Architecture of ParCA3 project

G.I.Malaschonok

Abstract. The main features of current version of parallel computer algebra system developed in Tambov university are presented.

1. Introduction

The project Parallel Computer Algebra (ParCA) have been started in Tambov University in 2003. The current version of the project is the version number 3. The main features of current version of ParCA system are presented in this talk.

We carried out experiments on Myrinet-IBM cluster of Tambov University and 4000-processors cluster of Russian Academy of Sciences Joint Supercomputer Center (JSCC).

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

2. 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.

This work was completed with the support of the Sci. Program "Devel. Sci. Potent. High. School", RNP 2.1.1/1853.

(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.

3. The computational paradigm

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*.

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: $V_1 = \{u, v \in \mathbb{C}, x, y, z \in \mathbb{R}, n, m \in \mathbb{Z}\}$.

4. 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 ∈Z and $Zp32 = Z/pZ$, with prime number p less than maximal positive number of 32-bite 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 pare 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.

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 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 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.

5. On-line access to the website with ParCA

The universities and research intitutes of the Academy of Scienses wich participated in the programm ”Universitetski Cluster” will have on-line access to the Cluster with ParCA System. Any browser may be used for this access.

6. Conclusion

First experiments showed that system demonstrates different scalability for different problem size. So for the best efficiency the number of processors have 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.

References

[1] Malaschonok G.I. Classes structure of ParCA-2 system. International conference Polynomial Computer Algebra. St. Petersburg, PDMI RAS, 2009, P. 66-68.

[2] 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.

[3] 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.

G.I.Malaschonok
Tambov State University
Internatsionalnaya, 33
392000 Tambov
Russia
e-mail: `malschonok@ya.ru`