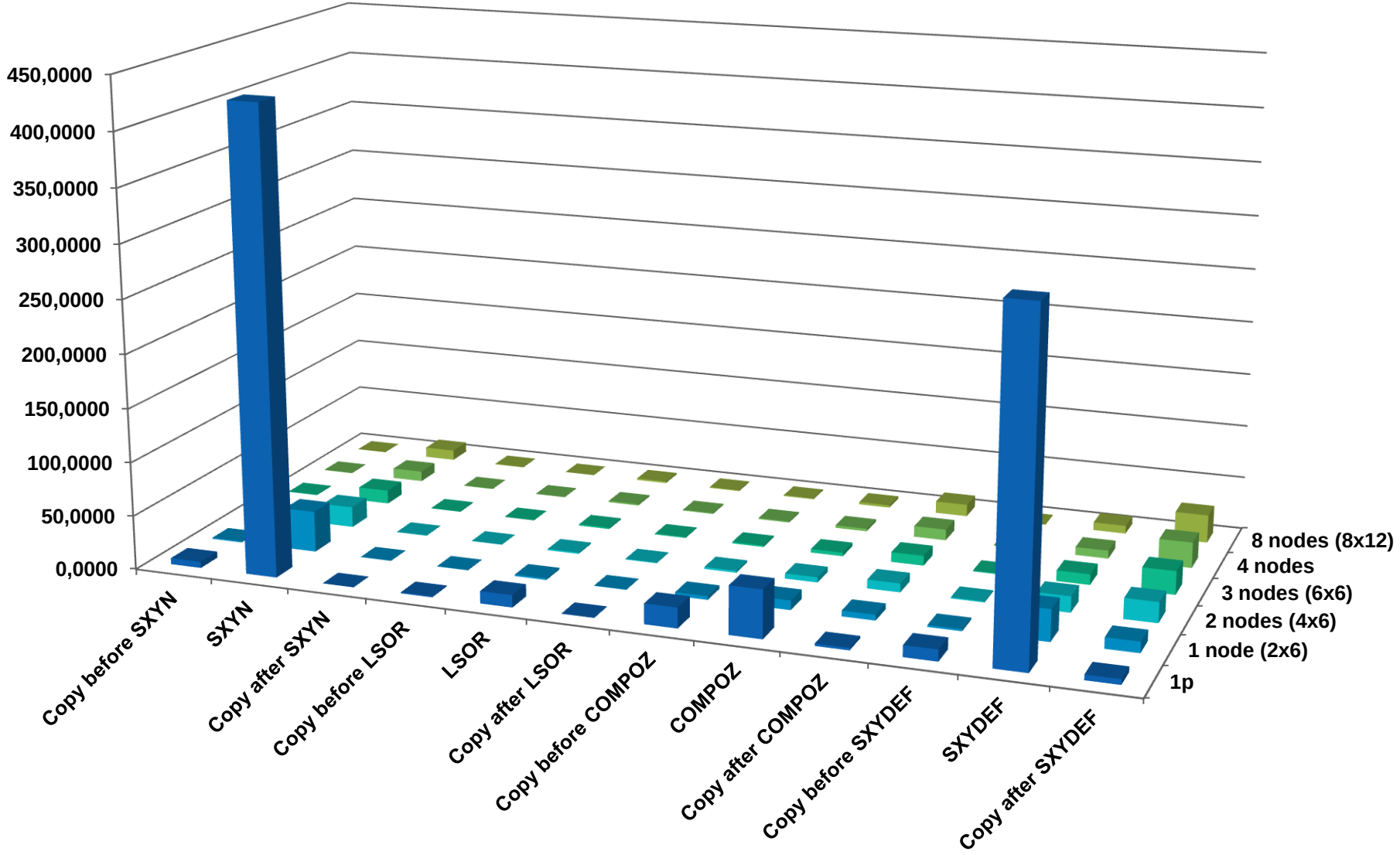
# Распараллеливание времяемких фрагментов программы

- Анализ операторов (циклов) программы на предмет их возможного параллельного выполнения
- Определение наилучшего варианта распределения данных, требуемых для выполнения этих операторов
- Организация параллельных вычислений (расстановка DVMH-директив)
- Если распределяемые данные используются в других фрагментах программы, то требуется завести массивы-копии
- Организация копирования информации между исходными массивами и их копиями

