НАЦИОНАЛЬНЫЙ СУПЕРКОМПЬЮТЕРНЫЙ ФОРУМ 2017

Автоматизация разработки параллельных программ для кластеров с ускорителями

30 ноября 2017г. | Переславль-Залесский

В.А. Бахтин, Н.А. Катаев, А.С. Колганов, В.А. Крюков, Н.В. Поддерюгина

Институт прикладной математики им. М.В. Келдыша РАН
Факультет вычислительной математики и кибернетики МГУ им. М.В. Ломоносова

План доклада

- DVM-система
- Система SAPFOR
- Опыт разработки параллельных программ:
  - Моделирование распространения упругих волн
  - Многокомпонентная многофазная изотермическая фильтрация
- Выводы

Средства параллельного программирования

DVM - система

- Создана в Институте прикладной математики им. М.В. Келдыша РАН

- Аббревиатура DVM:
  Distributed Virtual Memory
  Distributed Virtual Machine

- Существует для двух языков:
  C-DVMH и *Fortran-DVMH*

- Предназначена для использования на кластерах с аппаратурой различной архитектуры (GPU Nvidia, Intel Xeon Phi, multicore CPUs)

Компоненты DVM - системы

- Компилятор Fortran-DVMH
- Компилятор C-DVMH
- Библиотека поддержки Lib-DVMH:
  DVMH Run Time System
- Отладчик DVMH-программ
- Анализатор производительности DVMH-программ

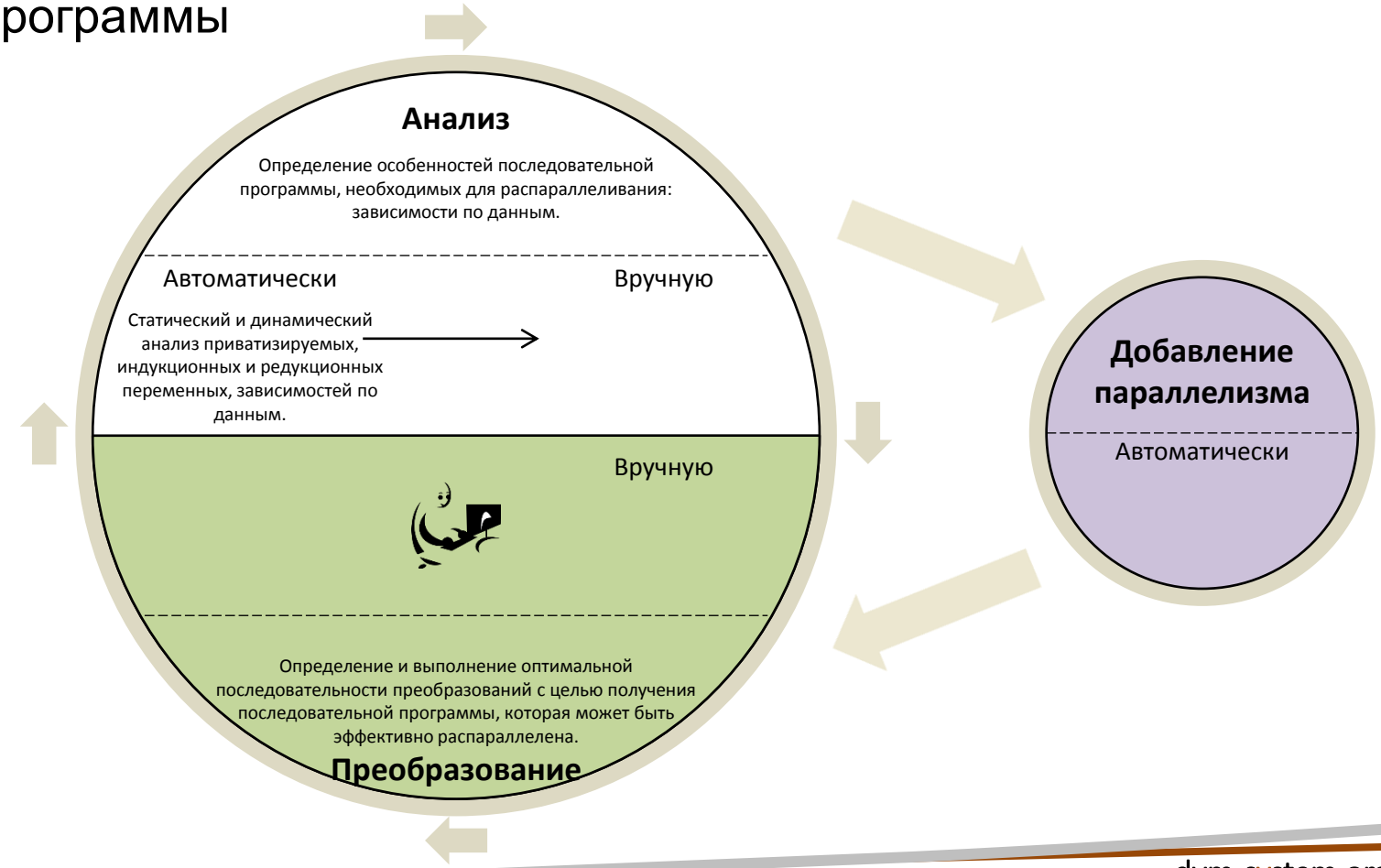
Средства программирования в DVM-системе

C-DVMH = Язык C 99 + спец. прагмы
Fortran-DVMH = Язык Fortran 95 + спец. комментарии

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

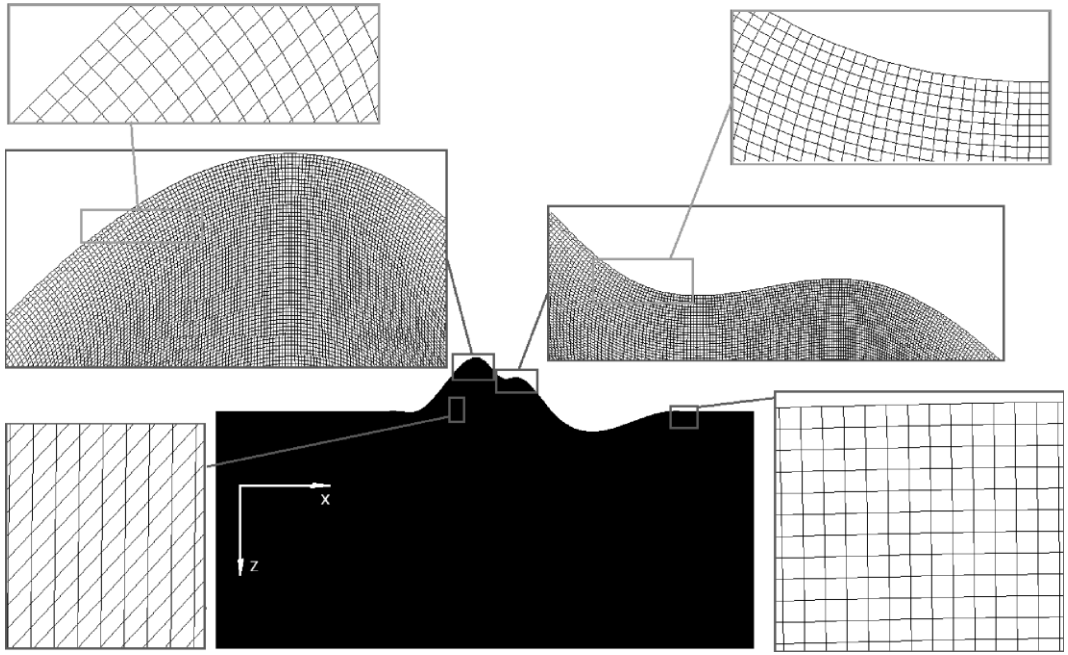
Автоматизация распараллеливания: SAPFOR

- Помогает программисту эффективно отображать его программы на многоядерные кластеры с ускорителями
- Организует параллельные вычисления полностью автоматически
- Взаимодействует с программистом в терминах последовательной программы



Моделирование распространения упругих волн в средах со сложной 3D геометрией

- Последовательная программа разработана в Институте вычислительной математики и математической геофизики СО РАН. Павел Титов
- Решение системы уравнений теории упругости в 3D области.
- Преобразование криволинейной сетки в структурированную: 2D срез



- **1 день для распараллеливания и 2 дня на получение результатов**

Этапы распараллеливания программы: SAPFOR

~ 3000 строк кода на языке Fortran 95, > 100 динамических массивов, > 90 циклов

Visualizer - [m_s_4_double_xy.for]

