

Project of Parallel Computer Algebra

G.I.Malaschonok

e-mail: malaschonok@ya.ru

Tambov State University

Программа «Университетский кластер»

Конференция "Облачные вычисления: образование, научные
исследования, разработки"

Москва, РАН, 15-16 апреля, 2010

16.04.2008

Параллельные программы поддерживают
численные вычисления.

Современная математика - это символьная математика,
элементарными объектами выступают не числа, а функции,
операторы, в частности, функциональные матрицы и другие
объекты.

Параллельные программы должны поддерживать
символьно-численные вычисления.

Что имеется сегодня в Intel MKL:

Что имеется сегодня в Intel MKL:

- линейная алгебра: плотные матрицы (BLAS, LAPACK, ScaLAPACK);
- линейная алгебра: разреженные матрицы (прямые/итеративные решатели, Sparse BLAS);
- быстрое Преобразование Фурье (SMP и MPI);
- векторная математическая библиотека (VML);
- векторная статистическая библиотека (VSL);
- функции расширенной точности (GMP);
- методы оптимизации (TRS).

Все это численные методы.

Естественным продолжением Math Kernel Library должна стать Math Symbolic Library - библиотека "пакетов параллельной символьной математики"(MSL).

текущие задачи проекта параллельной компьютерной алгебры

Мы занимаемся следующими символьно-численными вычислениями:

- вычисления с полиномами многих переменных,
- вычисление с символьными композициями элементарных функций,
- символьное интегрирование и дифференцирование,
- тригонометрические, показательные и логарифмические упрощения,
- решение некоторых классов дифференциальных уравнений и их систем в символьном виде,
- FFT для полиномов и целых чисел и алгоритмы умножения на основе FFT,
- вычисления с матрицами над полиномами многих переменных и над композициями функций,
- вычисление, обратной, присоединенной и эшелонной матрицы, ядра линейного оператора,
- вычисление характеристического полинома, определителя, спектра оператора.

The project Parallel Computer Algebra have been started in Tambov University in 2003 in collaboration with A.I.Avetisyan group of ISP RAS.

We carried out experiments on cluster of ISP RAS, then on Myrinet-IBM cluster of Tambov University and 8000-processors cluster of Joint Supercomputer Center (JSCC RAS).

In 2008 we have started the ParCA-2 project [1] and this year we plan to realize the main part of ParCA-3 system.

Structure of ParCA-3 system

Three parts of the ParCA-3 system are Front End, Kernel and Parallel part. Each part consists of several packages.

(1) The Front End part consists of two packages: *frontend* and *showGraph*. The *frontend* package is a collection of classes which support the low-level notebook programming for users of ParCA system. The *showGraph* package is devoted to the visualization of functions,

(2) The Kernel part consists of four packages: *number*, *polynom*, *func*, and *matrix*. These packages support corresponding mathematical objects: numbers, polynomials, functions and matrices.

(3) The Parallel part consists of two packages: *parjava* and *parca*. The *parjava* package is a Java-MPI interface. The *parca* package consists of all parallel programmes of the ParCA system.

The computational paradigm

1. Principles of algebraic structures

1. Each mathematical object is an element of some algebra (algebraic structure).
2. Each algebra is represented by some class in corresponding package. Therefore each element of algebra may be defined by the value of *dynamical fields of their class* and each operation of this algebra is represented by some *method of their class*.
3. In the top of the algebraic hierarchy we put some abstract algebra and call it “*element*”. The algebraic hierarchy is represented by the corresponding *class hierarchy* with *Element class* in the top. Each class which represents some real algebraic structure has to extend this abstract class of *Elements*. There are no dynamical fields in this class but all possible operations for the main types of usually used algebraic structures with one and two binary operations must be defined in this class of *Elements*.

The computational paradigm

II. Principles of working context

1. The direct product of several numerical algebras we will call an algebraic space. For example we can say about the follow $\mathbb{R}^n \mathbb{C}^m \mathbb{Z}^k$ algebraic space.
2. To carry out some symbolic-numerical computations we have to fix the algebraic space. A set of mathematical expressions which are defined on one algebraic space we call the *page*. A set of pages we call a *notebook*.
3. Page is a class. To create some page, or more correct, to create an instance of the page class, we have to fix some algebraic space. All expressions at the one page are considered as functions on this space.
3. To map this algebraic space V_1 into another algebraic space V_2 we have to make the map from this page into another page, which has it algebraic space V_2 . We may perform this map in another way. We have to fix the direct product $V_1 \times V_2$ as a algebraic space of page and then we may take any map from V_1 to V_2 in the same page.
4. To fix the algebraic space we must define the set of numbers and the names of variables, for example: $V_1 = \{u, v \in \mathbb{C}, x, y, z \in \mathbb{R}, n, m \in \mathbb{Z}\}$.

