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

Создание параллельной версии программы для моделирования многокомпонентной фильтрации при разработке месторождений нефти и газа с использованием DVM-системы

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

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

Саров, 2016

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

План доклада

- Композиционная модель фильтрации (Королев А.В.)
- Автоматизация разработки параллельных программ (Бахтин В.А.)
  - Система DVM
  - Система САПФОР
- Распараллеливание программы Композит (Бахтин В.А.)

Примеры применения композиционной модели

- 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

Формулировка композиционной модели с использованием уравнения состояния

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

$$\frac{\partial}{\partial t}\left[\phi \xi_w S_w\right] + div\left(\xi_w \vec{W}_w\right) - q_w = 0\,, \qquad (2)$$

$$\vec{W}_\alpha = -k k_{r\alpha} \mu_\alpha^{-1}\left(grad P_\alpha + \rho_\alpha g\, grad D\right), \qquad \alpha = o, g, w\,, \qquad (3)$$

$$f_i^o - f_i^g = 0\,, \qquad i = 1,2,\ldots,N_c\,, \qquad (4)$$

$$\sum_{i=1}^{N_c} x_i = 1\,, \qquad \sum_{i=1}^{N_c} y_i = 1\,, \qquad (5)$$

$$S_o + S_g + S_w = 1\,, \qquad (6)$$

$$P_o - P_w = P_{cwo}\,,\ \ P_g - P_o = P_{cog}\,,\ \ P_g - P_w = P_{cwo} + P_{cog}\,, \qquad (7)$$

Обозначения

N_c - общее число компонентов в нефтяной и газовой фазах,

k и ϕ - соответственно абсолютная проницаемость и пористость пласта,

t - время, P_α - давление в фазе α, S_α - насыщенность фазой α,

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

ξ_α, μ_α, ρ_α - мольная плотность, динамическая вязкость, массовая плотность фазы α,

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

Преобразование уравнений

$$\frac{\partial}{\partial t}\left[\phi\left(\xi_w S_w\right)\right] - div\left[\xi_w k k_{rw} \mu_w^{-1}\left(grad(P - P_{cwo}) + \rho_w g\, grad D\right)\right] - q_w = 0. \qquad (8)$$

$$\frac{\partial}{\partial t}\left[\phi\, z_i\left(\xi_o S_o + \xi_g S_g\right)\right] - div\left[\xi_o x_i k k_{ro} \mu_o^{-1}\left(grad P + \rho_o g\, grad D\right) + \right.$$
$$\left. + \xi_g y_i k k_{rg} \mu_g^{-1}\left(grad(P - P_{cgo}) + \rho_g g\, grad D\right)\right] - q_i = 0, \qquad (9)$$

$$z_i = x_i L + y_i V, \quad i = 1,2,\ldots,N_c,$$

$$L = \frac{\xi_o S_o}{\xi_o S_o + \xi_g S_g}, \; V = \frac{\xi_g S_g}{\xi_o S_o + \xi_g S_g}, \qquad \sum_{i=1}^{N_c} z_i = 1 \qquad (10)$$

$$\frac{\partial}{\partial t}\left[\phi\left(\xi_o S_o + \xi_g S_g + \theta \xi_w S_w\right)\right] - div\left[\xi_o k k_{ro} \mu_o^{-1}\left(grad P + \rho_o g\, grad D\right) + \right.$$
$$+ \xi_g k k_{rg} \mu_g^{-1}\left(grad(P - P_{cgo}) + \rho_g g\, grad D\right) + \qquad (11)$$
$$\left. + \theta \xi_w k k_{rw} \mu_w^{-1}\left(grad(P - P_{cwo}) + \rho_w g\, grad D\right)\right] - q_o - q_g - \theta q_w = 0.$$

Фазовые равновесия на основе уравнений состояния

$$Z^3 + E_2 Z^2 + E_1 Z + E_0 = 0\,. \qquad (12)$$

$$\xi_\alpha = \frac{P}{Z_\alpha RT}, \qquad \alpha = o, g. \qquad (13)$$

$$f_i^o(P, T, x_1, x_2, \ldots x_{Nc}) - f_i^g(P, T, y_1, y_2, \ldots y_{Nc}) = 0\,, \qquad i = 1, 2, \ldots, N_c\,. \qquad (14)$$

$$x_i = \frac{z_i}{1 + (K_i - 1)V}, \quad y_i = \frac{K_i z_i}{1 + (K_i - 1)V}, \quad K_i = \frac{y_i}{x_i}, \quad i = 1, 2, \ldots N_c. \qquad (15)$$

$$g(V) = \sum_{i=1}^{N_c} (y_i - x_i) = \sum_{i=1}^{N_c} \frac{z_i (K_i - 1)}{1 + V(K_i - 1)} = 0\,. \qquad (16)$$

$$K_i^{l+1} = K_i^l \frac{f_i^o}{f_i^g}, \qquad \sum_{i=1}^{N_c} \left(\frac{f_i^o}{f_i^g} - 1 \right)^2 < \varepsilon \qquad (17)$$

Итерационный метод Ньютона-Рафсона с минимумом переменных для решения уравнений равновесия

$$\Phi_i = \left(y_2,\ldots,y_{N_c},V\right) = \ln\left[\ f_i^o\left(x_1,\ldots,x_{N_c}\right)\ \right] - \ln\left[\ f_i^g\left(y_1,\ldots,y_{N_c}\right)\ \right] = 0,\ i = 1..N_c. \qquad (22)$$

$$\sum_{j=2}^{N_c}\left(\frac{\partial\Phi_i}{\partial y_j}\right)^k\left(y_j^{k+1} - y_j^k\right) + \left(\frac{\partial\Phi_i}{\partial V}\right)^k\left(V^{k+1} - V^k\right) = -\Phi_i^k,\ i = 1,\ldots,N_c$$

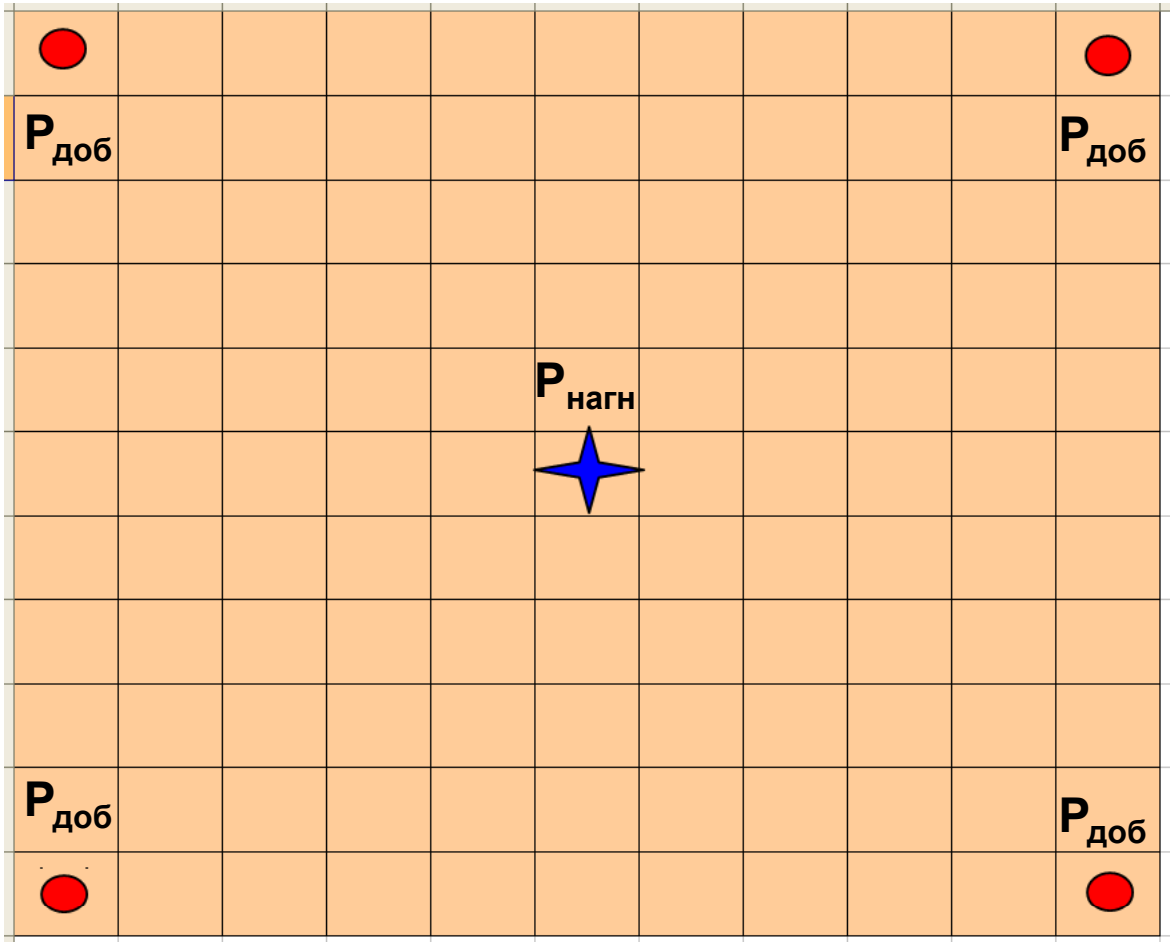
$$\frac{\partial\Phi_i}{\partial y_j} = \sum_{k=1}^{N_c}\frac{\partial\ln\left(f_i^o\right)}{\partial x_k}\frac{\partial x_k}{\partial x_j} - \sum_{k=1}^{N_c}\frac{\partial\ln\left(f_i^g\right)}{\partial y_k}\frac{\partial y_k}{\partial y_j} = \frac{V}{L}\frac{\partial\ln\left(f_i^o\right)}{\partial x_1} - \frac{V}{L}\frac{\partial\ln\left(f_i^o\right)}{\partial x_j} + \frac{\partial\ln\left(f_i^g\right)}{\partial y_1} + \frac{\partial\ln\left(f_i^g\right)}{\partial y_j};$$