File Project View Tools Panels Window Help

Project
Source files
m_s_4_double

```
!================================== Inside medium ======================================
!DVM$ REGION

!DVM$ PARALLEL (k,j,i) on amu(i,j,k), SHADOW_RENEW(U(CORNER),
!DVM$& V(CORNER),W(CORNER)), PRIVATE(
!DVM$& dc1Ux_dq1,dc1Ux_dq2,dc1Ux_dq3,dc1Vy_dq1,dc1Vy_dq2,dc1Vy_dq3,
!DVM$& dc1Wz_dq1,dc1Wz_dq2,dc1Wz_dq3,dSxx_dx,dcUy_dq1,dcUy_dq2,
!DVM$& dcUy_dq3,dcVx_dq1,dcVx_dq2,dcVx_dq3,dSxy_dy,dcUz_dq1,dcUz_
!DVM$& dcUz_dq3,dcWx_dq1,dcWx_dq2,dcWx_dq3,dSxz_dz,dSxy_dx,dc
!DVM$& dc2Ux_dq2,dc2Ux_dq3,dc2Vy_dq1,dc2Vy_dq2,dc2Vy_dq3,dc2W
!DVM$& dc2Wz_dq2,dc2Wz_dq3,dSyy_dy,dcVz_dq1,dcVz_dq2,dcVz_d
!DVM$& dcWy_dq1,dcWy_dq2,dcWy_dq3,dSyz_dz,dSxz_dx,dSyz_dy,
!DVM$& dc3Ux_dq2,dc3Ux_dq3,dc3Vy_dq1,dc3Vy_dq3,dc3Wz_
!DVM$& dc3Wz_dq2,dc3Wz_dq3,dSzz_dz)
    do k = k_DOWN, k_UP
    do j = j_DOWN, j_UP
    do i = i_DOWN, i_UP

! dSxx_dx ------------------------------
    dc1Ux_dq1 =
   & rdd*((0.5d0*alambda(i+1,j,k)+amu(i+1,j,k)+
   &      0.5d0*alambda(i,j,k)+amu(i,j,k))*
   &       dq1_dx(2*i+1,2*j,2*k)*(U(i+1,j,k)-U(i,j,k)) -
   &      (0.5d0*alambda(i,j,k)+amu(i,j,k)+
```

До 53 приватных переменных в одном цикле.

Исследование свойств программы в SAPFOR.

Проверка корректности спецификаций в DVM.

Loop / PU	Progr...	File n...	Level...	Tigtl...	Dependency	... variables	Read variables
k=k_source-1,k_...		m_s_...	1.31.3	0	PRIV(i,j,k);	k	k_source
j=j_source,j_sou...		m_s_...	1.31....	1	PRIV(i,j);	j	j_source
i=i_source,i_sou...		m_s_...	1.31....	1	PRIV(i);	i;u(i,j,k);v(i,j,k);...	i_source;k;k_so...
k=k_source,k_s...		m_s_...	1.31.4	0	PRIV(i,j,k);	k	k_source
j=j_source,j_sou...		m_s_...	1.31....	1	PRIV(i,j);	j	j_source
i=i_source-1,i_s...		m_s_...	1.31....	1	PRIV(i);	i;uu(i,j,k);vv(i,j,...	i_source;i;uu(i,j...
k=k_source,k_s...		m_s_...	1.31.5	0	PRIV(i,j,k);	k	k_source
j=j_source-1,j_s...		m_s_...	1.31....	1	PRIV(i,j);	j	j_source
i=i_source,i_sou...		m_s_...	1.31....	1	PRIV(i);	i;uu(i,j,k);vv(i,j,...	i_source;j;j_sou...
k=k_source-1,k_...		m_s_...	1.31.6	0	PRIV(i,j,k);	k	k_source
j=j_source,j_sou...		m_s_...	1.31....	1	PRIV(i,j);	j	j_source
i=i_source,i_sou...		m_s_...	1.31....	1	PRIV(i);	i;uu(i,j,k);vv(i,j,...	i_source;k;k_so...
i_time=i_time,iti...		m_s_...	1.32	0	PRIV(i_time);OUT+FLOW+ANTI(v,vv,w,ww,u,uu,alambda,amu,ttorhoj,dq1_dx,dq2_dx,dq3_dx,dq1_dy,dq2_dy,dq3_dy,dq1_dz,dq2_dz,dq3_dz,a1,...	i_time	i_time;itime
k=k_down,k_up	com...	m_s_...	2.1	0	PRIV(dcvz_dq2,dcuz_dq3,dc2vy_dq3,dc1vy_dq1,dcvx_dq1,dsyz_dy,dc1wz_dq2,dsxz_dz,dcwy_dq2,dc3wz_dq2,dcwx_dq3,dc2wz_dq3,dc1wz_dq1,d...	k	k_down;k_up
j=j_down,j_up	com...	m_s_...	2.1.1	1	PRIV(dcvz_dq2,dcuz_dq3,dc2vy_dq3,dc1vy_dq1,dcvx_dq1,dsyz_dy,dc1wz_dq2,dsxz_dz,dcwy_dq2,dc3wz_dq2,dcwx_dq3,dc2wz_dq3,dc1wz_dq1,d...	j	j_down;j_up
i=i_down,i_up	com...	m_s_...	2.1.1.1	1	PRIV(dcvz_dq2,dcuz_dq3,dc2vy_dq3,dc1vy_dq1,dcvx_dq1,dsyz_dy,dc1wz_dq2,dsxz_dz,dcwy_dq2,dc3wz_dq2,dcwx_dq3,dc2wz_dq3,dc1wz_dq1,d...	i;dc1ux_dq1;dc...	i_down;i_up;rd...
j=n_down_y,n_...	edges	m_s_...	4.1	0	PRIV(k,r2,dv_dq1,r3,dv_dq2,znam1,dw_dq1,znam2,dw_dq2,znam3,znam4,i,du_dq1,j,r1,du_dq2);OUT+FLOW+ANTI(dv_dq3,dw_dq3,du_dq3);RE...	j	n_down_y;n_u...
k=0,0	edges	m_s_...	4.1.1	1	PRIV(k,r2,dv_dq1,r3,dv_dq2,znam1,dw_dq1,znam2,dw_dq2,znam3,znam4,i,du_dq1,r1,du_dq2);	k	
i=n_down_x,n_u...	edges	m_s_...	4.1.1.1	1	PRIV(r2,dv_dq1,r3,dv_dq2,znam1,dw_dq1,znam2,dw_dq2,znam3,znam4,i,du_dq1,r1,du_dq2);OUT+FLOW+ANTI(dv_dq3,dw_dq3,du_dq3);	i;znam1;znam2...	n_down_x;n_u...

Output Characteristics

Ready Ln 1618

Этапы распараллеливания программы: DVM

- Распределение элементов массива между процессорами:
  директивы ***distribute / align***
- Распределение витков цикла между вычислительными устройствами:
  директива ***parallel***
- Организация эффективного доступа к удаленным данным, расположенным на других вычислительных устройствах:
  спецификации ***shadow / across / remote***
- Организация эффективного выполнения редукционных операций – глобальных операций с расположенными на различных вычислителях данными:
  спецификация ***reduction: max/min/sum/maxloc/minloc/…***
- Определение фрагментов программы (регионов) для возможного выполнения на ускорителях и многоядерных CPU:
  директива ***region***
- Управление перемещением данных между памятью CPU и памятью GPU:
  директивы ***actual / get_actual***

Распределение данных

Поддержка динамических массивов

```
double precision, allocatable::
      & dq1_dx(:,:,:),dq2_dx(:,:,:),dq3_dx(:,:,:),
      & dq1_dy(:,:,:),dq2_dy(:,:,:),dq3_dy(:,:,:),
      & dq1_dz(:,:,:),dq2_dz(:,:,:),dq3_dz(:,:,:),
        ...
```

Директивы не зависят от количества используемых процессоров

```
!DVM$ DISTRIBUTE dq1_dx(BLOCK, BLOCK, *)
!DVM$ ALIGN (i,j,k) WITH dq1_dx(i,j,k)::dq2_dx, dq3_dx, dq1_dy, dq2_dy
!DVM$ ALIGN (i,j,k) WITH dq1_dx(i,j,k)::dq3_dy, dq1_dz, dq2_dz, dq3_dz
        ...
```