Current algebraic objects

There are defined standard **numerical algebras**: Z , Q , Cz , Cq , Zp and $Zp32$. We denote $Cz = \{a + ib : a, b \in Z, i^2 = -1\}$ and $Cq = \{a + ib : a, b \in Q, i^2 = -1\}$, $Zp = Z/pZ$, with arbitrary prime number $p \in Z$ and $Zp32 = Z/pZ$, with prime number p less than maximal positive number of 32-bit word.

Several approximate numerical algebras are defined for rounded sets of numbers: R , C , $R64$, $C64$, $R128$ and $C128$. The $R64$ is a standard 64 bit-length floating point real number, R is an arbitrary-length real number, $R128$ is a pair of a 64 bit-length floating point real number and an additional 64 bit-length word for an order. Three complex classes C , $C64$ and $C128$ are obtained from the classes R , $R64$ and $R128$, consequently.

Current algebraic objects

The **polynomial package** consists of the classes of polynomials (polynom) of many variables, rationals (Rational), rationals in factoring form (FactorPol), and sum of factoring rationals (FactorPolSum). The class *Ring* of polynomial package defines the algebraic space in the form of polynomial ring. The algebraic space

$V_1 = \{u, v \in \mathbb{C}, x, y, z \in \mathbb{R}, n, m \in \mathbb{Z}\}$ is defined by the ring $R_1 = \mathbb{C}[u, v]\mathbb{R}[x, y, z]\mathbb{Z}[n, m]$.

The **func package** contains the class of functions (F). Each function of this class contains two dynamical fields: *name* – for title of this function and *X* – for the array of elements, which are the arguments of this function. Therefore this class permits to construct any composition of transcendental and rational functions.

The matrix package

The **matrix package** contains the classes of dense and sparse matrices in two forms: in-core and out-of-core. The out-of-core form of matrices is necessary for very large matrices, which cannot be disposed in core. The main matrix methods are: “adjoint”, “det”, “kernel”, “toEchelonForm”, “inverse”.

Computation algorithms for symbolic-numerical matrices are the most interesting objects for parallelization.

The following matrix algorithms were obtained in parallel form:

1. Matrix multiplication.
2. Computation of adjoint, inverse and echelon form of matrix.
3. Computation of determinant and kernel.
4. Computation of matrix characteristic polynomial.
5. Computation of adjoint matrix for the polynomial matrix on the base of polynomial FFT algorithm.
6. Solving system of linear equations in the fraction field of integer number on the base of p-adic lifting.

The polynomial algorithms

The most complicated and important problem among the polynomial problems is the Groebner basis computation.

Therefore the next polynomial algorithms were obtained in parallel form:

1. Polynomial multiplication.
2. Computation of the Groebner basis for given ideal.

Would be nice to do parallel algorithms of polynomial GCD later

On-line access to the website with ParCA

The universities and research institutes of the Academy of Sciences which participated in the program "Universitetski Cluster" will have on-line access to the Cluster with ParCA System. Any browser may be used for this access.

First experiments showed that the system demonstrates different scalability for different problem sizes. So for the best efficiency the number of processors has to be chosen according to the problem size. Future experiments let us make a table which shows how many processors are needed for the defined problems.

Выполнение операций в параллельной системе компьютерной алгебры

Интерфейс приложения *ParCA* реализован как Web-приложение. В рамках этого проекта собраны известные Web технологии, такие как:

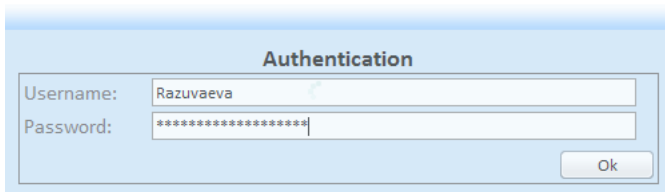
- 1 JavaScript — для включения кода в Web-приложение;
- 2 Каскадные таблицы стилей CSS — для определения стилей элементов Web-страницы;
- 3 Java Servlet — выполнение сценариев сервера;
- 4 DOM — доступ к объектам интерфейса;
- 5 XMLHttpRequest — для асинхронного взаимодействия с сервером;
- 6 XML;
- 7 Ajax.

В роли Web сервера используется Apache Tomcat. В качестве СУБД выступает MySQL.