$$i = 1,\ldots,N_c;\ j = 2,\ldots,N_c; \qquad (23)$$

$$\frac{\partial\Phi_i}{\partial V} = \sum_{k=1}^{N_c}\frac{\partial\ln\left(f_i^o\right)}{\partial x_k}\frac{\partial x_k}{\partial V} - \sum_{k=1}^{N_c}\frac{\partial\ln\left(f_i^g\right)}{\partial y_k}\frac{\partial y_k}{\partial V} = -\frac{1}{L}\sum_{k=1}^{N_c}\frac{\partial\ln\left(f_i^o\right)}{\partial x_k}\left(y_k - x_k\right);\ i = 1,\ldots,N_c. \qquad (24)$$

$$\left|\Phi_i^k\right| \le \varepsilon, \qquad i = 1,\ldots,N_c$$

Тестовые 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

Составы нефти и газа. Иллюстрация на фазовой диаграмме.

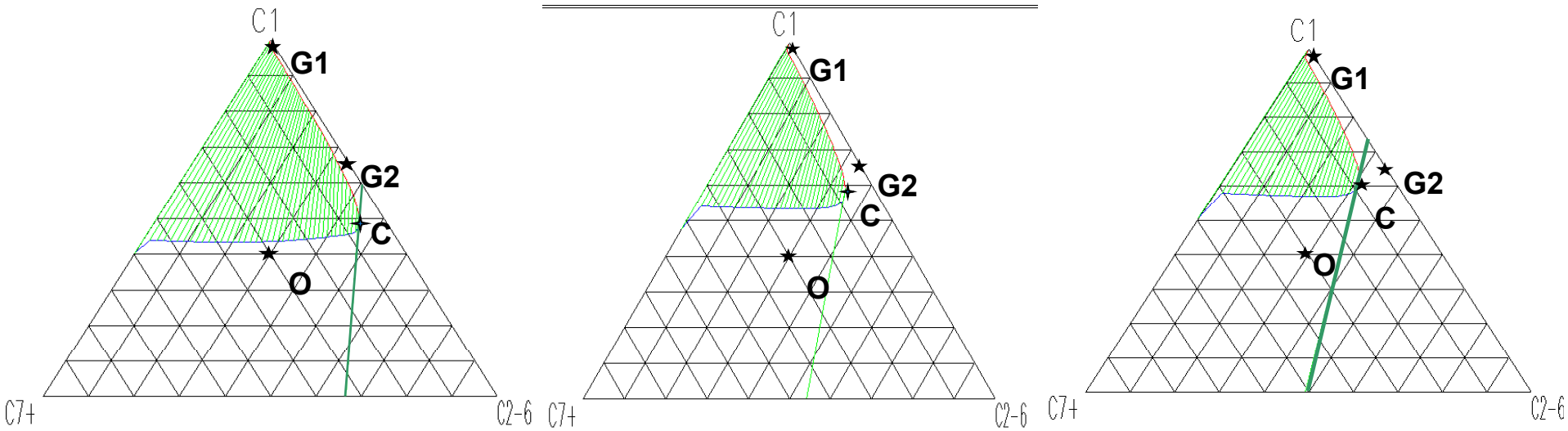


Рис. 1 - 3. Фазовая диаграмма при T=98°C, P=244, 350, 400 атм.

М. доля	C1	C2	C3	C4	C5	C6	C10	C15	C20	
O	0,4000	0,0800	0,0700	0,0450	0,0250	0,0750	0,1050	0,1000	0,1000	нефть
G1	0,9992	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	сухой газ
G2	0,6500	0,1250	0,1100	0,0650	0,0345	0,0140	0,0005	0,0005	0,0005	жирный газ

План доклада

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

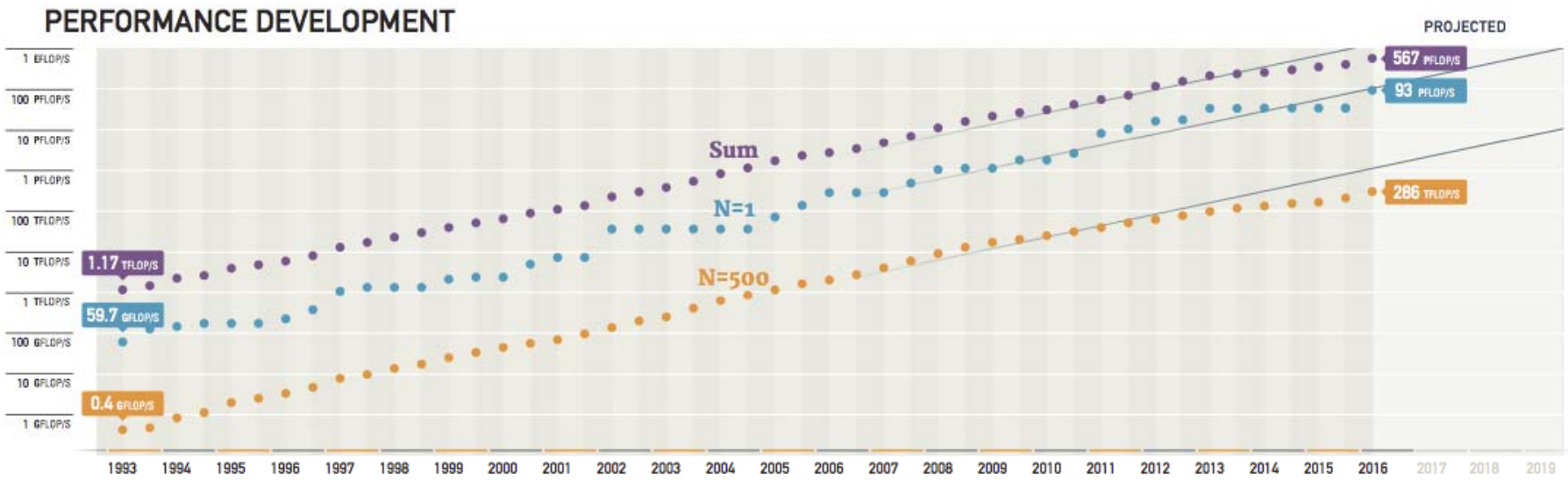
TOP 500 The List.