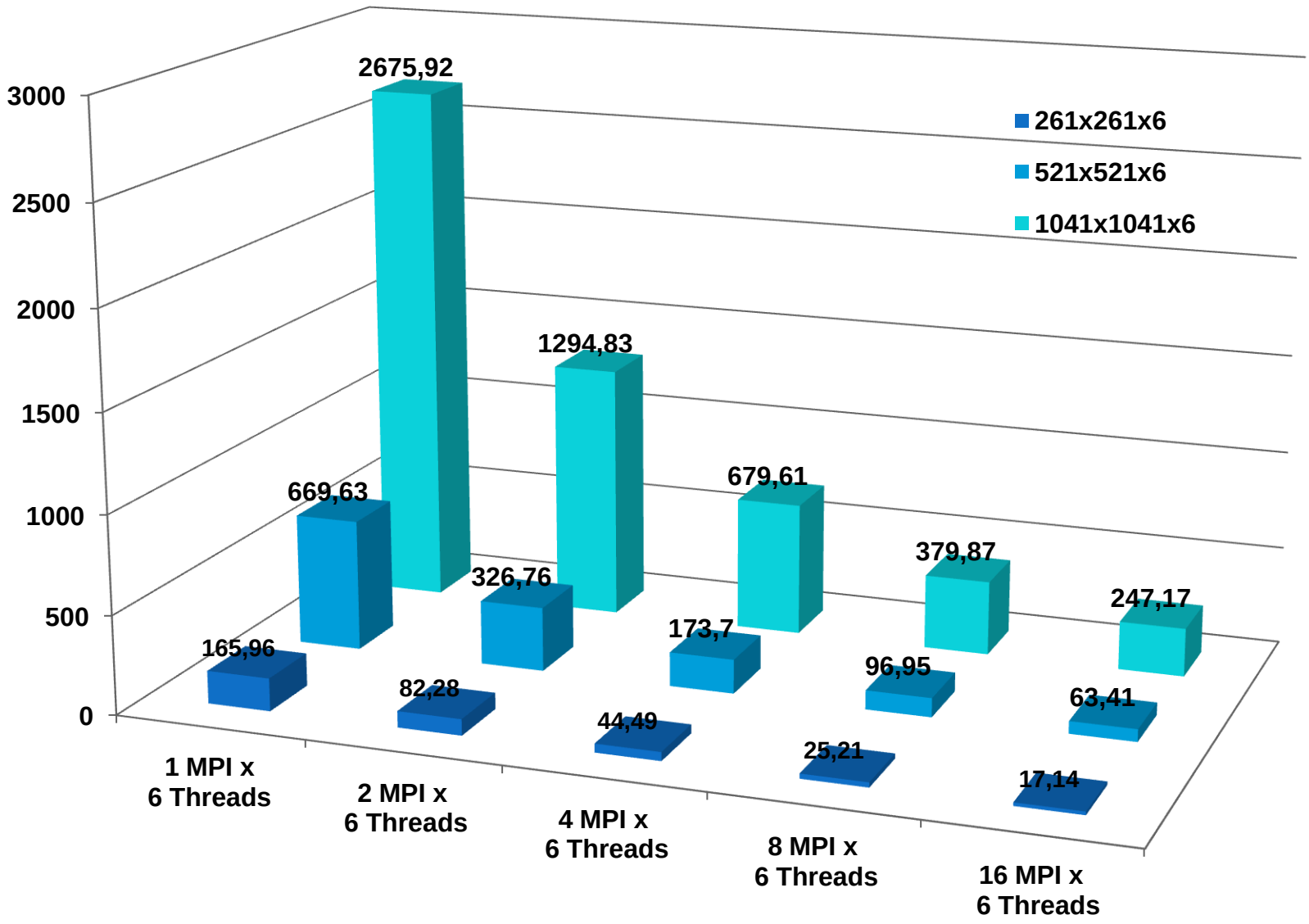
# Времена выполнения частично распараллеленной программы на разном числе узлов K-100



