ВСЕРОССИЙСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ

НАУЧНЫЙ СЕВИС В СЕТИ ИНТЕРНЕТ

Поддержка интерактивности в системе САПФОР

19 сентября 2017 г. | Абрау-Дюрсо

Н.А. Катаев, А.С. Колганов, А..А. Смирнов

Институт прикладной математики им. М.В. Келдыша РАН

Факультет вычислительной математики и кибернетики МГУ им. М.В. Ломоносова

Автоматизация распараллеливания

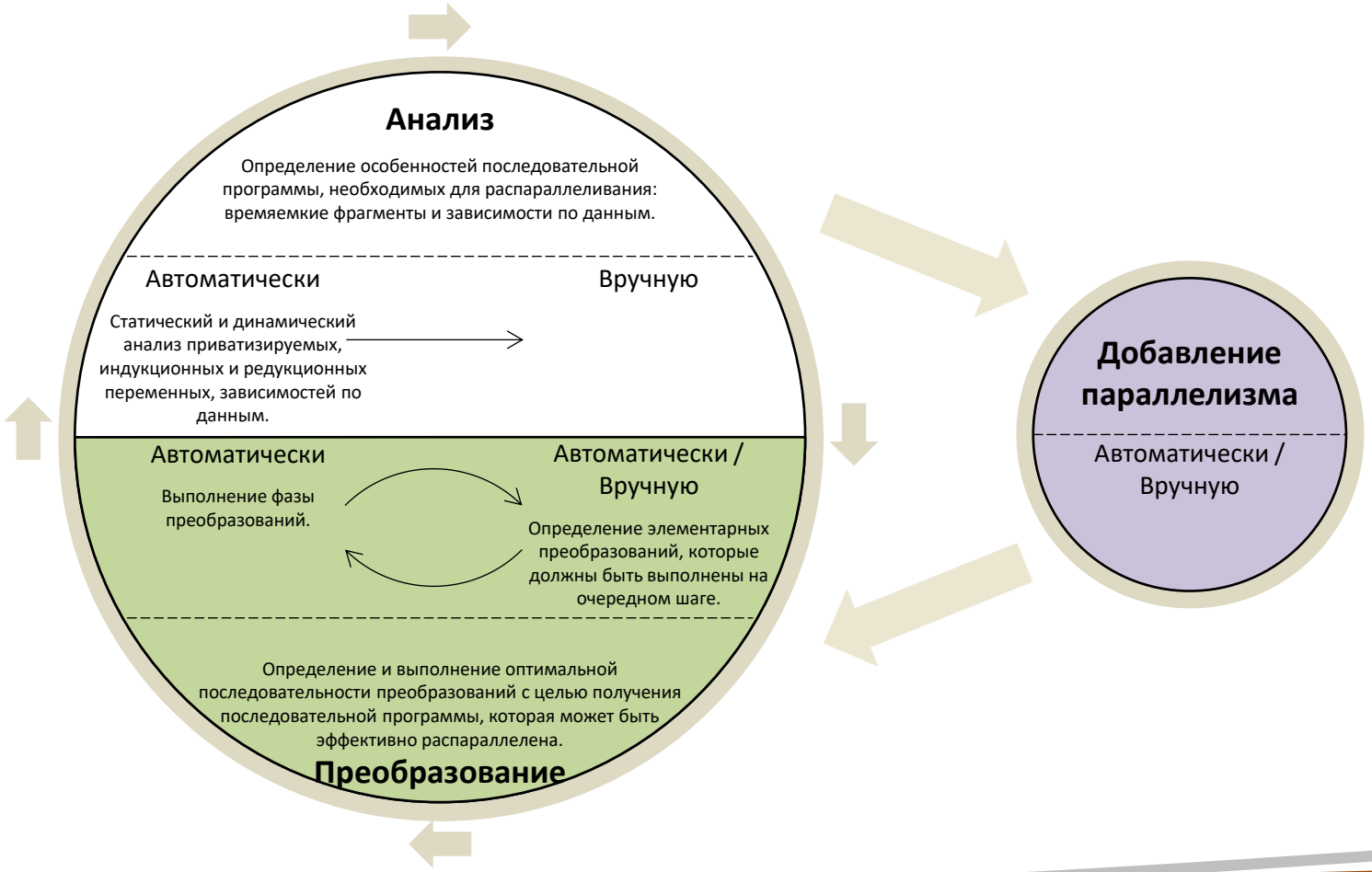
- Распараллеливание программ – процесс их адаптации для эффективного исполнения на вычислительной системе параллельной архитектуры. Как правило заключается либо в переписывании программ на языке параллельного программирования, либо в добавлении специальных указаний.