Выделение памяти стандартными операторами языка Fortran

```
        allocate(
      & dq1_dx(2*N_down_x-3:2*N_up_x+3,2*N_down_y-3:2*N_up_y+3,
      & 2*N_down_z-3:2*N_up_z+3),
      & dq2_dx(2*N_down_x-3:2*N_up_x+3,2*N_down_y-3:2*N_up_y+3,
      & 2*N_down_z-3:2*N_up_z+3),
      & dq3_dx(2*N_down_x-3:2*N_up_x+3,2*N_down_y-3:2*N_up_y+3,
      & 2*N_down_z-3:2*N_up_z+3),
        ...)
```

Распределение вычислений

Информация о редукционных и приватных переменных получена от SAPFOR

```
!DVM$ PARALLEL (k,j,i) on amu(i,j,k), PRIVATE(
!DVM$& dc1Ux_dq1,dc1Ux_dq2,dc1Ux_dq3,dc1Vy_dq1,dc1Vy_dq2,dc1Vy_dq3,
!DVM$& dc1Wz_dq1,dc1Wz_dq2,dc1Wz_dq3,dSxx_dx,dcUy_dq1,dcUy_dq2,
        ...)
      do k = k_DOWN, k_UP
      do j = j_DOWN, j_UP
      do i = i_DOWN, i_UP
        ...
```

```
!DVM$ PARALLEL (k,j,i) ON X(2*i,2*j,2*k),
!DVM$& REDUCTION(MINLOC(amax,newSources,3))
      do k=N_down_z,N_up_z
      do j=N_down_y,N_up_y
      do i=N_down_x,N_up_x

      if(((X(2*i,2*j,2*k)-x0)**2 +
     &    (Y(2*i,2*j,2*k)-y0)**2 +
     &    (Z(2*i,2*j,2*k)-z0)**2 < amax)) then
        ...
```

Организация доступа к удаленным данным

Динамический контроль корректности директив с помощью отладчика DVM-системы

./dvm fpdeb M_S_4_double_xy.for

./dvm err M_S_4_double_xy

```
!DVM$ PARALLEL (k,j,i) on amu(i,j,k), PRIVATE(
!DVM$& dc1Ux_dq1,dc1Ux_dq2,dc1Ux_dq3,dc1Vy_dq1,dc1Vy_dq2,dc1Vy_dq3,
!DVM$& dc1Wz_dq1,dc1Wz_dq2,dc1Wz_dq3,dSxx_dx,dcUy_dq1,dcUy_dq2,
       ...)
      do k = k_DOWN, k_UP
      do j = j_DOWN, j_UP
      do i = i_DOWN, i_UP
       dc1Ux_dq1 =
       ...
     & 0.25d0*rdd*( (alambda(i+1,j,k)+2.0d0*amu(i+1,j,k))*
     &  dq2_dx(2*i+2,2*j,2*k)*(U(i+1,j+1,k)-U(i+1,j-1,k)) -
     &              (alambda(i-1,j,k)+2.0d0*amu(i-1,j,k))*
     &  dq2_dx(2*i-2,2*j,2*k)*(U(i-1,j+1,k)-U(i-1,j-1,k)) ) +
       ...
```

*** DYNCONTROL *** : Loop(No(46), Iter(0)), Loop(No(54), Iter(1,2,2)).

Access to non-local element dq2 _dx(2 * i + 2,2 * j,2 * k)

File: M_S_4_double_xy.for Line: 1710

Организация доступа к удаленным данным

```
!DVM$ SHADOW(1:1, 1:1, 1:2) :: U,V,W
!DVM$ SHADOW(2:2, 2:2, 2:2) :: dq1_dx,dq2_dx, dq3_dx, dq1_dy, dq2_dy
!DVM$ SHADOW(2:2, 2:2, 2:2) :: dq3_dy, dq1_dz, dq2_dz, dq3_dz,X,Y,Z
      ...
!DVM$ PARALLEL (k,j,i) on amu(i,j,k), PRIVATE(
!DVM$& dc1Ux_dq1,dc1Ux_dq2,dc1Ux_dq3,dc1Vy_dq1,dc1Vy_dq2,dc1Vy_dq3,
!DVM$& dc1Wz_dq1,dc1Wz_dq2,dc1Wz_dq3,dSxx_dx,dcUy_dq1,dcUy_dq2,
      ...)
      do k = k_DOWN, k_UP
      do j = j_DOWN, j_UP
      do i = i_DOWN, i_UP
       dc1Ux_dq1 =
      ...
     & 0.25d0*rdd*( (alambda(i+1,j,k)+2.0d0*amu(i+1,j,k))*
     &  dq2_dx(2*i+2,2*j,2*k)*(U(i+1,j+1,k)-U(i+1,j-1,k)) -
     &              (alambda(i-1,j,k)+2.0d0*amu(i-1,j,k))*
     &  dq2_dx(2*i-2,2*j,2*k)*(U(i-1,j+1,k)-U(i-1,j-1,k)) ) +
      ...
```

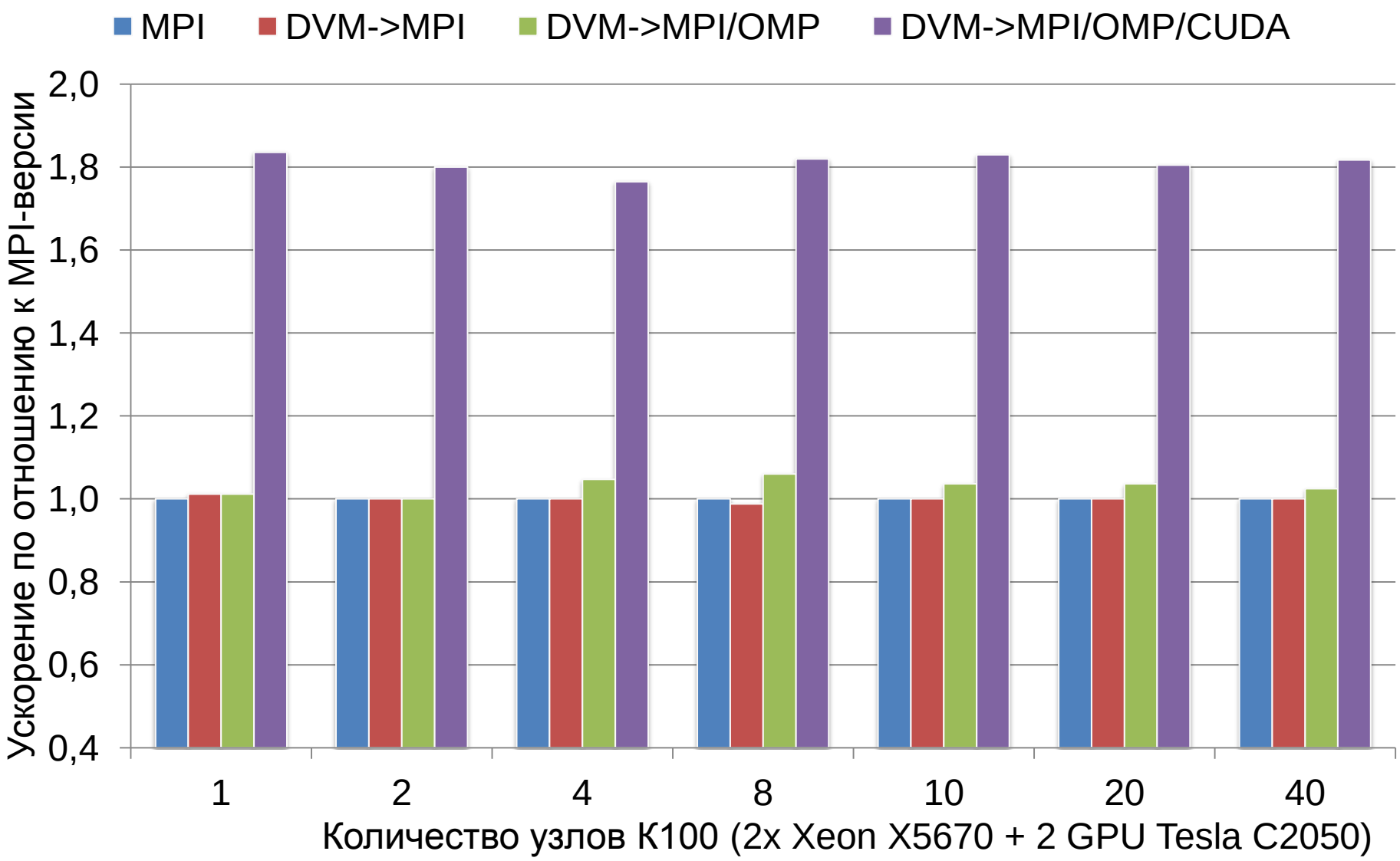
Достаточно увеличить ширину теневой грани до 2 элементов (по умолчанию 1 элемент)