PRESENTED BY

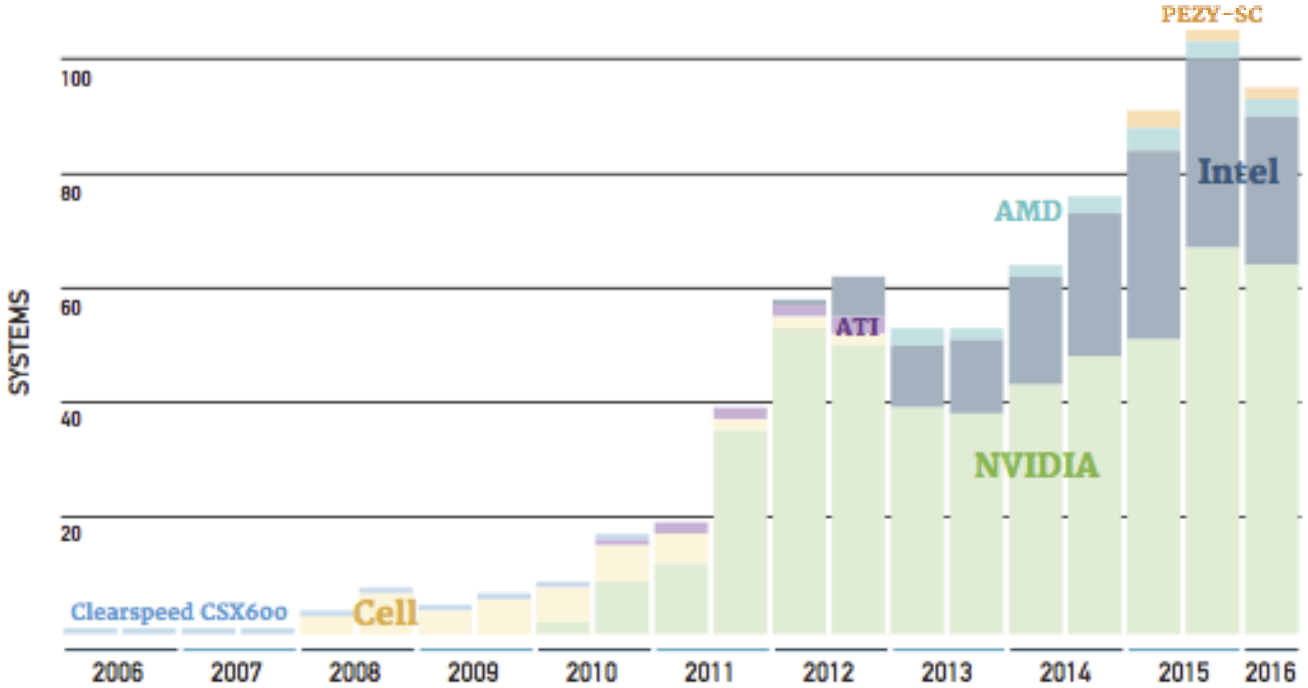
ICL INNOVATIVE COMPUTING LABORATORY · BERKELEY LAB Lawrence Berkeley National Laboratory · ISC GROUP Moving Forward.

FIND OUT MORE AT top500.org

	NAME	SPECS	SITE	COUNTRY	CORES	RMAX PFLOP/S	POWER MW
1	Sunway TaihuLight	Shenwei SW26010 (260C 1.45 GHz) Custom interconnect	NSCC in Wuxi	China	10,649,600	93.0	15.4
2	Tianhe-2 (Milkyway-2)	Intel Ivy Bridge (12C 2.2 GHz) & Xeon Phi (57C 1.1 GHz), Custom interconnect	NSCC in Guangzhou	China	3,120,000	33.9	17.8
3	Titan	Cray XK7, Opteron 6274 (16C 2.2 GHz) + Nvidia Kepler GPU, Custom interconnect	DOE/SC/ORNL	USA	560,640	17.6	8.2
4	Sequoia	IBM BlueGene/Q, Power BQC (16C 1.60 GHz), Custom interconnect	DOE/NNSA/LLNL	USA	1,572,864	17.2	7.9
5	K computer	Fujitsu SPARC64 VIIIfx (8C 2.0 GHz), Custom interconnect	RIKEN AICS	Japan	705,024	10.5	12.7



Top 500. Использование ускорителей



Accelerator/CP Family	Count	Rmax (GFlops)	Rpeak (GFlops)	Cores
Nvidia Kepler	53	63,601,255	97,087,546	1,624,745
Intel Xeon Phi	23	55,697,690	87,266,552	4,797,172
Nvidia Fermi	10	7,923,309	16,399,917	621,316
ATI Radeon	3	1,148,600	2,280,349	94,304
Hybrid	3	4,621,240	7,933,520	415,960
PEZY-SC	2	1,291,504	1,982,361	1,641,600

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
```

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;
}
```

Алгоритм Якоби в модели DVMH

```
FILE *f;
double A[L][L];
double B[L][L];
int main(int argc, char *argv[]) {
   for(int it = 0; it < ITMAX; it++) {
      #pragma dvm region
      {
         #pragma dvm parallel(2)
         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(2)
         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;
}
```

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

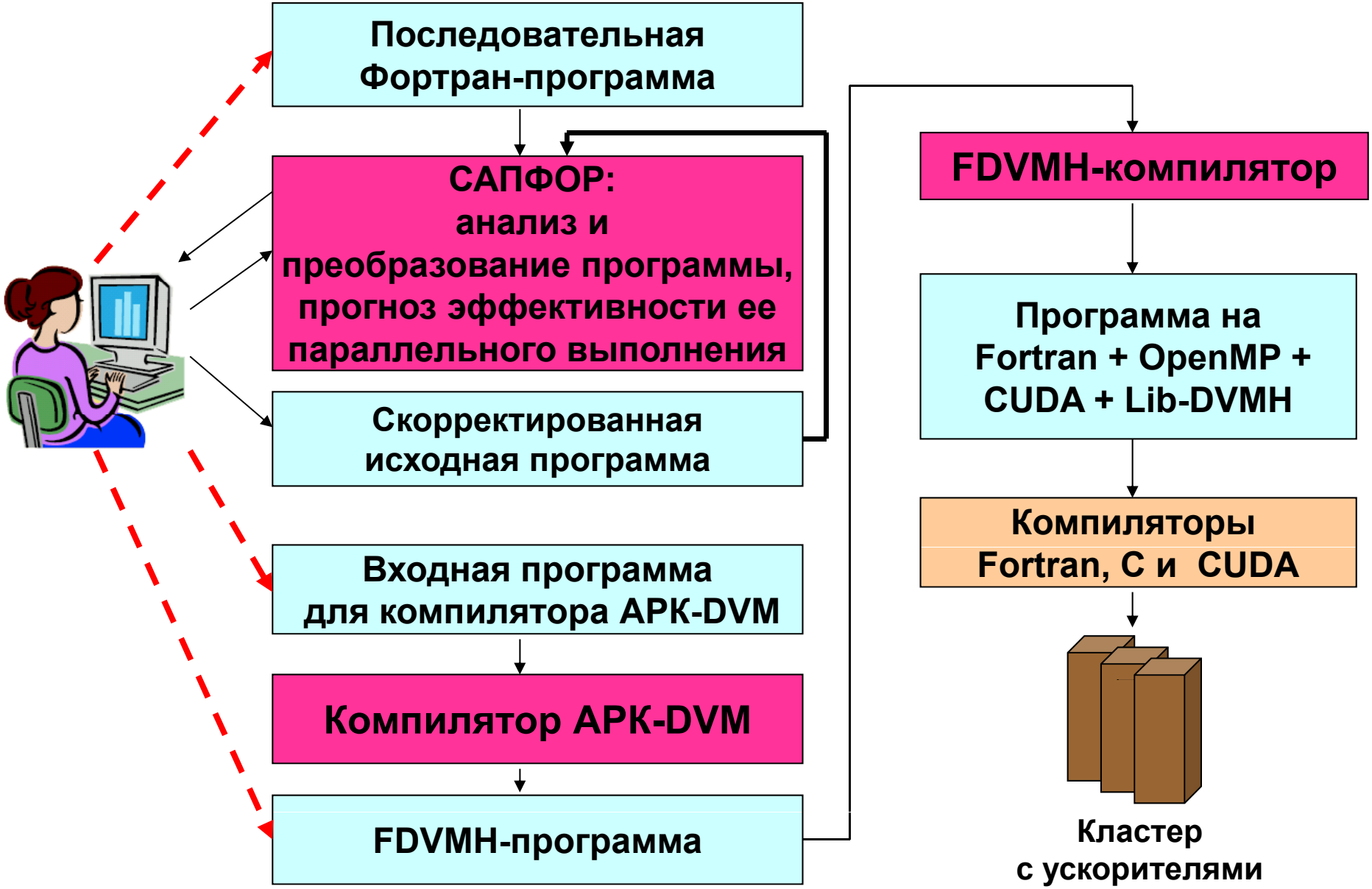
- Организация эффективного выполнения редукционных операций - глобальных операций с расположенными на различных процессорах/ускорителях данными (таких, как их суммирование или нахождение их максимального или минимального значения)
- Определение фрагментов программы (регионов) для возможного выполнения 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