- Автоматическое распараллеливание – оптимизация программы компилятором, состоящая в автоматическом её преобразовании для эффективного выполнения на параллельном компьютере, например, на SMP или NUMA машине.

- Автоматизация распараллеливания – создание и использование инструментальных средств для распараллеливания программ.

Автоматизация распараллеливания: САПФОР

Ориентирована на инкрементальное, итерационное, интерактивное распараллеливание последовательных программ на многоядерные кластеры с ускорителями.



Взаимодействие с пользователем: цели

- Исследование информационной структуры распараллеливаемой программы, представление ее в виде различных абстракций, включающих текстовое и графовое представления.
- Задание свойств программы, знание которых необходимо для автоматического принятия решений по распараллеливанию, но которые не могут быть получены автоматически, например, из-за требования консервативности средств статического анализа.
- Ручной выбор преобразований программы, которые должны быть выполнены с целью устранения проблем, препятствующих распараллеливанию, если автоматически устранить данные проблемы не удается.
- Частичное распараллеливание программы пользователем, например, за счет выбора распределения данных или ручного распараллеливания отдельных фрагментов программы.

Исследование свойств программы

- Моделирование распространения упругих волн в средах со сложной 3D геометрией

  ~ 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,dcU
!DVM$& dcUz_dq3,dcWx_dq1,dcWx_dq2,dcWx_dq3,dSxz_dz,dSxy_dx,
!DVM$& dc2Ux_dq2,dc2Ux_dq3,dc2Vy_dq1,dc2Vy_dq2,dc2Vy_dq3,dc2
!DVM$& dc2Wz_dq2,dc2Wz_dq3,dSyy_dy,dcVz_dq1,dcVz_dq2,dcVz_
!DVM$& dcWy_dq1,dcWy_dq2,dcWy_dq3,dSyz_dz,dSxz_dx,dSyz_dy
!DVM$& dc3Ux_dq2,dc3Ux_dq3,dc3Vy_dq1,dc3Vy_dq3,dc3W
!DVM$& 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	...ables	Read variables
k=k_source-1,k_...		m_s_...	1.31.3	0	PRIV(i,j,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

Задание свойств программы

Предусмотрен набор указаний для описания свойств программы, которые не удалось определить на этапе статического анализа.

```
!$spf analysis reduction(sum(rsdnm))
    do  k__90 = 2,nz0 - 1
      do  j__91 = jst,jend
        do  i__92 = ist,iend
          do  m__89 = 1,5
            rsdnm(m__89) = rsdnm(m__89) + rsd(m__89,i__92,j__91,k_
  &_90) * rsd(m__89,i__92,j__91,k__90)
          enddo
        enddo
      enddo
    enddo
```

LU NAS NPB
v. 3.3

Спецификации свойств программы	
!$spf analysis independent(<var-list>)	Отсутствие межитерационных зависимостей по данным в цикле.
!$spf analysis (private(<var-list>) !$spf analysis firstprivate(<var-list>) !$spf analysis lastprivate(<var-list>)	Приватизируемые (по входу или по выходу) переменные в цикле.
!$spf analysis flow|anti|output <from> <to> [<len>]	Наличие зависимости по данным между двумя операторами.
!$spf analysis reduction (<op>(<var-list>))	Редукционные переменные для цикла.
…	

Поиск последовательности преобразований

– *автоматический выбор способа устранение проблемы*

Предусмотрен набор оптимизационных последовательностей, которые применяются, если указанная проблема удовлетворяет определенным критериям.

```
do j = 1,ny
  do i = 0,nx
    hy4 = hy(j+1)+2.d0*hy(j)+hy(j-1)
    ...
    Frox = ...
    ro1(i,j) = ro1(i,j) + Frox
    ro1(i+1,j) = ro1(i+1,j) - Frox
    E1(i,j) = E1(i,j) + FEx
    E1(i+1,j) = E1(i+1,j) - FEx
  enddo
enddo
```