Указание регионов для выполнения на ускорителе

Позволяет выполнять программу на мультипроцессоре, GPU, Xeon Phi

```
!DVM$ REGION
!DVM$ PARALLEL (k,j,i) on amu(i,j,k), PRIVATE(
!DVM$& dc1Ux_dq1,dc1Ux_dq2,dc1Ux_dq3,dc1Vy_dq1,dc1Vy_dq2,dc1Vy_dq3,
!DVM$& dc1Wz_dq1,dc1Wz_dq2,dc1Wz_dq3,dSxx_dx,dcUy_dq1,dcUy_dq2,
       ...)
      do k = k_DOWN, k_UP
      do j = j_DOWN, j_UP
      do i = i_DOWN, i_UP
       dc1Ux_dq1 =
       ...
     & 0.25d0*rdd*( (alambda(i+1,j,k)+2.0d0*amu(i+1,j,k))*
     &  dq2_dx(2*i+2,2*j,2*k)*(U(i+1,j+1,k)-U(i+1,j-1,k)) -
     &              (alambda(i-1,j,k)+2.0d0*amu(i-1,j,k))*
     &  dq2_dx(2*i-2,2*j,2*k)*(U(i-1,j+1,k)-U(i-1,j-1,k)) ) +
       ...
      end do
      end do
      end do
!DVM$ END REGION
```

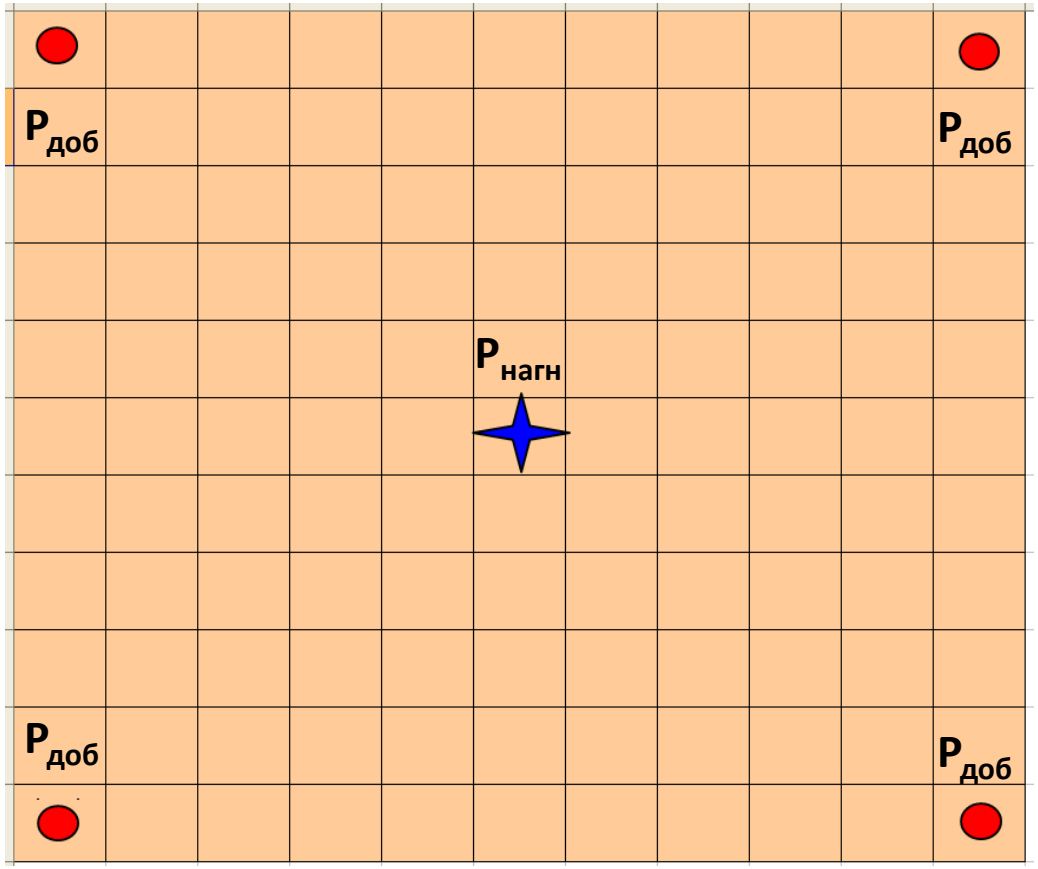
Сравнение с ручным распараллеливанием



Многокомпонентная многофазная изотермическая фильтрация

- Композиционные модели фильтрации используются при подробном моделировании залежей, содержащих легкие углеводороды (конденсат и газ) в том случае, когда необходимо тщательно описывать массообмен между фазами, либо когда пластовые флюиды содержат ценные неуглеводородные компоненты
- Часто используются для изучения методов увеличения нефтеотдачи при закачке газов высокого давления, азота, углекислого газа и других агентов
- Последовательная версия программы (далее **Композит**) разработана в НИИСИ РАН. Королев А.В.