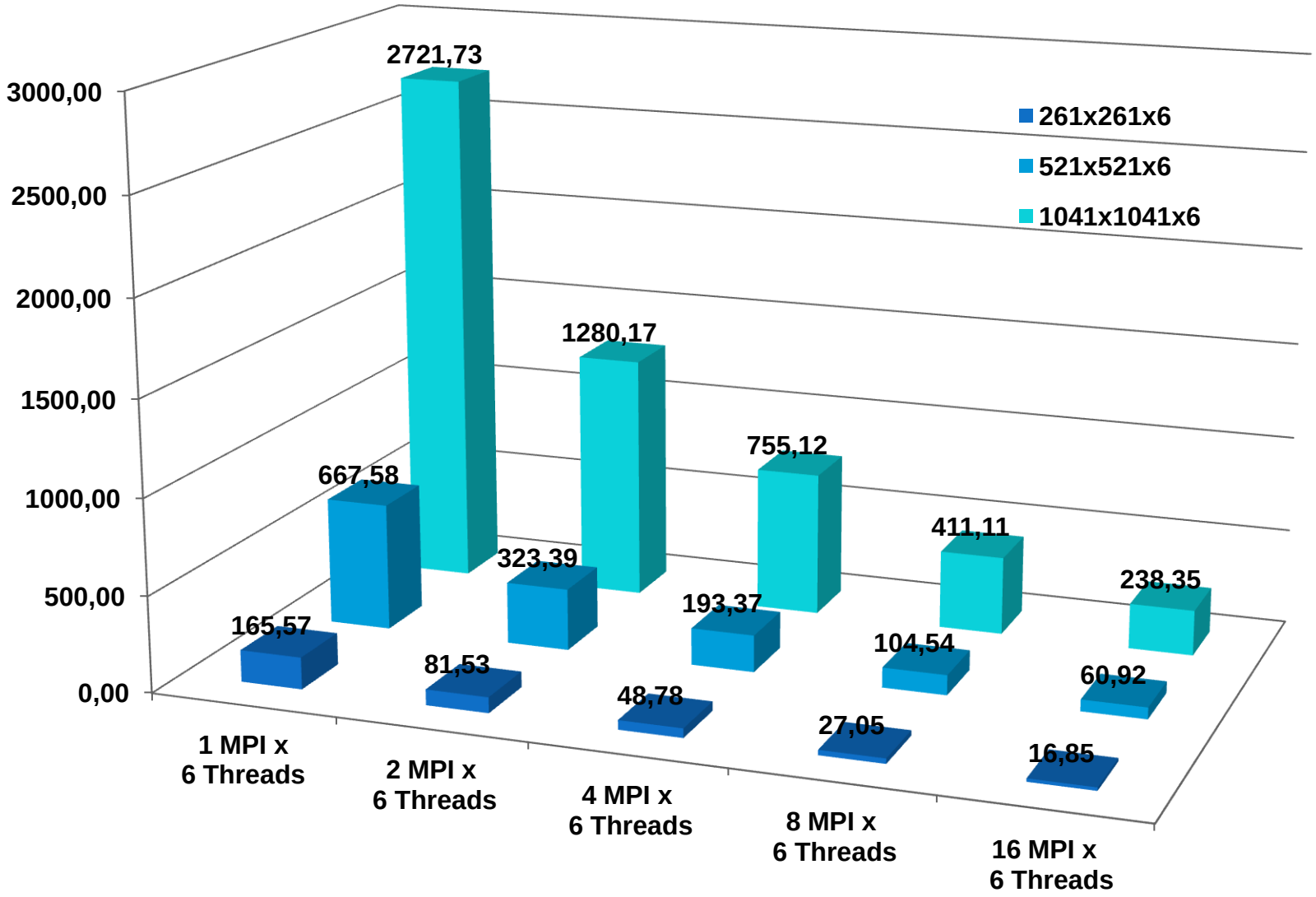
# Накладные расходы на копирование данных между исходными массивами и их копиями