⇨

```
do j = 1,ny
  do i = 0,nx
    hy4 = hy(j+1)+2.d0*hy(j)+hy(j-1)
    ...
    Frox = ...
    tmp1(i+1,j) = Frox
    ro1(i+1,j) = ro1(i+1,j) – Frox
    E1(i,j) = E1(i,j) + FEx
    E1(i+1,j) = E1(i+1,j) - FEx
  enddo
enddo

do i = 1,nx
  do j = 0, ny
    ro1(i,j) = ro1(i,j) + tmp1(i+1,j)
  enddo
enddo
```

«данные для записи витка ro1(i, j) и ro(i,j+1) находятся на разных процессорах»

Поиск последовательности преобразований

– *автоматический выбор способа устранение проблемы*

Предусмотрен набор оптимизационных последовательностей, которые применяются по критериям.

```
do j = 1,ny
  do i = 0,nx
    hy4 = hy(j+1)+
    ...
    Frox = ...
    ro1(i,j) = ro1(i
    ro1(i+1,j) = ro
    E1(i,j) = E1(i,j)
    E1(i+1,j) = E1(i
  enddo
enddo
```

«данные для
находятся на

Visualizer - [cavity_step_2]

File Project View Tools Panels Window Help

Project cavity_step_2
Source files
cavity_step_2.for

```
      Wy = tau*(rm*uxm*(uy(i+1,j)-uy(i,j))/h
     *     + (p(i+1,j+1)+p(i,j+1)-p(i+1,j-1)-p(i,j-1))/hy4
     *     + rm*uym*(uy(i+1,j+1)+uy(i,j+1)-uy(i+1,j-1)-uy(i,j-1))
     *     / hy4)
      Rs = tau*(uxm*(p(i+1,j)-p(i,j))/h+gm*pm*divu
     *     + uym*(p(i+1,j+1)+p(i,j+1)-p(i+1,j-1)-p(i,j-1))/hy4)
      Pxx = 2.d0*visc*(ux(i+1,j)-ux(i,j))/h-2.d0/3.d0*visc*divu
     *      + uxm*Wx+Rs
      Pxy = uxm*Wy + visc*((uy(i+1,j)-uy(i,j))/h
     *      + (ux(i+1,j+1)+ux(i,j+1)-ux(i+1,j-1)-ux(i,j-1))/hy4)
      Wx = ((ro(i+1,j)*ux(i+1,j)*ux(i+1,j)
     *     - ro(i,j)*ux(i,j)*ux(i,j)+p(i+1,j)-p(i,j))/h
     *     + (ro(i+1,j+1)*ux(i+1,j+1)*uy(i+1,j+1)
     *     + ro(i,j+1)*ux(i,j+1)*uy(i,j+1) - ro(i+1,j-1)*ux(i+1,j-1)
     *     * uy(i+1,j-1) - ro(i,j-1)*ux(i,j-1)*uy(i,j-1))/hy4)*tau
     *     / rm
      Fmx = rm*(uxm - Wx)
      Frox = - Fmx*hy(j)
      Ent = ((E(i+1,j)+E(i,j))*0.5d0 + pm)/rm
      FEx = (Pxx*uxm + Pxy*uym - Qx - Ent*Fmx)*hy(j)
      ro1(i,j) = ro1(i,j) + Frox
      ro1(i+1,j) = ro1(i+1,j) - Frox
      E1(i,j) = E1(i,j) + FEx
      E1(i+1,j) = E1(i+1,j) - FEx
    enddo
  enddo
c
```

Loop / PU	Progra...	File name	dvm_xp_0 Decisions	Write varia
i=0,nx	cavity	cavity_step_2.for (202)	часть гнезда, данные для записи витка ro1(i,j) и ro1(i+1,j) находятся на разных процессорах;	
i=0,nx				
j=0,ny	cavity	cavity_step_2.for (259)	часть гнезда, данные для записи витка ro1(i,j) и ro1(i,j+1) находятся на разных процессорах;	
j=0,ny				
i=0,nx	cavity	cavity_step_2.for (345)	часть гнезда, данные для записи витка ux1(i,j) и ux1(i+1,j) находятся на разных процессорах;	
i=0,nx				