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

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
```

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

План доклада

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

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

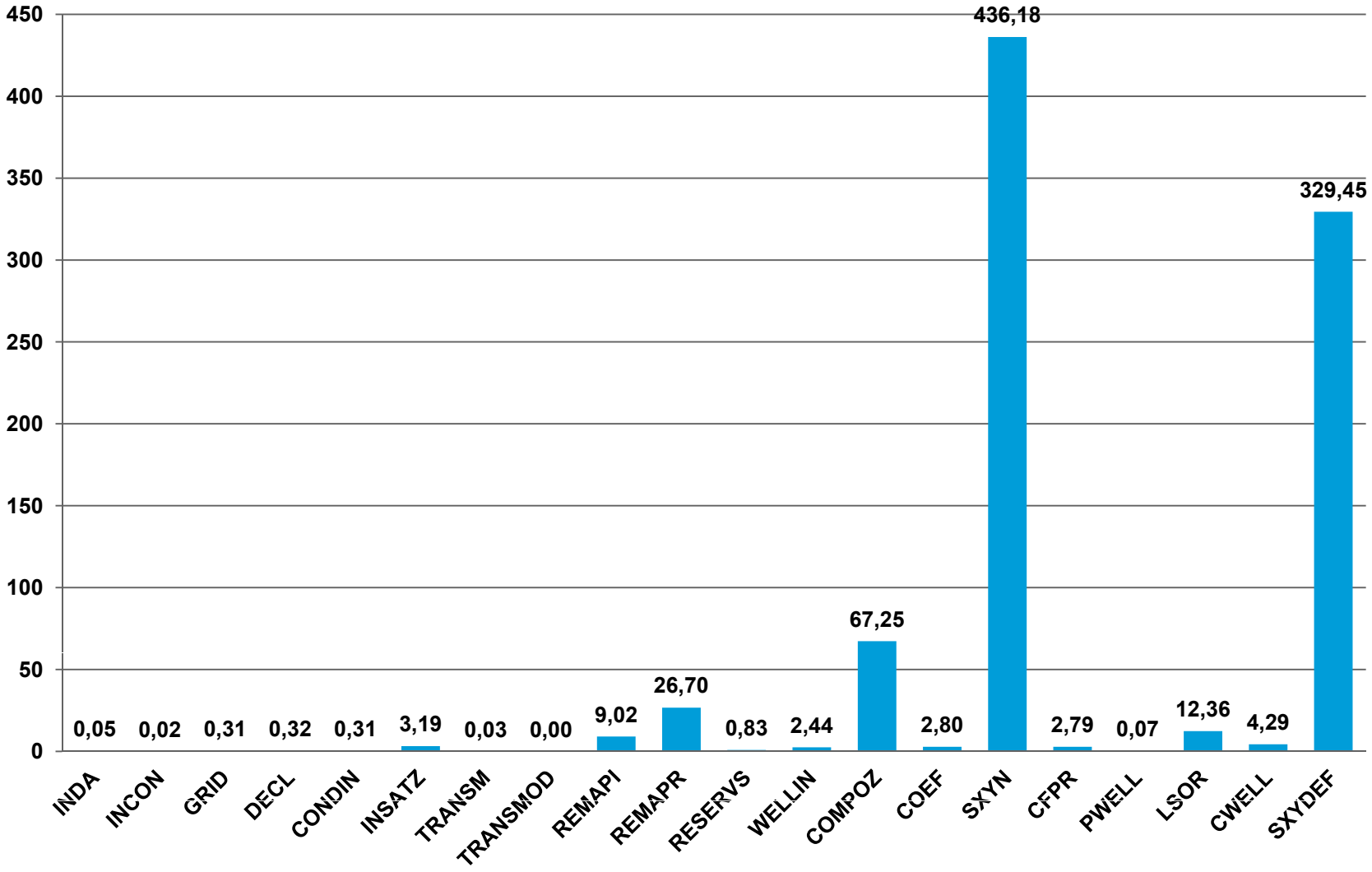
Сравнительная отладка программы

- 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

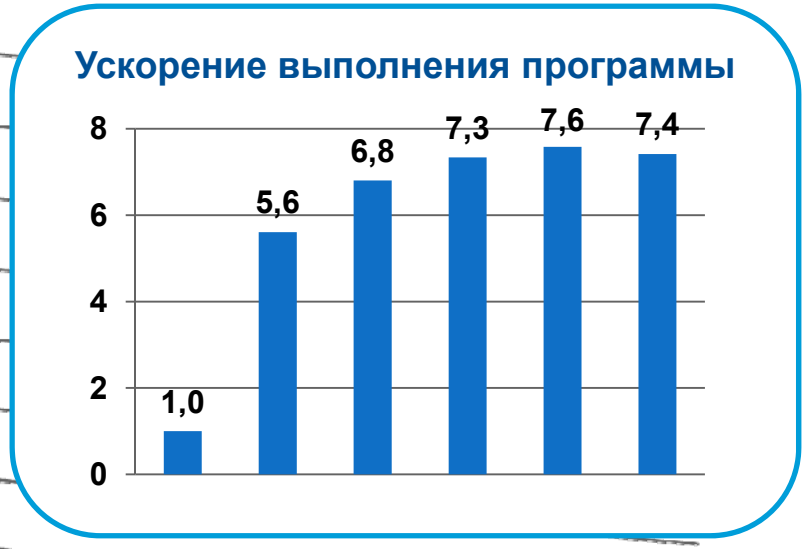