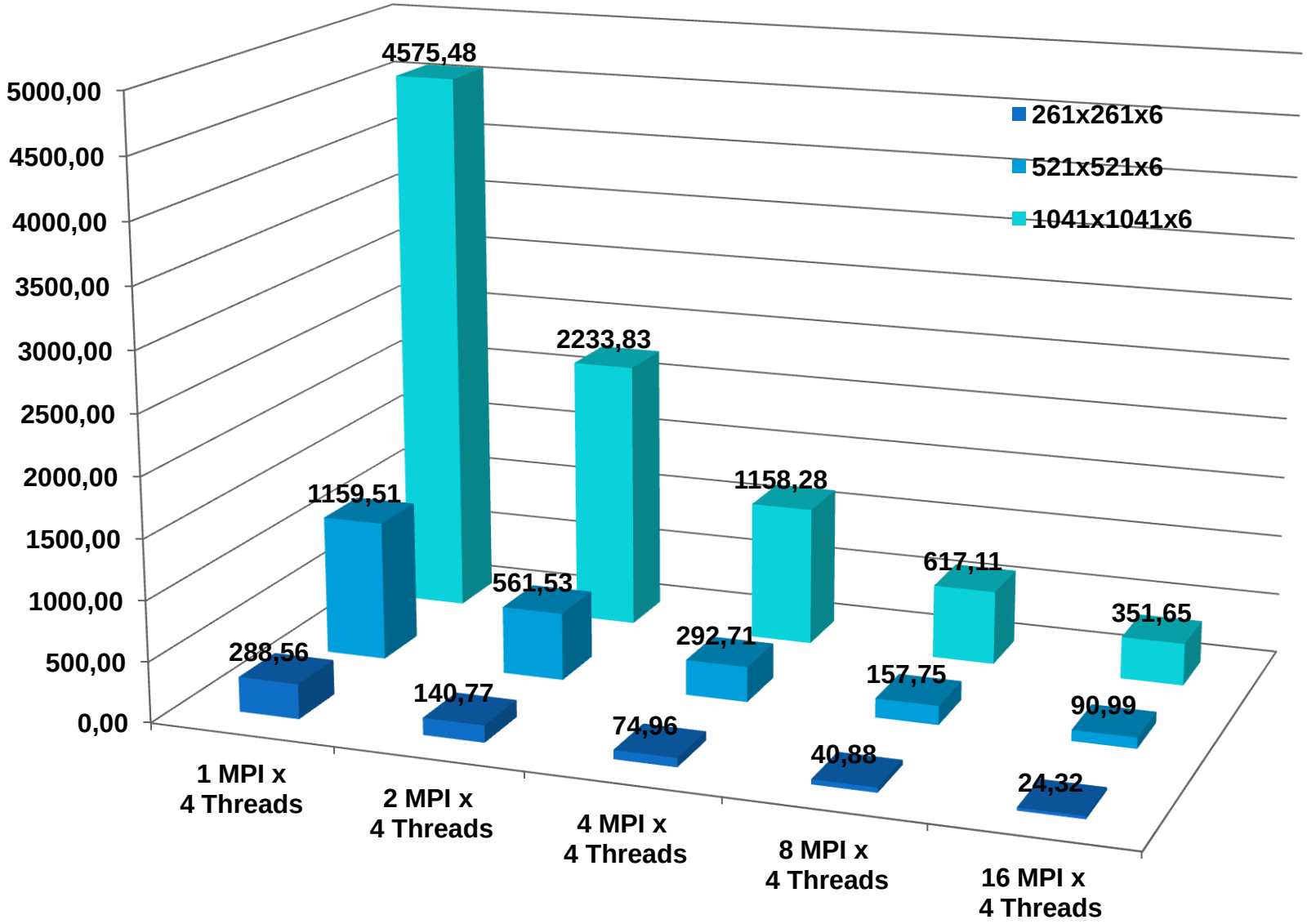
# Времена выполнения 100 итераций программы Композит на суперкомьютере К-100. Сухой газ