Output Decisions

Ready Ln 202

Поиск последовательности преобразований

– *активное вовлечение пользователя в процесс выбора*

Система предоставляет пользователю набор элементарных преобразований, которые могут быть выполнены автоматически.

```
for (I = 1; I < Nx + 1; ++I)
  for (J = 1; J < Nz + 1; ++J)
    #pragma x unroll
    for (II = 0; II < 2; ++II)
      for (JJ = 0; JJ < 2; ++JJ) {
        …
        Flx[I][J - 1 + JJ] = Flx[I][J - 1 + JJ] - Iloc[0] * Fl[0];
        Flz[I - 1 + II][J] = Flz[I - 1 + II][J] - Jloc[1] * Fl[1];
      }
```

```
for (I = 1; I < Nx + 1; ++I)
  for (J = 1; J < Nz + 1; ++J) {
    …
    Flx[I][J - 1] = Flx[I][J - 1] - 1 * Fl[0];
    Flz[I - 1][J] = Flz[I - 1][J] - 1 * Fl[1];
    …
    Flx[I][J] = Flx[I][J] - 1 * Fl[0];
    Flz[I - 1][J] = Flz[I - 1][J] - -1 * Fl[1];
    …
    Flx[I][J - 1] = Flx[I][J - 1] - -1 * Fl[0];
    Flz[I][J] = Flz[I][J] - 1 * Fl[1];
    …
    Flx[I][J] = Flx[I][J] - -1 * Fl[0];
    Flz[I][J] = Flz[I][J] - -1 * Fl[1];
}
```

#pragma x unroll [(index-list)]

index ::= var : int-value

Двухмерная задача моделирования процессов плавления многокомпонентных порошков при селективном лазерном спекании на основе многокомпонентной и многофазной гидродинамической модели.

Поиск последовательности преобразований

– *активное вовлечение пользователя в процесс выбора*

Сис… …й,
кото…

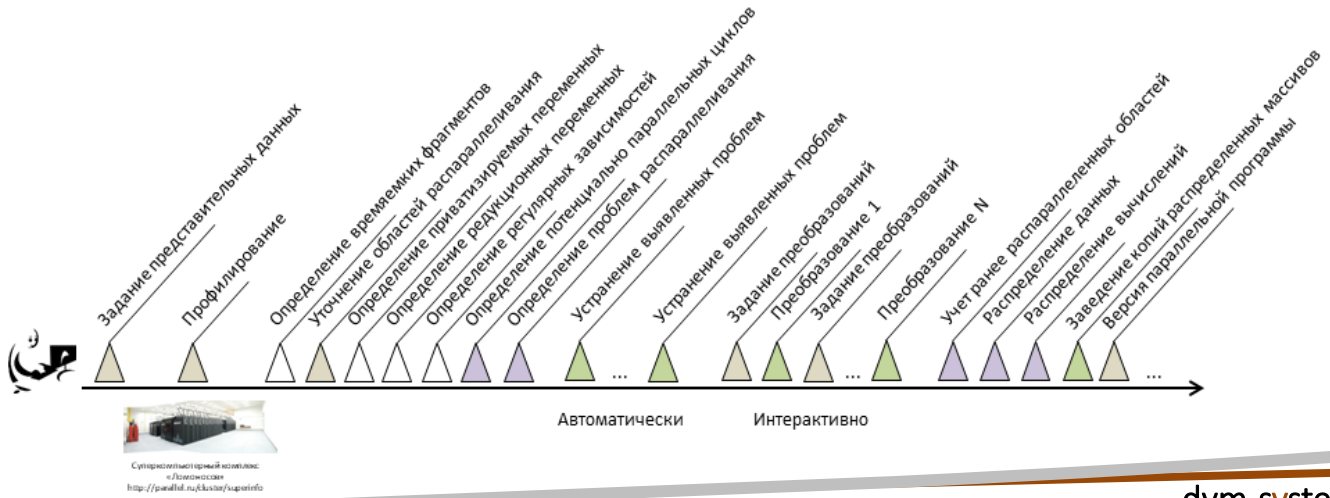
```
for (
 for (
  #p
  for
   fo
    }
```

Преобразования	Последовательность элементарных преобразований для программы Спекание 2D									Итого
	1	2	3	4	5	6	7	8	9	
Подстановка переменных	8									8
Вынесение инвариантов	1									1
Разворачивание циклов	5	1					4			10
Распределение циклов			1	2				2		5
Перестановка гнезд циклов					1				2	3
Объединение циклов					4				2	6
Перестановка циклов в гнезде										0
Сдвиг индекса						2				2
Итого	14	1	1	2	5	2	4	2	4	35

Двухмерная задача моделирования процессов плавления многокомпонентных порошков при селективном лазерном спекании на основе многокомпонентной и многофазной гидродинамической модели.