Система *ParCA* обеспечивает управление запуском задач на кластерных системах. Предусмотрены следующие инструменты:

- 1 запуск задачи с формы,
- 2 запуск задачи из файла загруженного на сервер,
- 3 получение результата в режиме реального времени,
- 4 управление очередью запущенных задач,
- 5 управление результатами,
- 6 загрузка файлов на сервер,
- 7 управление файлами, загруженными на сервер.

Прежде чем приступить к работе, необходимо войти в систему. Для этого нужно ввести в поля *Username* и *Password* ваши данные аутентификации, как показано на рисунке ниже.



The image shows a standard Windows-style authentication dialog box. The title bar reads "Authentication". Inside the dialog, there are two text input fields. The first is labeled "Username:" and contains the text "Razuvaeva". The second is labeled "Password:" and contains a series of asterisks "*****" to mask the password. A small cursor is visible at the end of the password field. At the bottom right of the dialog, there is a button labeled "Ok".

Чтобы подтвердить корректность введенных данных, нажмите *Ok*.

Для запуска задач следует выполнить следующие шаги:

- 1 Заполнение главной формы.
- 2 Заполнение формы параметров запуска задачи.
- 3 Запуск задачи.

Запуск задачи можно осуществить двумя способами: запуск задачи с формы и запуск задачи из файла. В первом случае следует ввести тип задачи и текст задачи (входные данные) на главную форму, как показано на рисунке ниже.

```
In:gbasis
xyzu -1
yzu +xzu +xyu +xyz
zu +xu +xz +xy
u +z +y +x
Out:
```

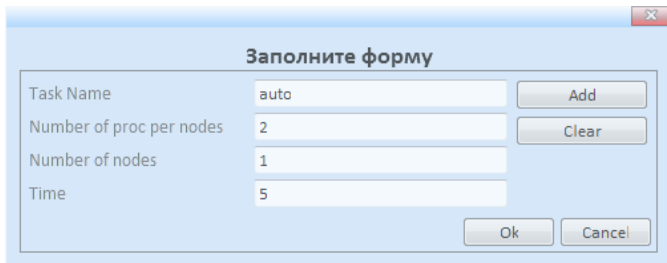
На данный момент пользователь может ввести одну из восьми типов задач:

- 1 `gbasis` — вычисление базиса Гребнера;
- 2 `adjoint` — вычисление присоединенной матрицы;
- 3 `adjointDet` — вычисление присоединенной матрицы и определителя матрицы;
- 4 `echelonForm` — вычисление ступенчатого вида матрицы;
- 5 `inverse` — вычисление обратной матрицы;
- 6 `det` — вычисление определителя матрицы;
- 7 `kernel` — вычисление ядра линейного оператора;
- 8 `charPol` — вычисление характеристического полинома.

Для запуска задачи из файла нужно в поле главной формы ввести имя существующего файла на сервере в виде следующей строки:
filename=имя файла.

Система *ParCA* дает возможность запуска нескольких задач с одной формы.

Далее следует заполнить форму параметров запуска задачи.



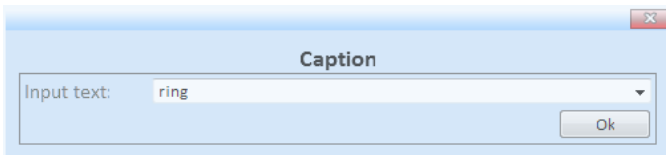
The image shows a dialog box with a light blue background and a title bar containing a close button (X). The title of the dialog is "Заполните форму" (Fill the form). Inside the dialog, there are four rows of input fields:

Task Name	auto	Add
Number of proc per nodes	2	Clear
Number of nodes	1	
Time	5	

At the bottom right of the dialog, there are two buttons: "Ok" and "Cancel".

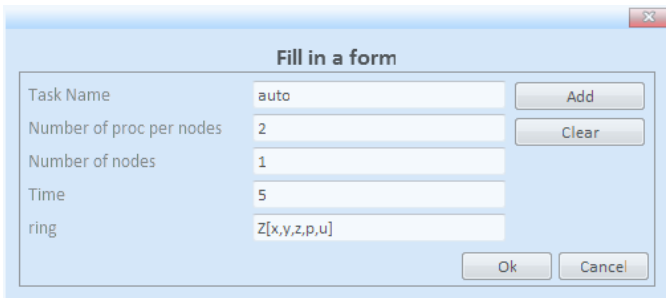
Если в задачу нужно передать дополнительные параметры, то выполняем следующие шаги:

- 1 нажимаем кнопку *Add*.
- 2 В появившемся окне вводим имя параметра:



A screenshot of a Windows-style dialog box titled "Caption". It features a single text input field with the text "ring" entered. To the right of the input field is a small downward-pointing arrow. Below the input field is a button labeled "Ok". The dialog box has a standard title bar with a close button (X) in the top right corner.

