

Network version of the computer algebra system bergman

S. Cojocaru A. Colesnicov L. Malahova

Abstract

The article describes problems arising at presentation of the computer algebra system bergman in the Web, and their solution using client-server approach.

1 General description of the problem

Bergman [1] is a package of symbolic calculations that operates over commutative and non-commutative graded algebras and graded modules over them. Bergman calculates Gröbner basis and provides some facilities to calculate appropriate invariants of the algebra and modules, such as the Hilbert series, and (in the non-commutative case only) the Poincarè series, Anick's resolution and the Betti numbers. Calculations can be performed in different polynomial rings. Field of coefficients, ordering and weights of variables, etc., can be varied in addition to commutativity and non-commutativity. Bergman is an efficient system, but can meanwhile use enormous resources of processor time and memory depending of problem to be resolved. Calculations can take hours of time and produce hundreds Mbytes of results.

Bergman is mainly developed by Prof. Jörgen Backelin (Stockholm University) in Portable Standard Lisp (PSL) targeting to efficiency and portability. Bergman is free but PSL is a commercial product. Now an experimental version of bergman is developed in Common Lisp and is distributed free by FTP. This version is less effective than PSL bergman but has the advantage of being free. To use bergman is therefore necessary to have pre-installed PSL or Common Lisp. Installation of both

Lisp systems is not a trivial task and is the competence of system administrator. If a researcher or an engineer is interested only in rapid obtaining of results, or wants to test the system before to get necessary tools for its installation, the single solution is to supply the Internet access to bergman.

We have then a problem of implementation Web interface to bergman taking into account comfortable usage and security, and the fact that the initial package was not developed with interface and security features.

2 Problem analysis

Transformation of the system into the Internet version implied series of modifications to assure the following features:

1. Comfortable mode to formulate tasks for calculations. In the original bergman, the information is entered in the format determined by Lisp and by Lisp functions call. It was necessary to study Lisp or User's Manual before to use the system. We can not suppose this for a visitor from Internet.
2. Demonstration of all base system functions including non-commutative ones that constitute a principal advantage of this system.
3. Reliability of the system. At errors in data bergman stops and shows error message of the Lisp interpreter. They use terms of this language and are not understandable for an arbitrary user. This happens, in particular, in the case of non-homogeneous polynomials. It is necessary to protect user of such situations.
4. Protection of computers. Bergman initiates Lisp interpreter that can execute any Lisp instruction including system ones. It is necessary to exclude the possibility to use intentionally or unintentionally the Web site as a tool destroying the computer system. It is necessary also to exclude pilling in the system big volumes of calculation results.

3 Particularities of working in Internet

From the beginning bergman was not projected to work in Internet. It produced series of problems to prepare the Internet version. One of principal problems is that during calculations big computational resources are used (processor time, operative and disk memory). It implies impossibility of placing the calculation program and the Web server on the same computer, because of reducing their productivity. We selected in our case a special computer for calculations only. Therefore the Web server contains an intermediate CGI program. This intermediate program take data from the net, transmit them to the calculation server, and returns in the net the obtained results. The general scheme of bergman Web interface is presented in Fig. 1.

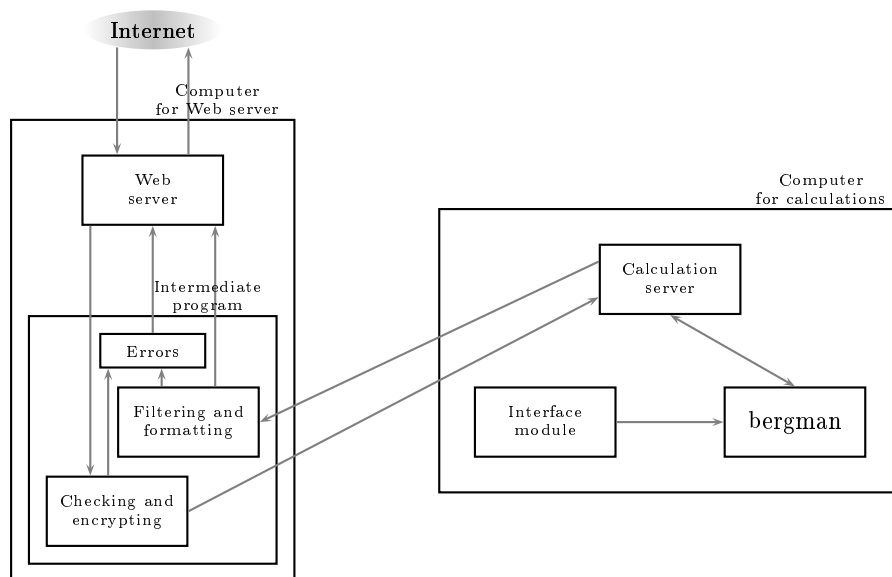


Figure 1. General scheme of bergman Web interface

Initial data for calculations are entered in a form on demo Web page. Then the data are checked for the most frequent errors by a