Инкрементальное распараллеливание

- Поэтапное распараллеливания, начиная с времяемких фрагментов.
- Постепенное добавления новых областей распараллеливания с сохранением уже найденных решений для ранее объявленных областей.
- Повышения качества распараллеливания выделенных областей за счет отказа от распараллеливания сложных фрагментов программы.
- Возможность ручного распараллеливания некоторых фрагментов программы и учет принятых программистом решений при распараллеливании других фрагментов.



Взаимодействие с пользователем: проблемы

- Отсутствие интеграции с существующими средами разработки.

  Отсутствие полноценной поддержки языков программирования и необходимость использования дополнительных приложений для написания кода программ.

- Использование директив для описания преобразований не применимо при распараллеливание программных комплексов.

  Последовательное применение различных преобразований может приводить к существенному росту объема программного кода, это, в свою очередь, усложняет расстановку директив без дополнительной визуальной поддержки.

- Отсутствие кроссплатформенности диалоговой оболочки.

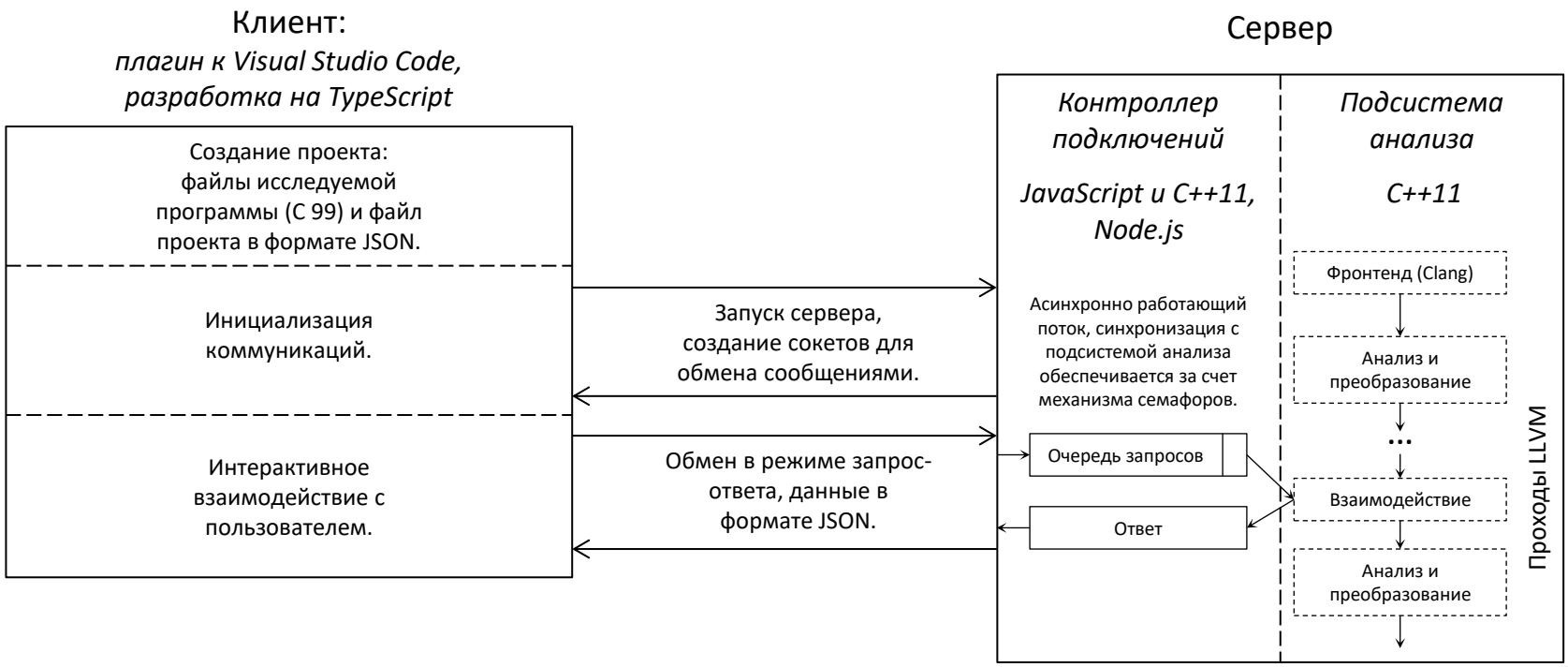
  Применение библиотеки MFC предопределило ориентированность САПФОР на вычислительные системы под управлением операционной системы Windows и невозможность установки разрабатываемых инструментов на большинство вычислительных кластеров.