Распараллеливание времяемких фрагментов программы

- 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 К-100

На первом этапе были реализованы параллельные версии процедур COMPOZ, SXYN, SXYDEF, LSOR. (2700 из 13844 строк программы)

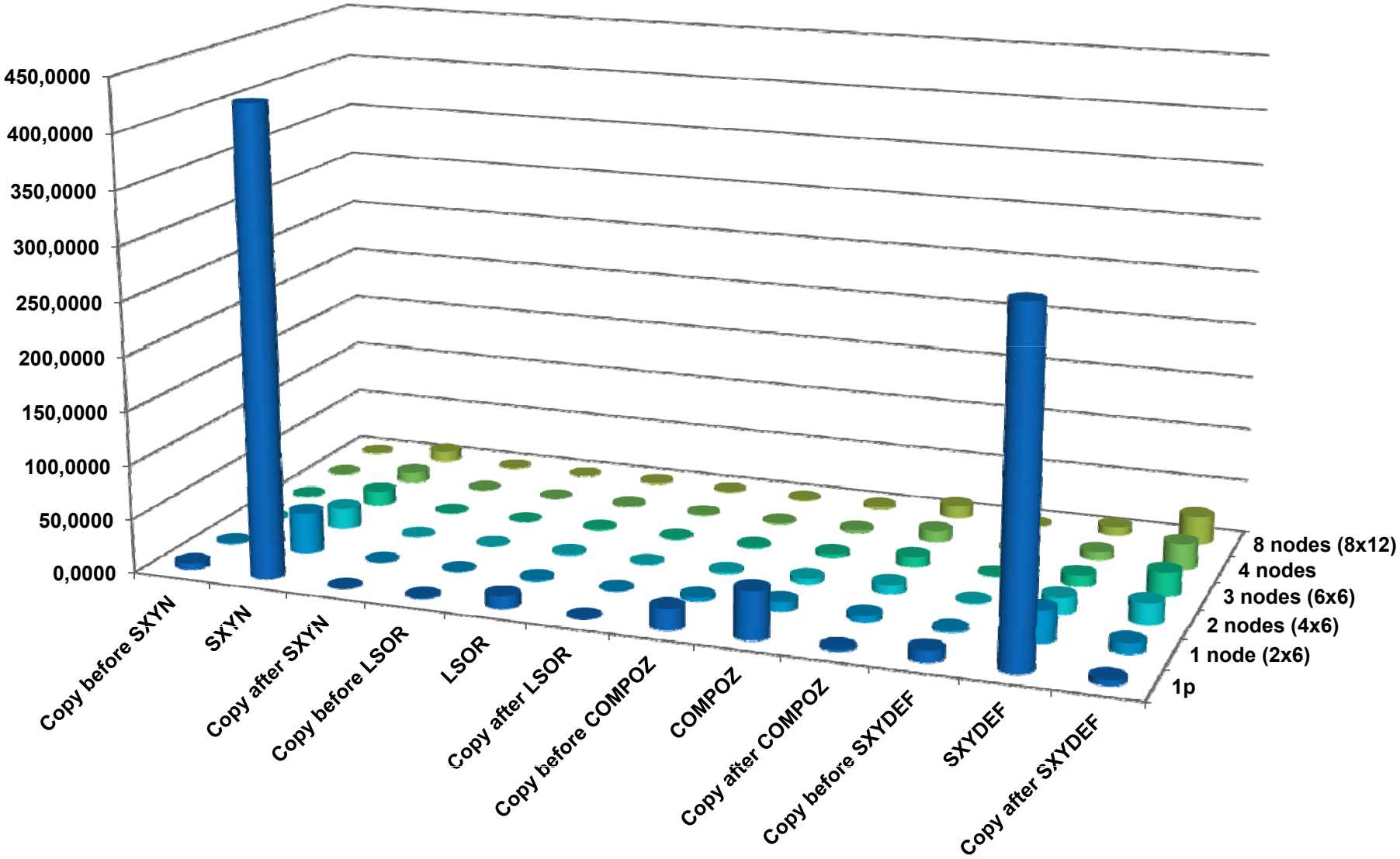


1000
900
800
700
600
500
400
300
200
100
0

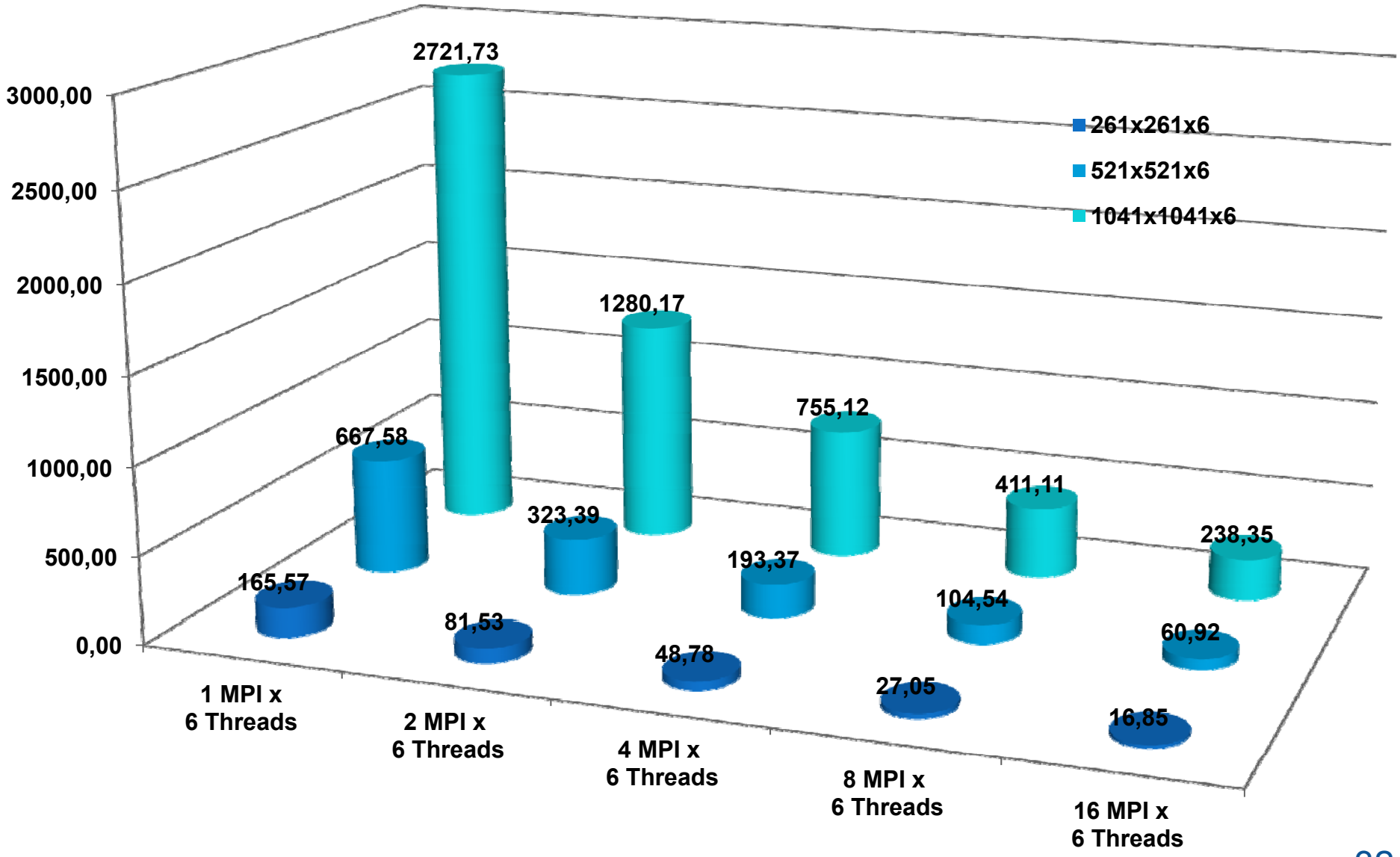
ALL PROGRAM, INDA, INCON, GRID, DECL, CONDIN, INSATZ, TRANSM, TRANSMOD, REMAPI, REMAPR, RESERVS, WELLIN, COMPOZ, COEF, SXYN, CFPR, PWELL, LSOR, CWELL, SXYDEF

4 nodes
2 nodes (4x6)
1p

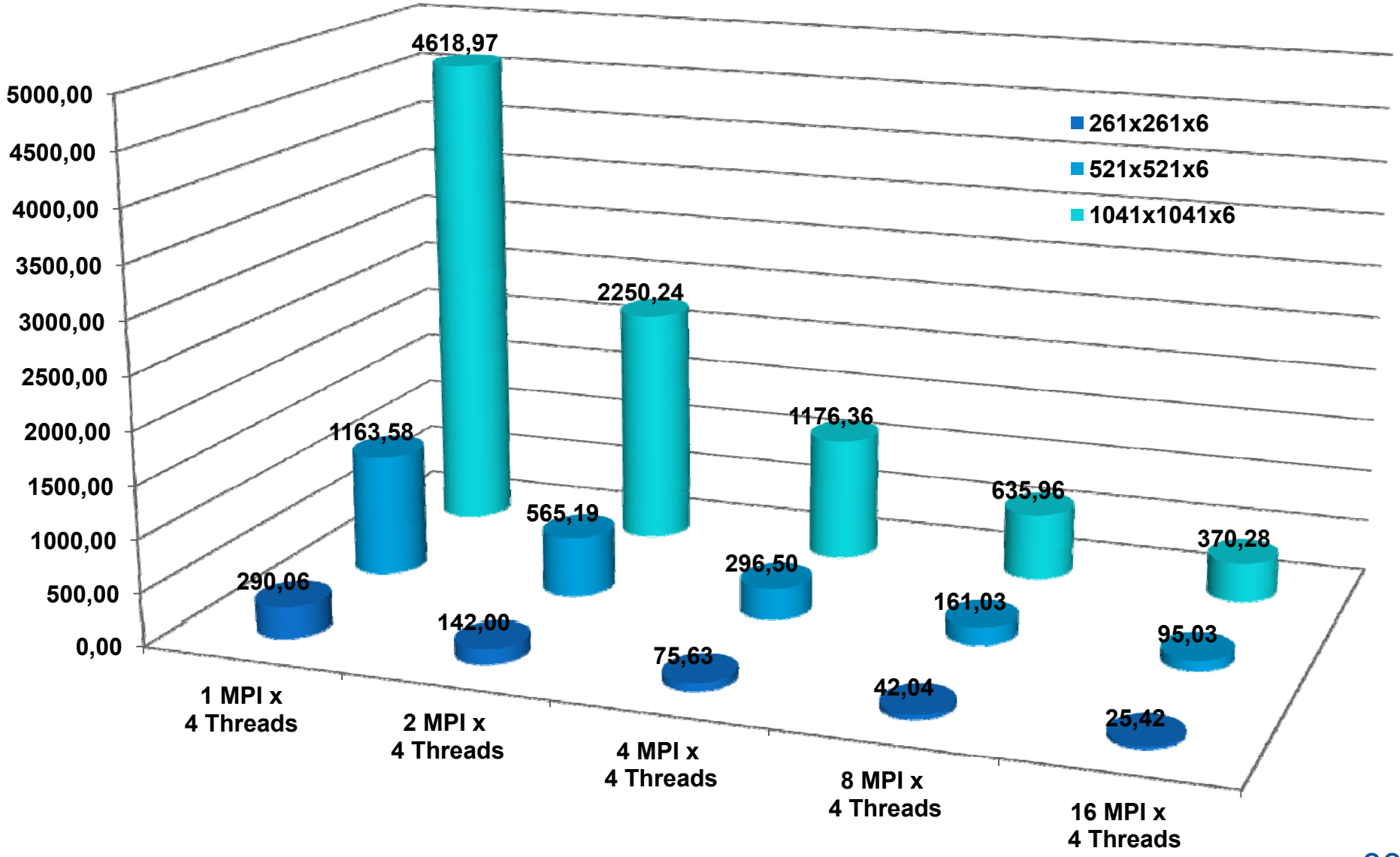
Накладные расходы на копирование данных между исходными массивами и их копиями



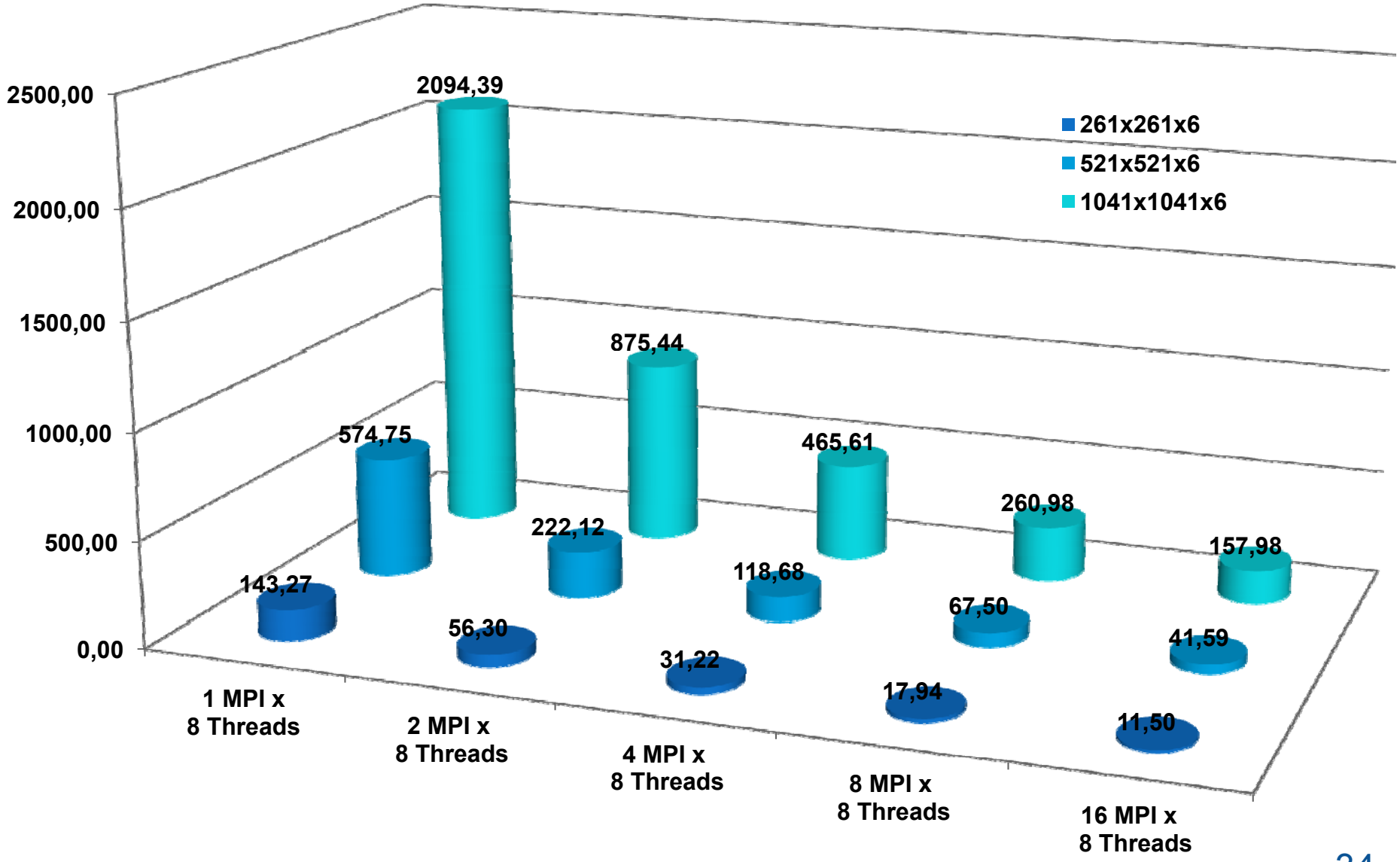
Времена выполнения 100 итераций программы Композит на суперкомпьютере К-100



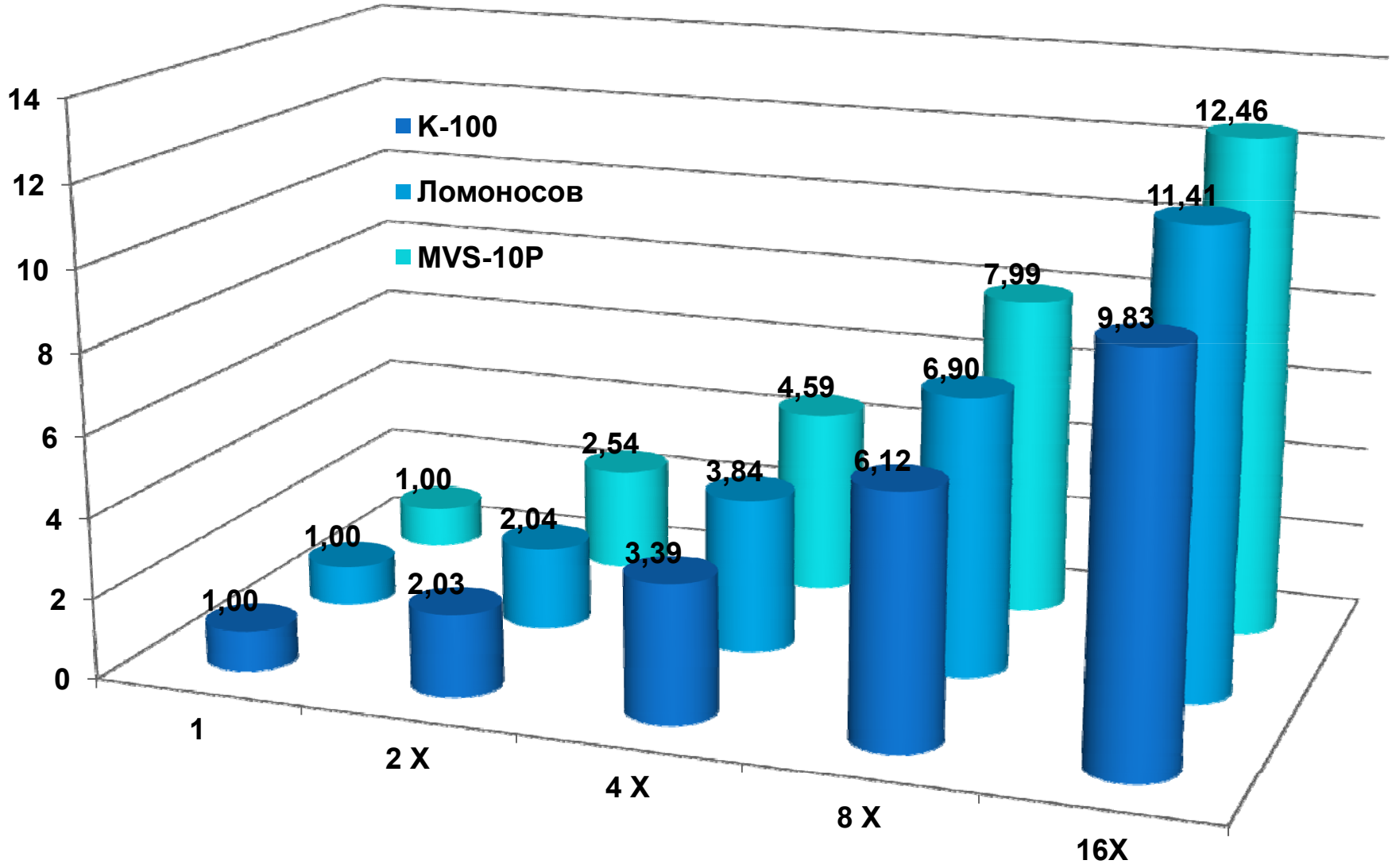
Времена выполнения 100 итераций программы Композит на суперкомпьютере Ломоносов