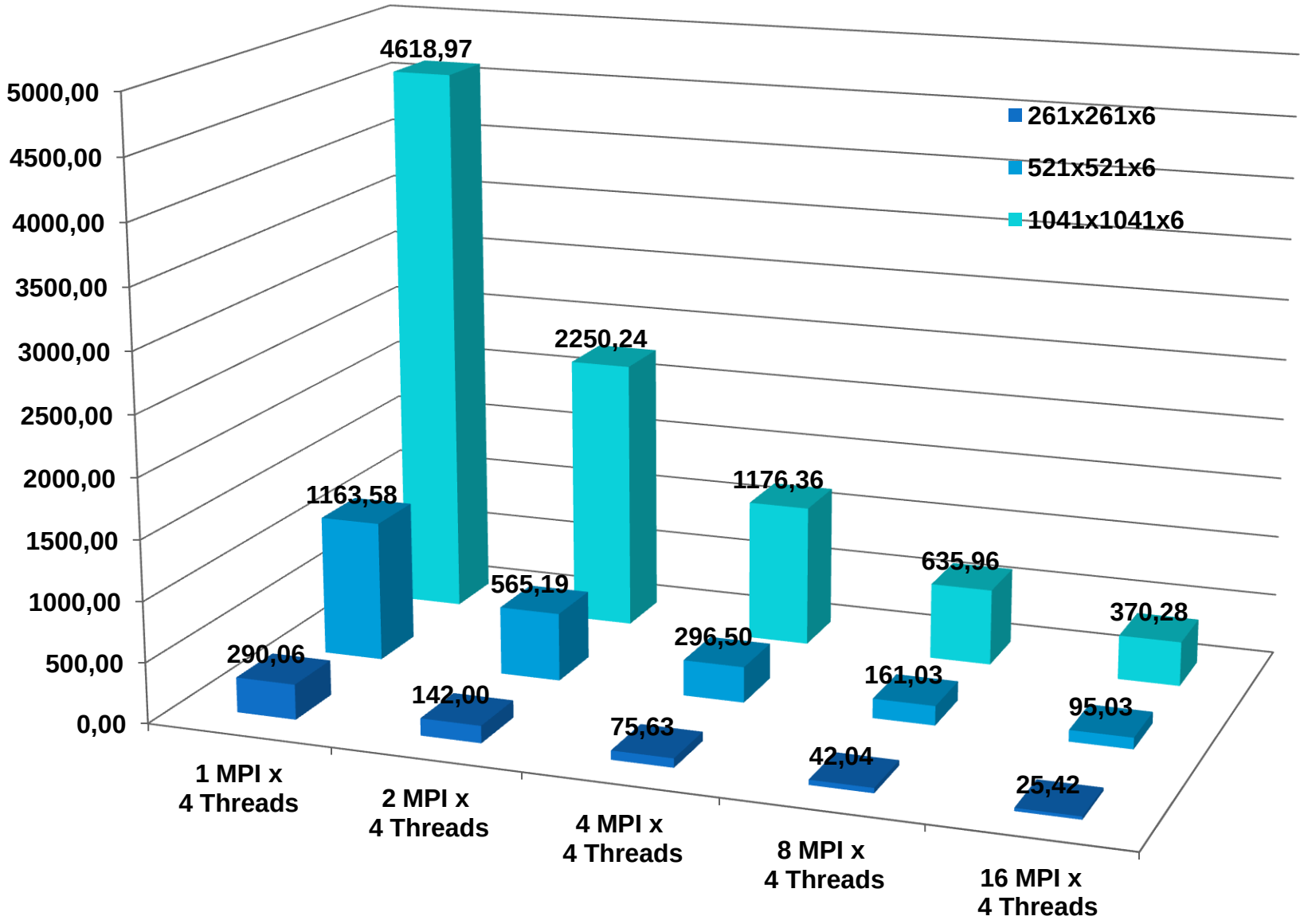
# Времена выполнения 100 итераций программы Композит на суперкомьютере К-100. Жирный газ