Взаимодействие с пользователем: решения

- Клиент-серверное взаимодействие между диалоговой оболочкой и ядром системы (средствами анализа, преобразования, принятия решений по распределению данных и вычислений и организации межпроцессорных коммуникаций).
- Серверная часть может быть развернута как локально, на вычислительной системе пользователя, так и на удаленном сервере.
- Возможно использовать различные технологии программирования при разработке клиентской (C#, JavaScript, Java) и серверной частей (C, C++).
- Унифицированный интерфейс обмена сообщениями между клиентской и серверной частью позволяет разработать комплекс интерактивных сред (плагины для существующих систем, отдельное приложение, Web-сервис).

Апробация предложенного подхода

- Плагин для кроссплатформенного редактора Microsoft Visual Studio Code.
- Языки разработки: клиентская часть – TypeScript (JavaScript), сервер – C++.
- Взаимодействие с клиентской частью обеспечивается отдельным видом проходов ядра системы САПФОР, которые принимают запросы от диалоговой оболочки, а затем подготавливают и отправляют на них ответы, контролируют поведение системы.



Апробация предложенного подхода

Плагин для кроссплатформенного редактора Microsoft Visual Studio Code.

[Узел разработки расширения] — jac.c — Visual Studio Code

Файл Изменить Выбор Просмотреть Перейти Справка

jac.c

```c
#define max(A, B) ((A) > (B) ? (A) : (B))
#define abs(A) ((A) < 0 ? -(A) : (A))

#define L 16000
#define M 6250
#define ItMax 200

float MaxEps = 0.5f;
float A[L][M];
float B[L][M];

int main() {
  double TS, TE;
  int I, J, It;
  float Eps, Tmp;
  for (I = 0; I < L; I ++)
    for (J = 0; J < M; J ++) {
      A[I][J] = 0;
      if (I == 0 || J == 0 || I == L - 1 || J == L - 1)
        B[I][J] = 0;
      else
        B[I][J] = 3 + I + J ;
    }
  for (It = 1; It <= ItMax; ++It) {
    Eps = 0;
    for (I = 1; I < L - 1; ++I)
      for (J = 1; J < M - 1; ++J) {
        Tmp = abs(B[I][J] - A[I][J]);
        Eps = max(Tmp, Eps);
        A[I][J] = B[I][J];
      }
    for (I = 1; I < L - 1; ++I)
      for (J = 1; J < M - 1; ++J)
        B[I][J] = (A[I - 1][J] + A[I][J - 1] + A[I][J + 1] + A[I + 1][J]) / 4.0f;
    if (Eps < MaxEps )
      break ;
  }
  return 0;
}
```

Перейти к определению	F12
Показать определение	Alt+F12
Изменить все вхождения	Ctrl+F2
Форматировать документ	Alt+Shift+F
TSAR: Analyze file	
TSAR: Close tsar session	
Вырезать	Ctrl+X
Копировать	Ctrl+C
Вставить	Ctrl+V

TSAR Advisor | jac.c

Analysis result summary for jac.c

Total number of analyzed files is 1 including:

1 Source File

Analyzed files comprise 1 function with 10 variables and 7 loops.

The following loop traits have been explored (total 42):

8 shared variables
9 private variables
6 first private variables
0 last private variables
6 dynamic private variables
0 reduction variables
0 induction variables
13 unclassified dependencies
0 address accesses

0 0 17 c | jac.c Строка 19, столбец 40 Пробелов: 2 UTF-8 CRLF C

Заключение

- Развитие системы САПФОР в направлении распараллеливания крупных программных комплексов потребовало пересмотра подходов к организации взаимодействия с пользователем.
- Для обеспечения возможности инкрементального итерационного распараллеливания и поддержания диалога с пользователем был предложен и опробован подход на основе клиент-серверное архитектуры.
- Предложенный подход может быть использован для организации удаленного распараллеливания и предоставления функциональности системы как облачного сервиса, доступ к которому будет организован по сети Интернет.

Спасибо за внимание

http://dvm-system.org