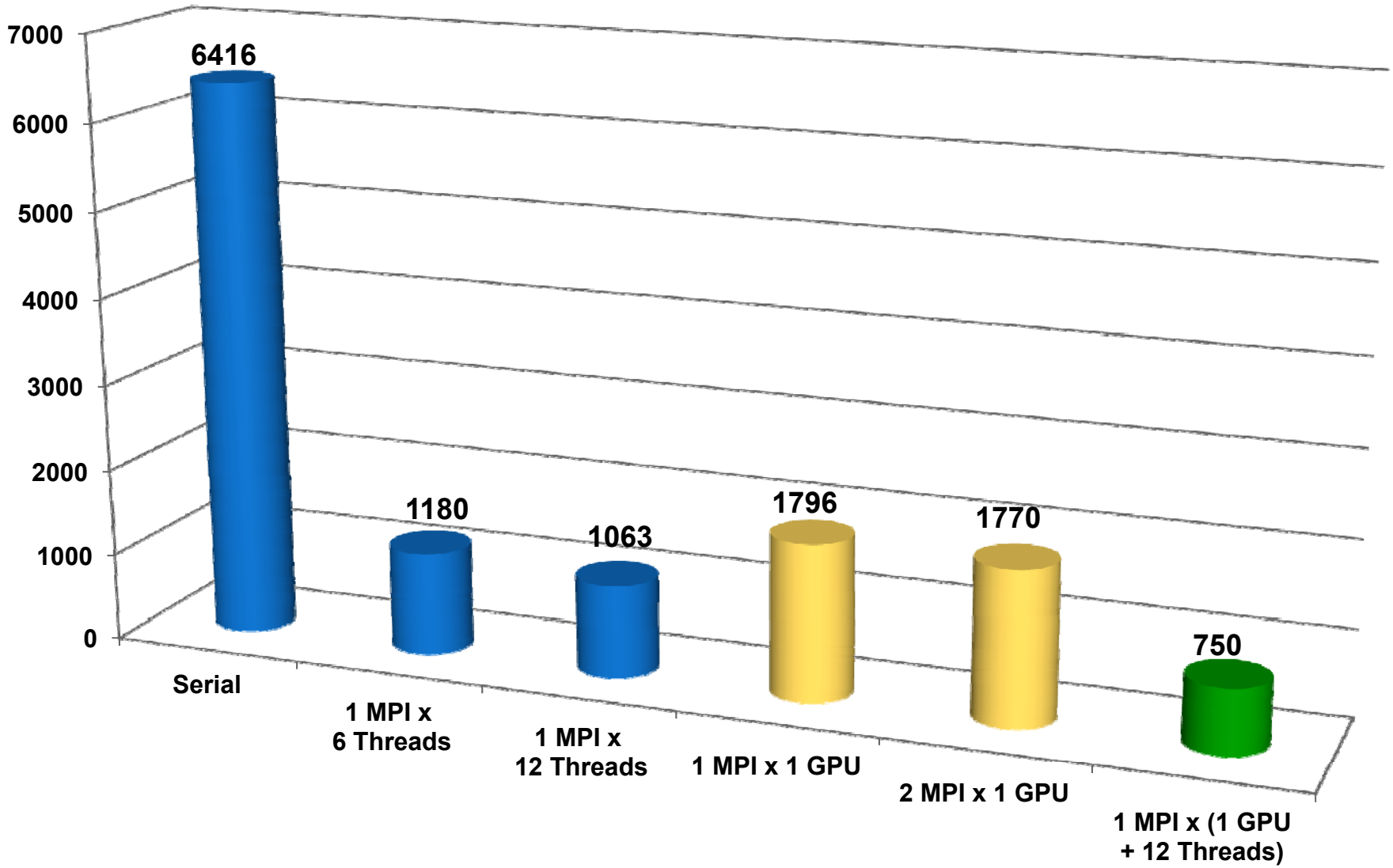
Времена выполнения 100 итераций программы Композит на суперкомпьютере MVS-10P



Ускорение выполнения 100 итераций программы Композит на сетке 261x261x6



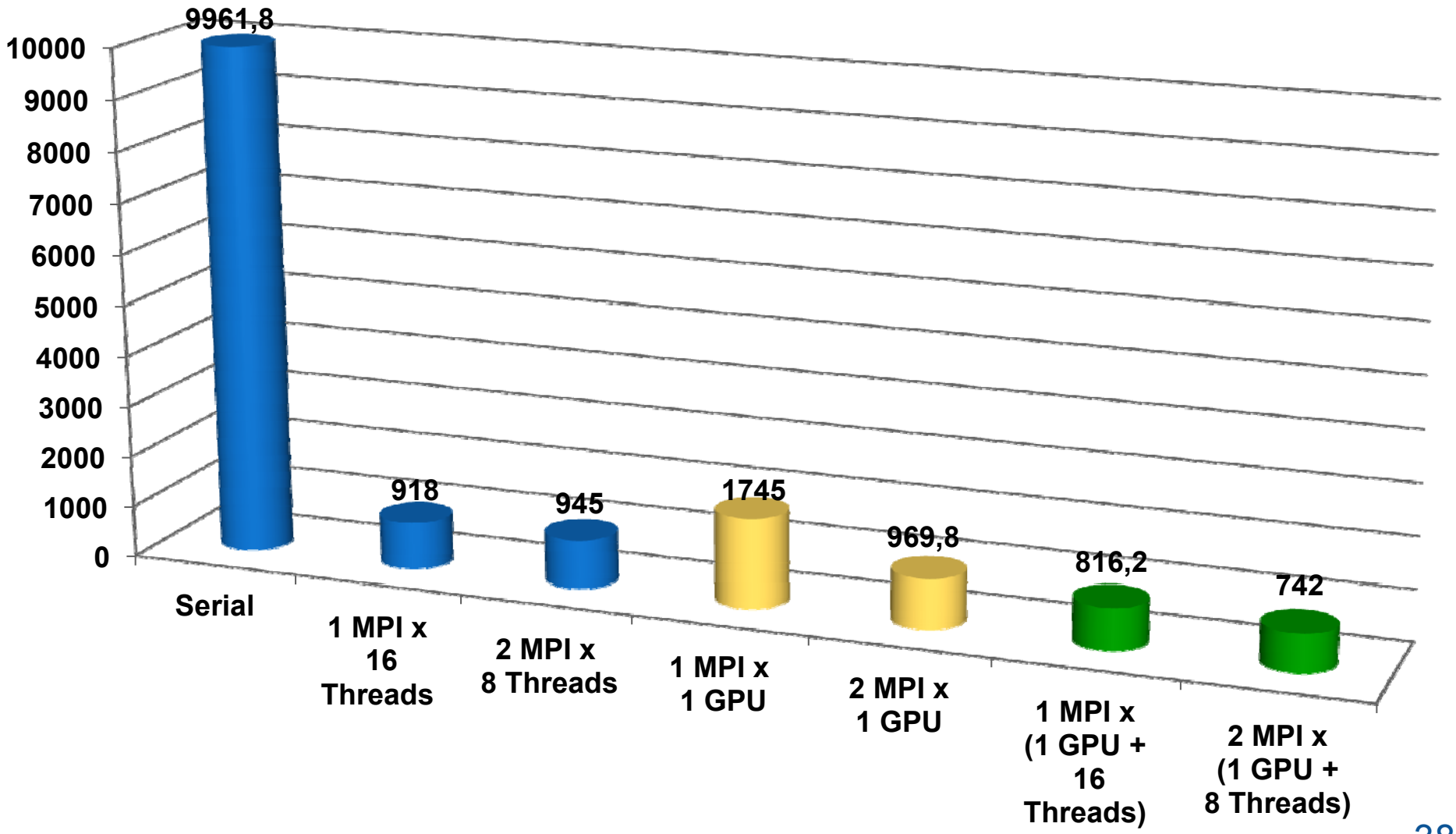
Время работы программы Композит на сетке 261x261x6 на гетерогенном узле (Intel Xeon E5-1660V2 + GPU NVIDIA Titan)



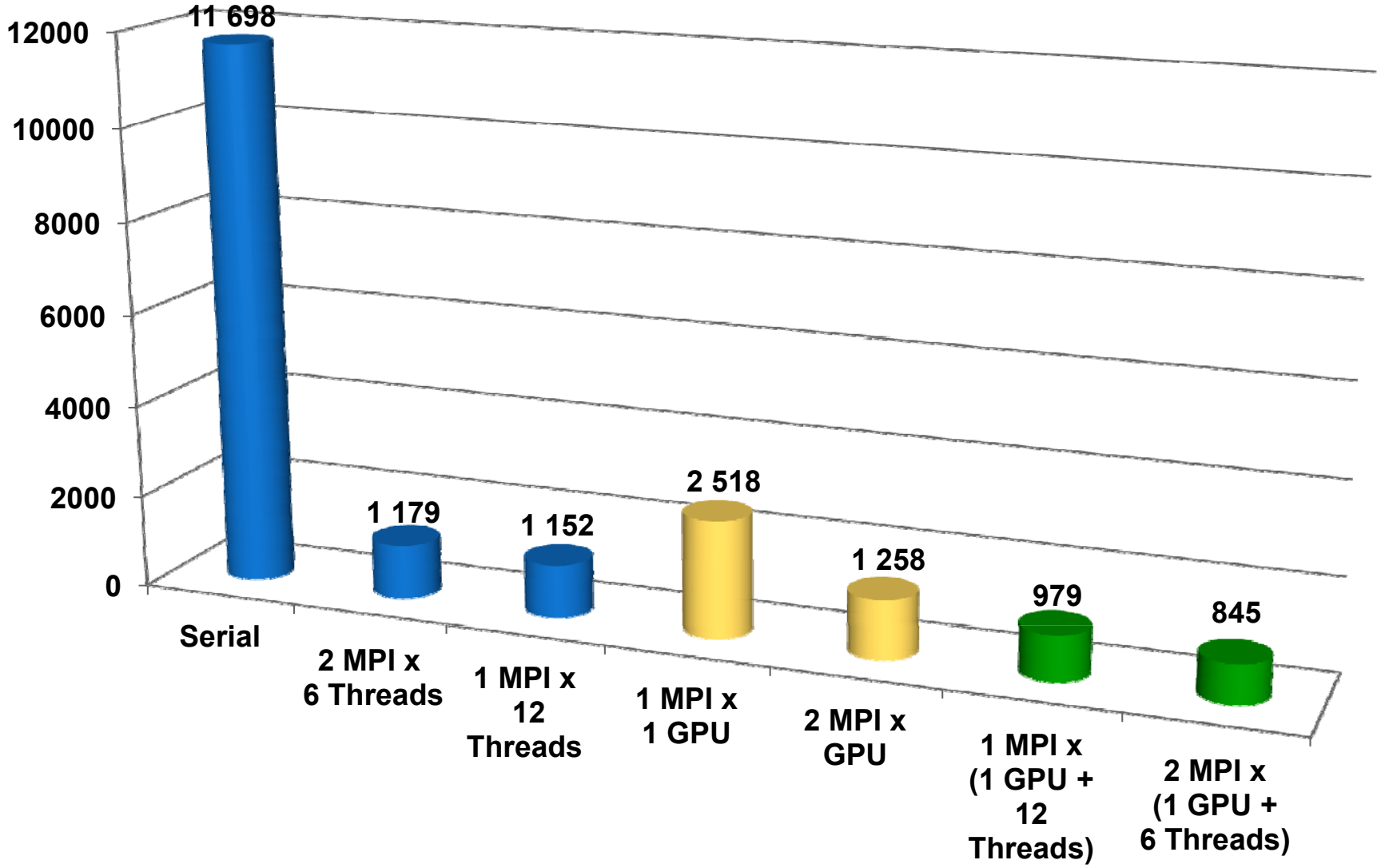
Новые возможности анализатора эффективности