Тестовые расчеты 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

Этапы распараллеливания 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

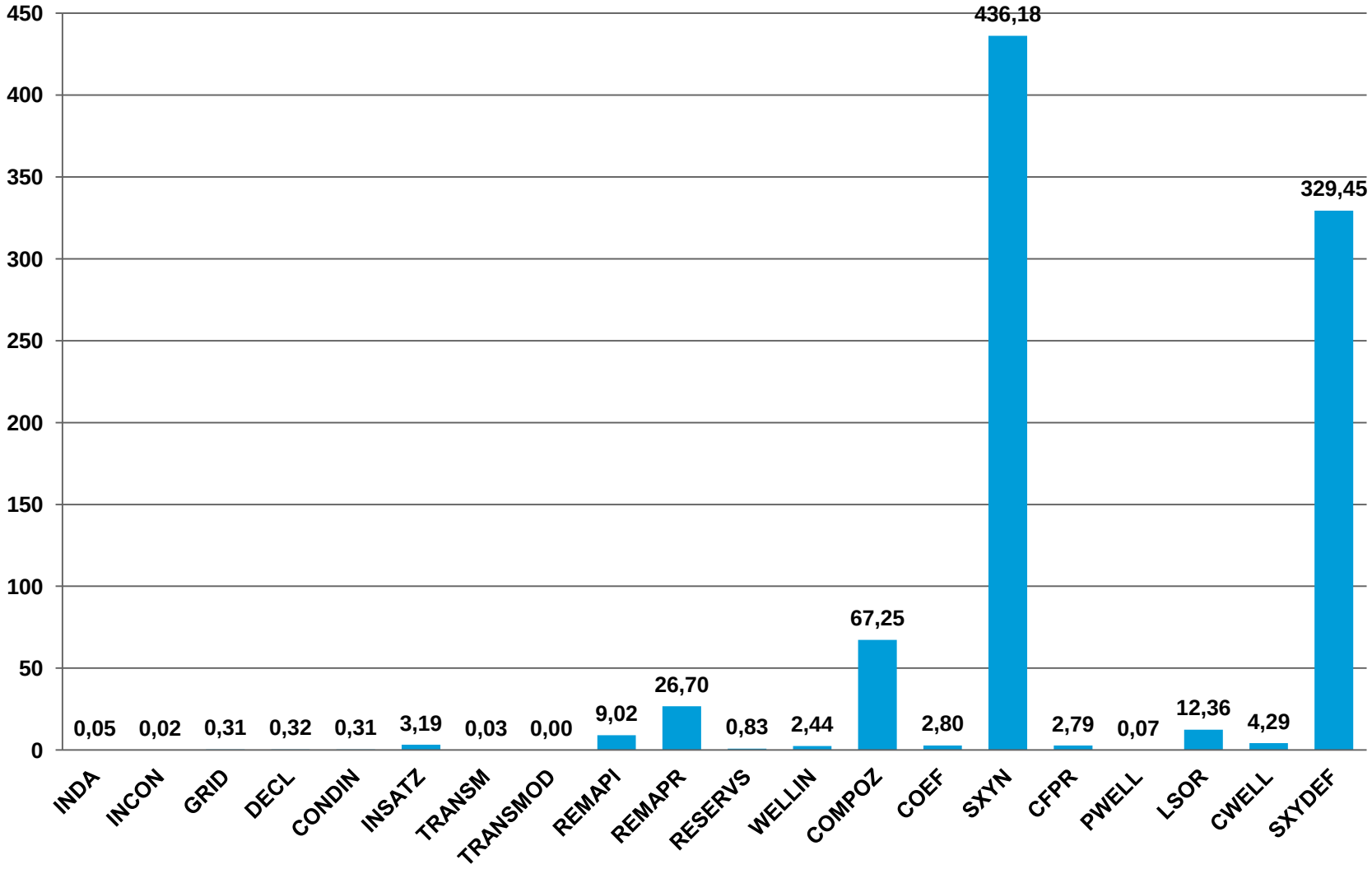
Сравнительная отладка программы

- 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
```

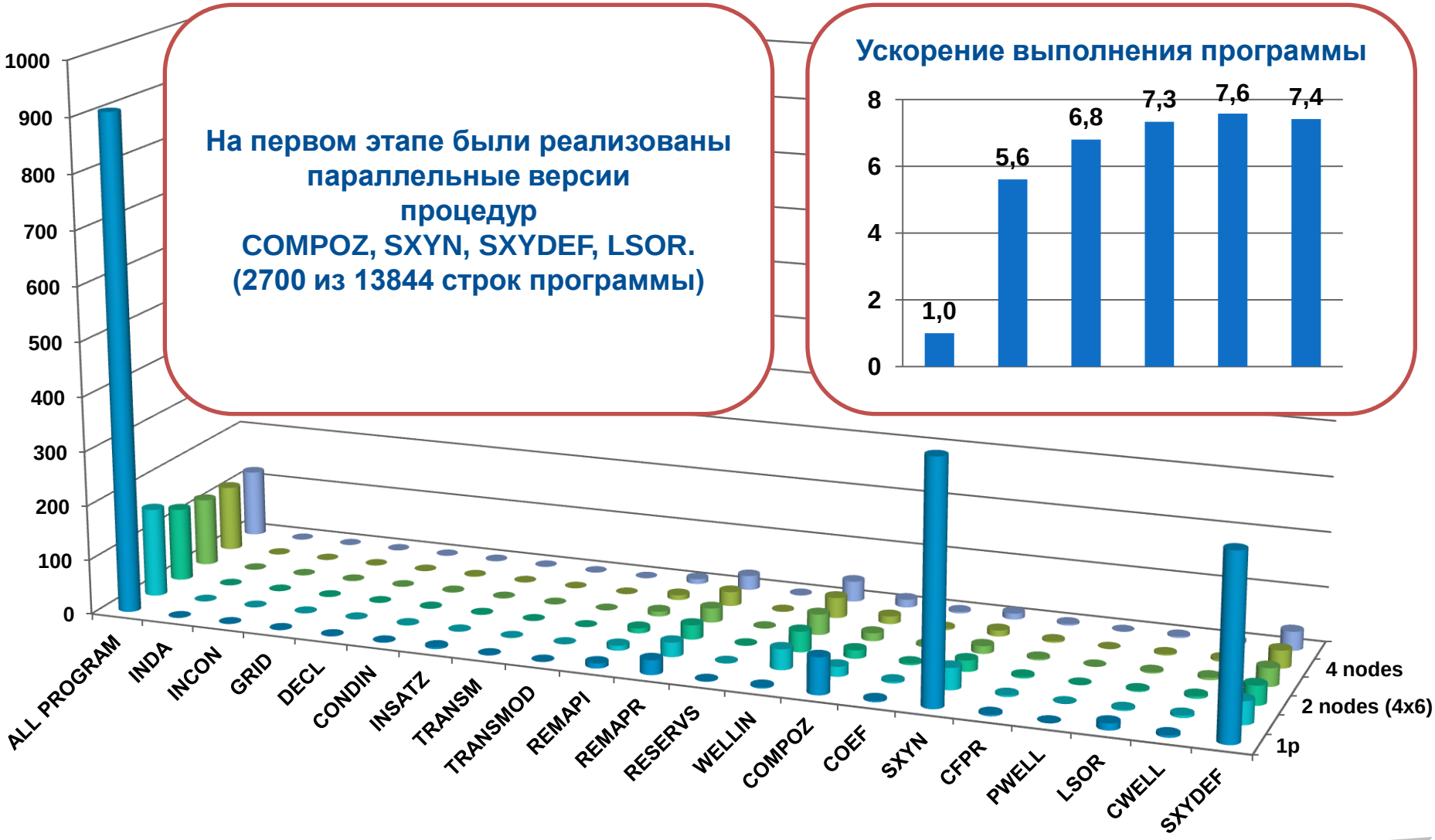
Времена выполнения различных процедур программы Композит



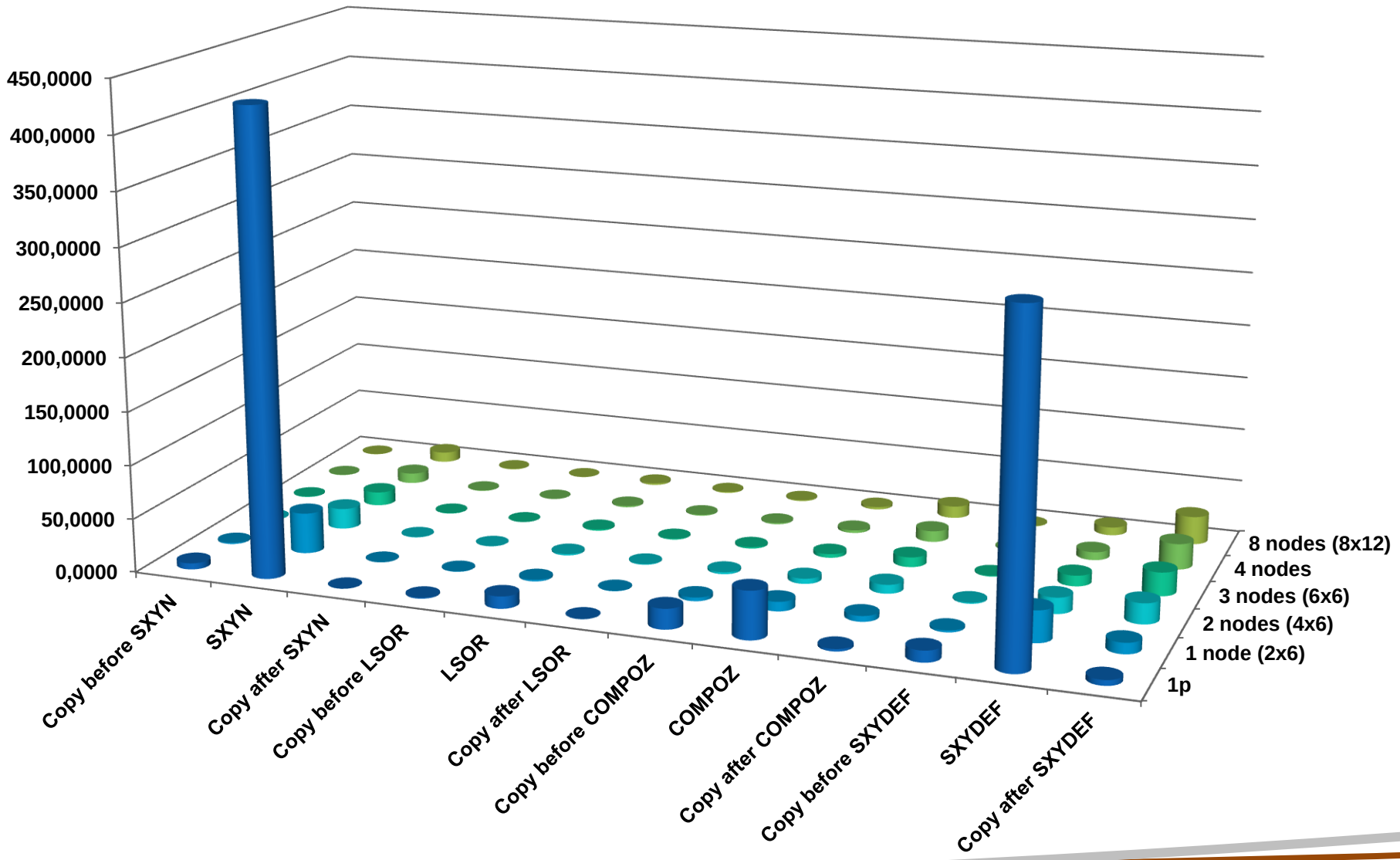
Распараллеливание времяемких фрагментов программы

- 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**

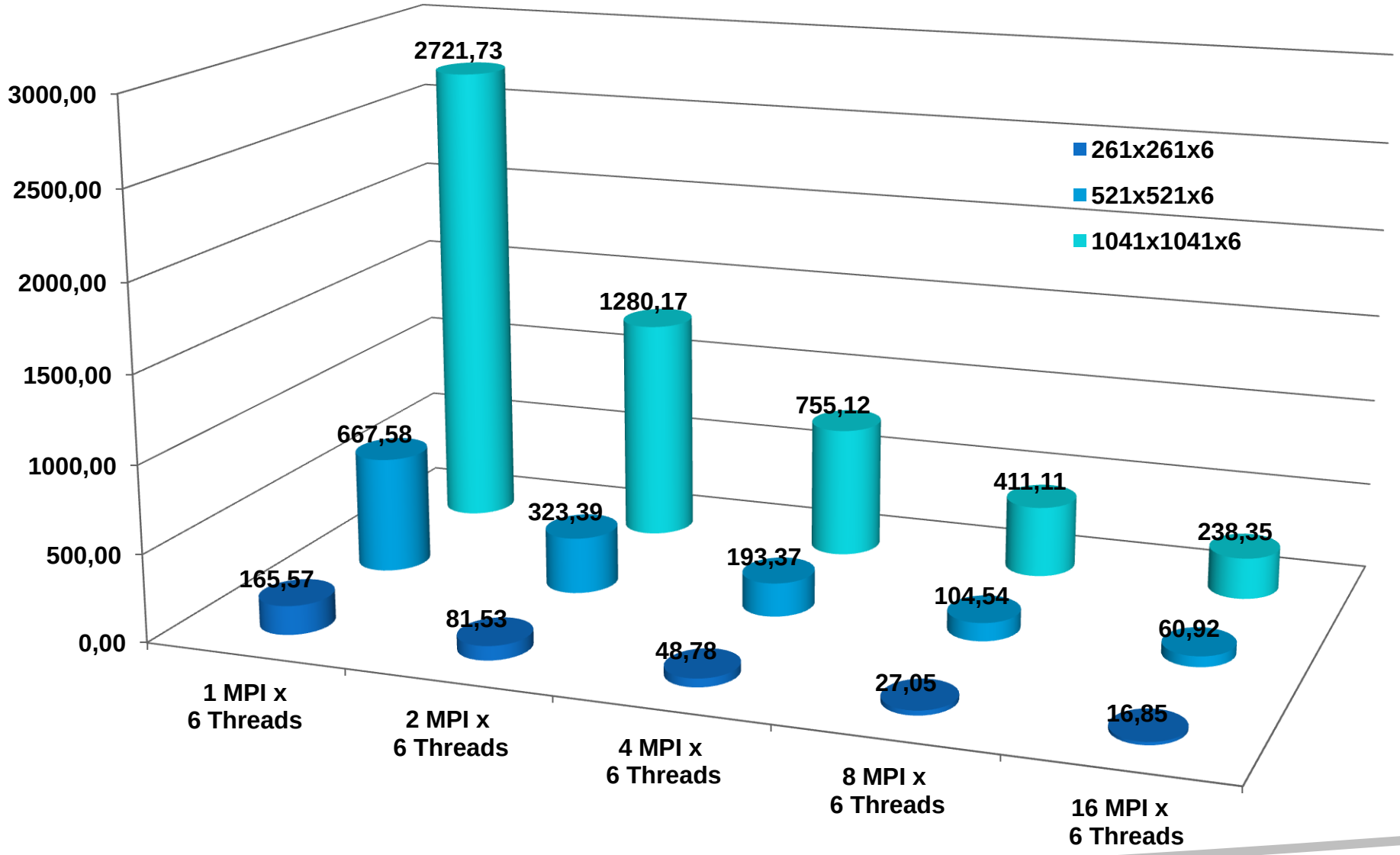
Времена выполнения частично распараллеленной программы на разном числе узлов K-100



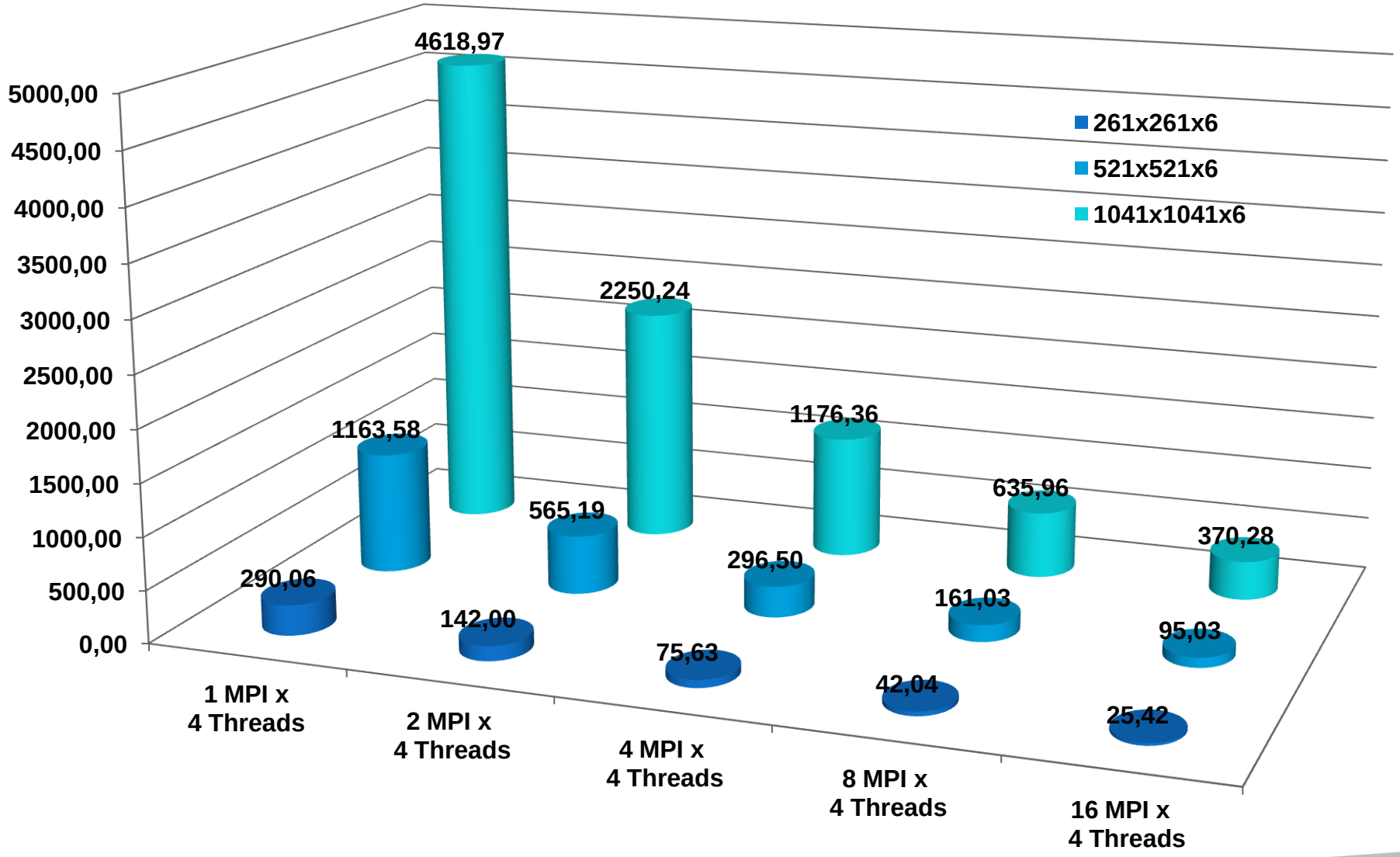