- 3 Подтверждаем имя параметра нажатием кнопки *Ok*.
- 4 В обновившейся форме вводим значение нового параметра:

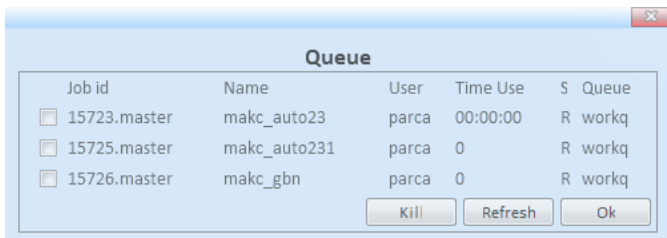


A screenshot of a Windows-style dialog box titled "Fill in a form". It contains several input fields and buttons. The fields are labeled as follows:

Task Name	auto	Add
Number of proc per nodes	2	Clear
Number of nodes	1	
Time	5	
ring	Z[x,y,z,p,u]	

At the bottom right of the form area are two buttons: "Ok" and "Cancel". The dialog box has a standard title bar with a close button (X) in the top right corner.

Для просмотра статуса запущенных пользователем задач, выберите в меню *Queue*. Перед вами появится форма, показанная на рисунке ниже:



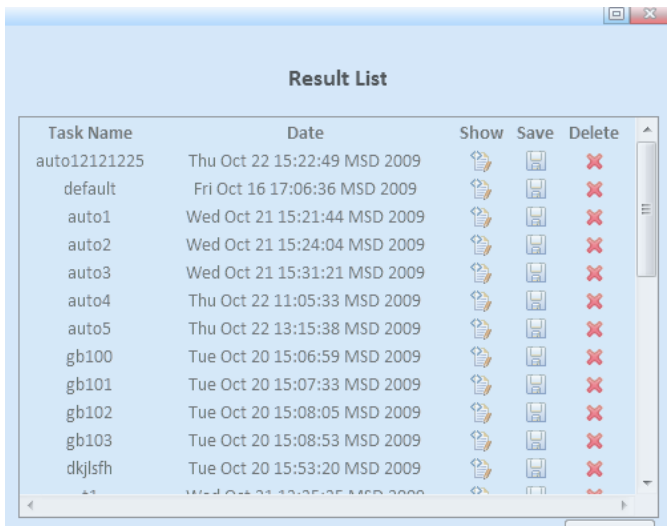
Эта форма дает возможность пользователю выполнить две функции:

- 1 предоставить актуальную на текущий момент информацию о всех запущенных пользователем и работающих или находящихся в очереди задачах;
- 2 удалить одну или несколько задач из очереди.

Для удаления задач из очереди выделите в левой колонке те, которые следует удалить и нажмите кнопку *Kill*.

Просмотр списка результатов

Для просмотра списка результатов вычислений выберите в меню раздел *Show Result*. Перед вами появится форма, показанная на рисунке ниже:

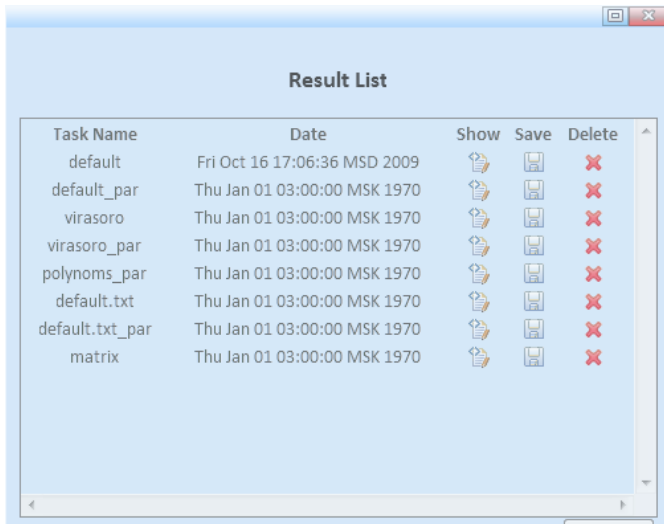


The screenshot shows a window titled "Result List" containing a table of task execution results. The table has five columns: "Task Name", "Date", "Show", "Save", and "Delete". Each row represents a task, and the "Date" column shows the completion date and time in MSD 2009 format. The "Show", "Save", and "Delete" columns contain icons for each respective action.