```
--- The GPU characteristics ---

 ┌──────────────┐
 │ Proc: #1     │
 └──────────────┘
       │
   ┌──────────────────┐
   │ GPU #1 (Tesla X2070) │
   ├──────────────────────────────────────────────────────────────────────────────────────────────┐
   │                                        #          Min          Max          Sum      Average   Productive         Lost │
   ├──────────────────────────────────────────────────────────────────────────────────────────────┤
   │ [Region IN] Copy CPU to GPU            2           0B      976.624M       1.907G     976.624M      0.3252s            - │
   │ GET_ACTUAL                             1           0B      976.624M     976.624M     976.624M      0.1649s            - │
   │ Loop execution time                   41       0.0000        0.0907       2.5133       0.0613      2.5133s            - │
   │ Reduction                             20       0.0000        0.0084       0.0161       0.0008            -      0.0161s │
   ├──────────────────────────────────────────────────────────────────────────────────────────────┤
   │ Productive time:     3.0033s                                                                 │
   │ Lost time      :     0.0161s                                                                 │
   └──────────────────────────────────────────────────────────────────────────────────────────────┘

-------------------------------------------------------------------------
Name (number) and performance time of processors
node6-155-07(1) 1.046391
min - node6-155-07(1) 1.046391; max - node6-155-07(1) 1.046391; mid - 1.046391
```

Время работы программы Композит на сетке 261x261x6 на узле K-10 (2 x Intel Xeon E5-2660 + 2 x GPU NVIDIA Fermi M2090)



Время работы программы Композит на сетке 261x261x6 на узле K-100 (2 x Intel Xeon X5670 + 2 x GPU NVIDIA Fermi C2050)



Выводы

- На основе исходного кода 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 распараллеливании данной программы, будет использован при реализации следующей версии системы САПФОР

Вопросы, замечания?

СПАСИБО !

http://dvm-system.org

dvm@keldysh.ru