Накладные расходы на копирование данных между исходными массивами и их копиями



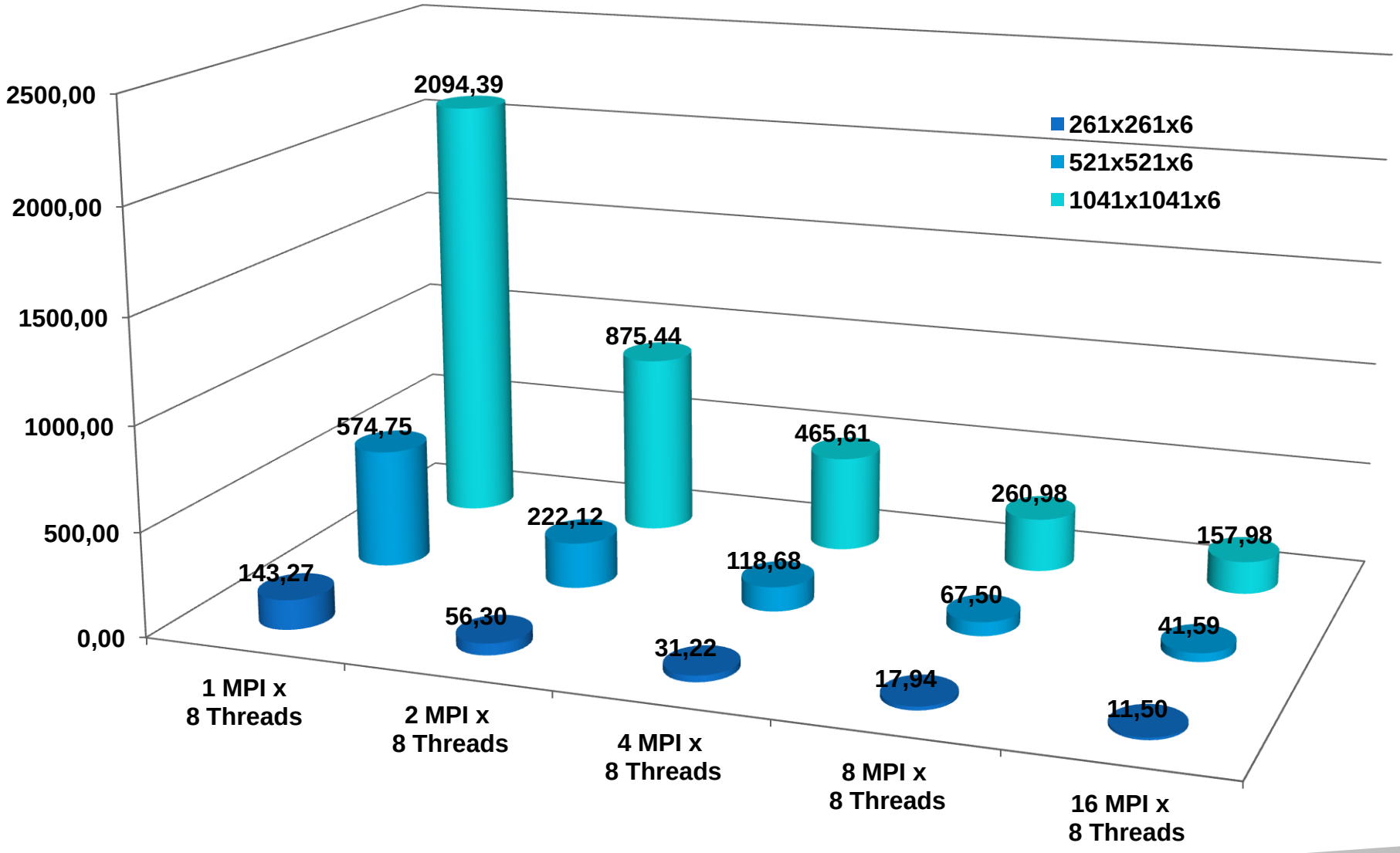
Времена выполнения 100 итераций программы Композит на суперкомпьютере К-100



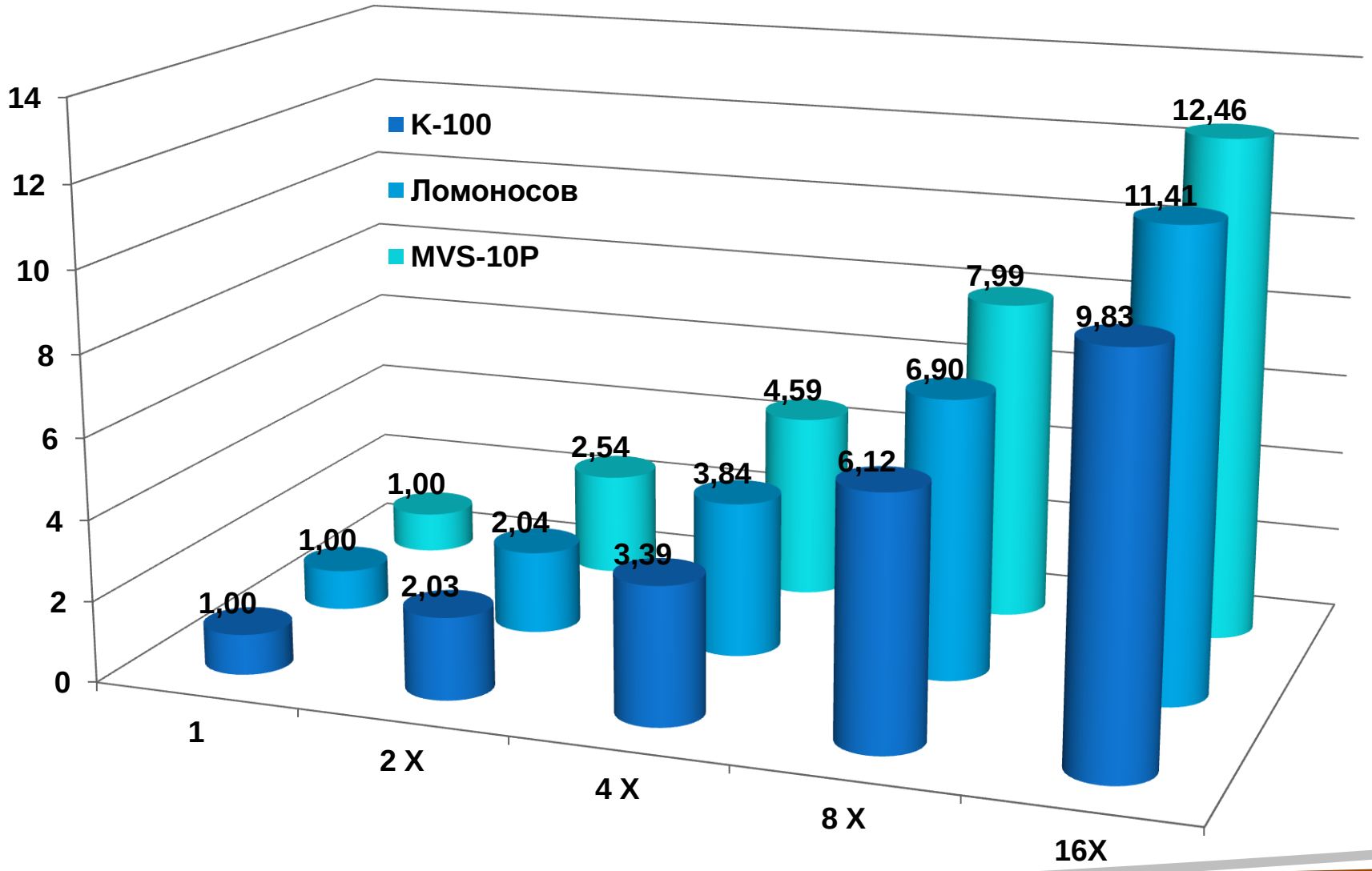
Времена выполнения 100 итераций программы Композит на суперкомпьютере Ломоносов



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

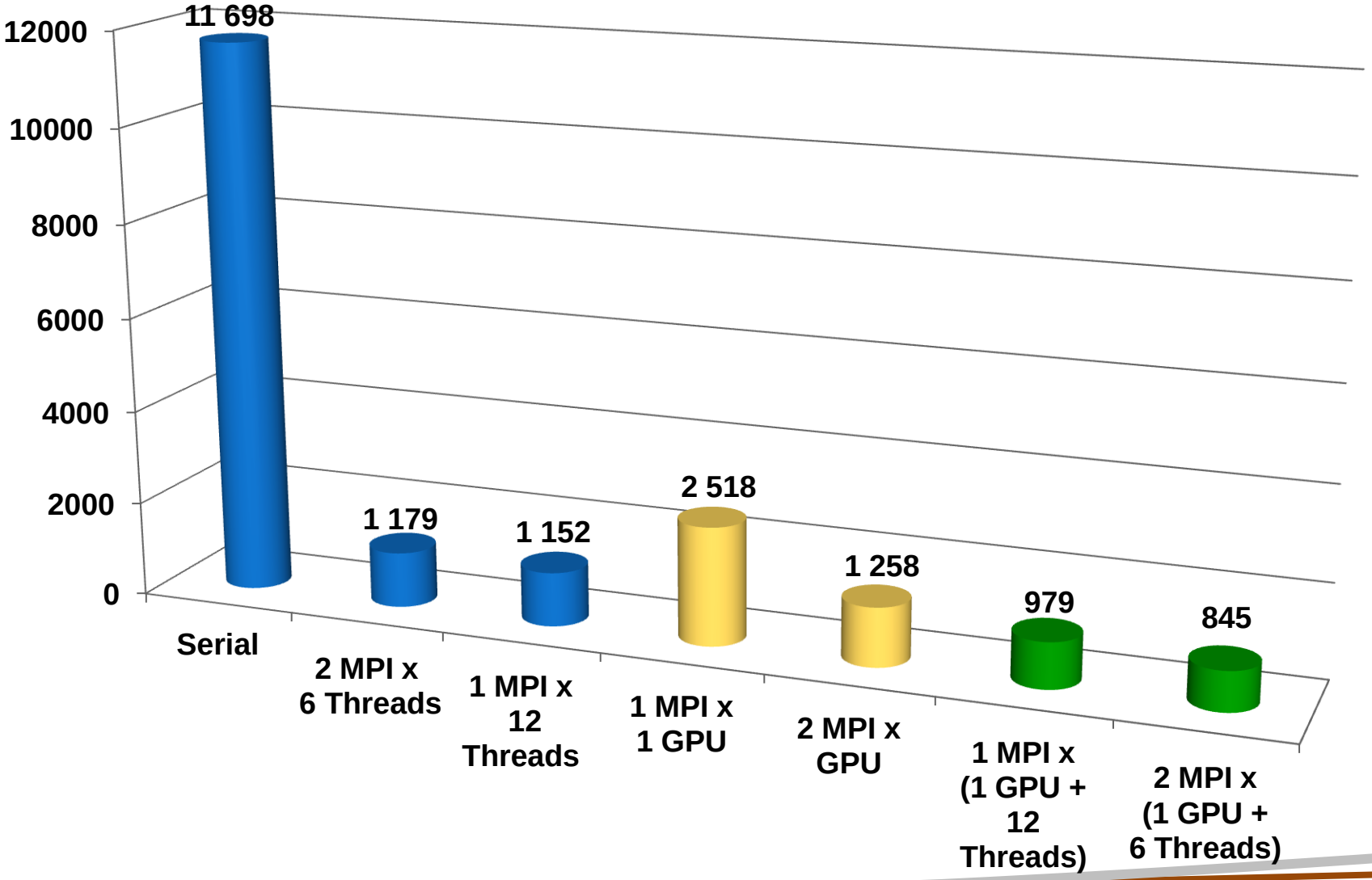


Ускорение выполнения 100 итераций программы Композит на сетке 261x261x6



Время работы программы Композит на сетке 261x261x6 на узле К-100

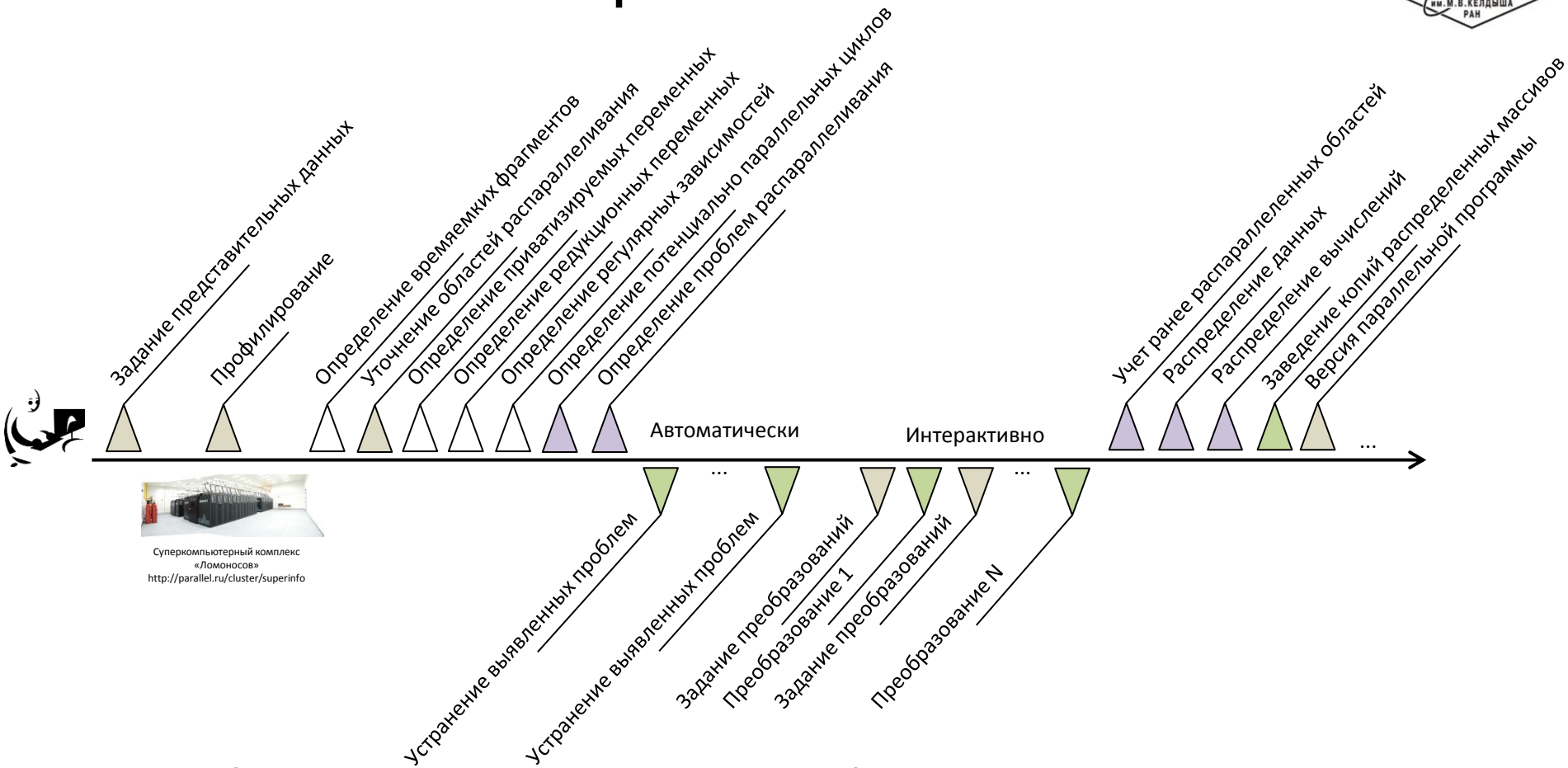
(2 x Intel Xeon X5670 + 2 x GPU NVIDIA Fermi C2050)



Система SAPFOR: 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

Система SAPFOR: развитие



Введены области распараллеливания для обеспечения инкрементального распараллеливания

Распараллеливание программы представляет собой последовательность проходов анализа и преобразований

Расширен класс распараллеливаемых программ за счет пересмотра алгоритмов распределения данных и вычислений

Система SAPFOR: развитие

Средства профилирования и выделения областей распараллеливания.

Средства поиска последовательностей преобразований для автоматического устранения проблем распараллеливания.

Средства автоматического выполнения наиболее часто встречающихся преобразований.

Средства интерактивного взаимодействия с пользователем в процессе распараллеливания.

Выводы

- DVM-система позволяет разрабатывать эффективные параллельные программы для гибридных вычислительных кластеров, использующих многоядерные универсальные процессоры, графические ускорители и сопроцессоры Intel Xeon Phi
- Наличие развитых средств для отладки и анализа производительности DVMH-программ существенно упрощает процесс разработки параллельных программ
- Релизация инкрементального распараллеливания, автоматизация выполнения наиболее часто встречающихся преобразований последовательных программ позволит использовать систему SAPFOR для распараллеливания больших программных комплексов

Спасибо за внимание

http://dvm-system.org
dvm@keldysh.ru