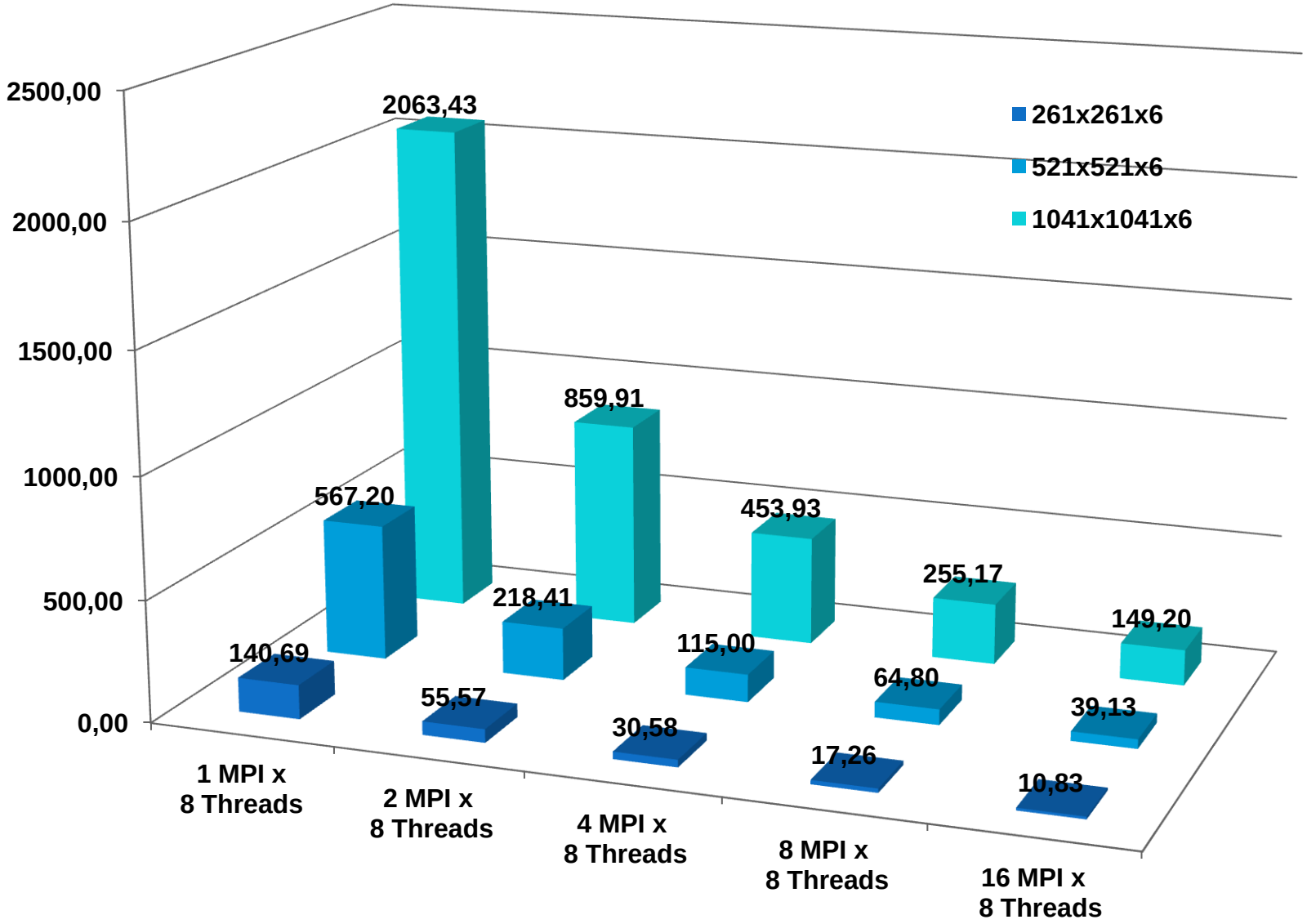
# Времена выполнения 100 итераций программы Композит на суперкомьютере Ломоносов. Сухой газ