Task Name	Date	Show	Save	Delete
auto12121225	Thu Oct 22 15:22:49 MSD 2009			
default	Fri Oct 16 17:06:36 MSD 2009			
auto1	Wed Oct 21 15:21:44 MSD 2009			
auto2	Wed Oct 21 15:24:04 MSD 2009			
auto3	Wed Oct 21 15:31:21 MSD 2009			
auto4	Thu Oct 22 11:05:33 MSD 2009			
auto5	Thu Oct 22 13:15:38 MSD 2009			
gb100	Tue Oct 20 15:06:59 MSD 2009			
gb101	Tue Oct 20 15:07:33 MSD 2009			
gb102	Tue Oct 20 15:08:05 MSD 2009			
gb103	Tue Oct 20 15:08:53 MSD 2009			
dkjlsfh	Tue Oct 20 15:53:20 MSD 2009			
...

Просмотр списка файлов

Для просмотра списка файлов загруженных вами на сервер выберите в меню пункт *Show Files*. Перед вами появится форма, показанная на рисунке ниже:



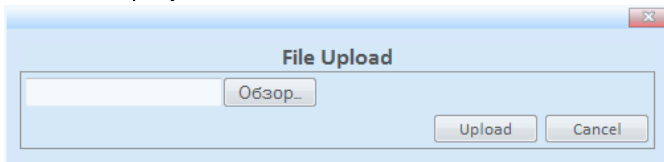
The screenshot shows a window titled "Result List" with a table of task entries. Each entry includes a task name, a date and time stamp, and three action buttons: "Show", "Save", and "Delete".

Task Name	Date	Show	Save	Delete
default	Fri Oct 16 17:06:36 MSD 2009			
default_par	Thu Jan 01 03:00:00 MSK 1970			
virasoro	Thu Jan 01 03:00:00 MSK 1970			
virasoro_par	Thu Jan 01 03:00:00 MSK 1970			
polynoms_par	Thu Jan 01 03:00:00 MSK 1970			
default.txt	Thu Jan 01 03:00:00 MSK 1970			
default.txt_par	Thu Jan 01 03:00:00 MSK 1970			
matrix	Thu Jan 01 03:00:00 MSK 1970			

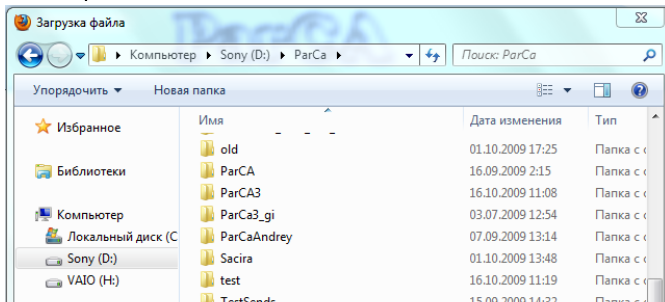
Загрузка файла на сервер

Для загрузки файла на сервер нужно выполнить следующие шаги:

- 1 Выберите на панели *Upload File*. Перед вами появится форма, показанная на рисунке ниже:






- 2 Нажмите кнопку Обзор. Перед вами появится диалоговое окно открытия файла.



```
In:adjointDet
[[2,7,0,1],
 [2,0,9,1],
 [8,7,2,8],
 [9,1,2,7]]
Out: find bound level=2
default type_element NumberZ
PARALLEL
-----
Number of processors = 2
Size of matrix = 4
boundLev = 2
disp process pause = 10000
adjoint=
[[-357, 2, 418, -427 ],
 [-138, -6, -45, 72 ],
 [27, -139, -35, 56 ],
 [471, 38, -521, 350 ]]

det=-1209
Time parallel = 1190 milliseconds
```

-  Malaschonok G.I. Classes structure of ParCA-2 system. International conference Polynomial Computer Algebra. St. Petersburg, PDMI RAS, 2009, P. 66-68.
-  Malaschonok G.I., Avetisan A.I., Valeev Yu.D., Zuev M.S. Parallel algorithmes of computer algebra. Proceeding of the institute of system programming, 2004., V.8, part. 2. P.169-180.
-  Malaschonok G.I. In the Direction of Parallel Computer Algebra System. Computer Science and Information Technologies. Proc. Conf. (Sept.19-23, 2005. Acad. Sci. of Armenia.) Yerevan, 2005, 451-453.

Invitation

International Conference "Parallel Computer Algebra"
29 June - 3 July 2010
Tambov, Tambov State University,

Conference Co-Chairs:
Gennadi Malaschonok, Gene Cooperman, Stephen Watt

Extended deadline for paper submission - 21 April 2010

URL <http://parca2010.parallel-computer-algebra.org>
or
<http://tsutmb.ru/imfi/parca>