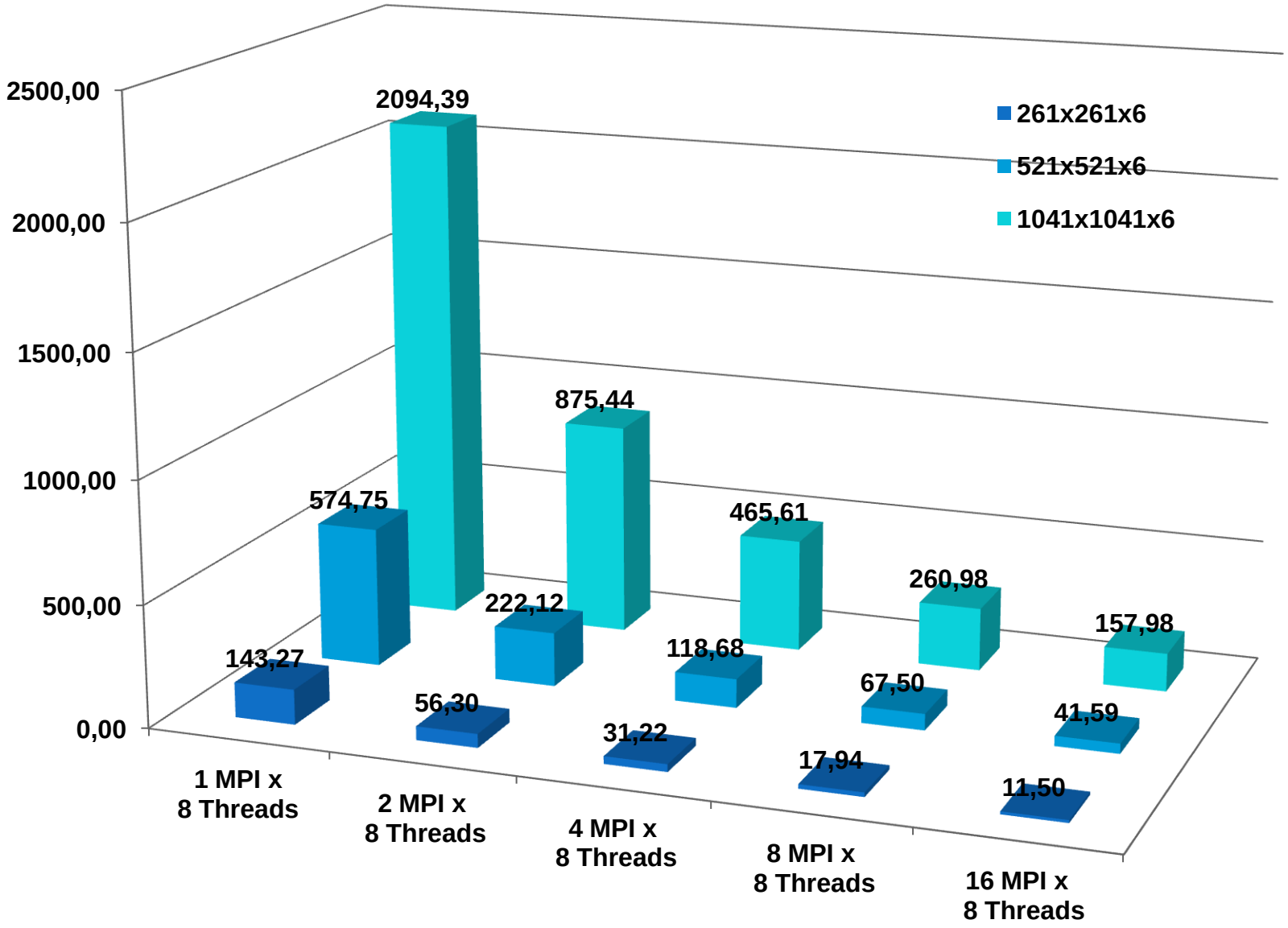
# Времена выполнения 100 итераций программы Композит на суперкомьютере Ломоносов. Жирный газ



# Времена выполнения 100 итераций программы Композит на суперкомьютере MVS-10P. Сухой газ



# Времена выполнения 100 итераций программы Композит на суперкомьютере MVS-10P. Жирный газ



# Выводы

- Разработана параллельная программа для моделирования многокомпонентной фильтрации при разработке месторождений нефти и газа
- Использование DVM-системы и системы САПФОР существенно упростило и ускорило процесс разработки параллельной программы
- Применение инкрементального подхода для распараллеливания позволило получить эффективную параллельную программу
- Опыт, полученный при распаралелливании данной программы, будет использован при реализации следующей версии системы САПФОР

# Вопросы, замечания?

СПАСИБО !

**http://dvm-system.org**

**dvm@keldysh.ru**