JavaScript program executed on client's (user's) side by user's browser. Examples of such errors are absence of parameters, impossible combinations of parameters (e.g., Hilbert series can be calculated only in non-commutative case), incorrect numerical values (e.g., field characteristic is to be a prime number). Then the data are transmitted to the Web server. On the server side, the intermediate CGI program is launched that takes these user's data. The intermediate program checks the input data very thoroughly, taking into account security aspects. E.g., only some prescribed characters should be used in relation lists, variable names, numeric fields, etc. Then the intermediate program encrypts the data and sends them to the calculation server. The encryption is executed in the manner not permitting inclusion in the transferred data any dangerous information (e.g., executable codes). The calculation server checks the data once more and generates a PSL program which is transmitted to the bergman system. The bergman system checks the data for algebraic errors (e.g., non-homogeneous polynomial relations) and executes the calculations. There is also an interface module that contains instructions (Lisp function definitions) that redefine bergman output commands to facilitate the posterior result formatting in HTML. This is made by introducing some special keywords. The results are returned, by the calculation server, back to intermediate CGI program. It filters this result flow, substitutes keywords by HTML commands and adds other formatting instructions. The formatted HTML result is returned to the user.

There is a possibility of background calculations. With this option, the calculation server compresses the results and puts the archive in the anonymous FTP server, and the user is informed by e-mail that his result is ready and can be taken from the specified address by FTP.

4 Aspects of security

This scheme has a vulnerable point from the security point of view. This is the connection between the intermediate program and the calculation server, because an attack is possible in this place. To solve this problem, we transmit encrypted data to the calculation server.

The mode of encryption does not permit invalid inclusions.

The repeated data checking by the intermediate program and by the calculation server also prevent system destabilization by incorrect data.

The following restrictions are imposed because of security considerations:

- variables can not contain arbitrary characters as in Lisp; only letters are permitted;
- number of users that can perform simultaneous calculations is limited;
- time of calculation and maximal power of resulting polynomials is also limited;
- a delay (approx. 1 second) is exposed between PSL program generation and its start;
- volume of results to be stored in the disk is limited.

5 Problem solution tools

To solve the described problem we used the following tools and languages:

- Perl version 5;
- Common Lisp;
- PSL version 3.4;
- JavaScript;
- HTML version 4;
- Bourne and C Shell;
- server WEB Apache 1.3;

- Sendmail version 8.9;
- standard UNIX utilities.

The intermediate program and the calculation server were implemented in Perl. Main consideration in language selection was the necessity to work with arbitrary character strings, because the volume of results is not known beforehand and may be very large. Perl is created on purpose to work with character strings of the arbitrary length and contains convenient tools to process calculation results. The language implementation uses garbage collection algorithms to free the unused memory in time.

Now we describe system components.

The intermediate program is a standard CGI script which takes user's data, encrypts them under the protocol that does not permit ambiguous data, and transfers them to server by TCP/IP. The calculation results returned to the server contains some keywords inserted by the interface module in the output data flow. The intermediate program filters the results and substitutes the keywords by corresponding HTML tags for presentation in the Web page.

The calculation server controls the resources and execution of bergman. It waits for the data from the net through the dedicated port. As the connection is established, the calculation server checks if the maximum number of connections is not exceeded closing the connection in the case. If the condition satisfies, the calculation server creates its copy using UNIX function fork, transfers to this copy all necessary data on the connection and waits for a next connection. The created copy checks the input data, converts them in Lisp statements and starts bergman. The output data are transferred to the intermediate program. The calculation server controls bergman execution and terminates it correctly if the fixed execution time is exceeded. During background calculations the calculation server packs output data from bergman in an archive file. Then the archive is moved to the anonymous FTP server, and an e-mail is send to the user with the address of the resulting archive.

The interface module is loaded inside the bergman. It redefines bergman output commands inserting in the output flow some keywords that are converted in HTML tags by the intermediate program. Except of this the interface module supplies some high level commands integrating several other commands of bergman and permitting, e.g., calculation of series with different parameters. This module is written in the base Lisp, the common subset of PSL and Common Lisp. Therefore this module can work in both bergman implementations.

6 Conclusions

The development of bergman is continuing. One of directions to develop bergman is to provide users with a reliable and easy-to-use interface. We described its implementation for the Web demo version of bergman. Of course the demo version restricts available features comparing to the full system with Lisp interface. A more complicated task is to provide local users with interface that accesses all bergman feature but do not supposes preliminary Lisp knowledge, permitting to pose tasks in algebraic terms.

Acknowledgements

The work was supported by the grant Nr. 1 of the Royal Swedish Academy of Sciences.

References

- [1] Bergman Web site: <http://www.matematik.su.se/bergman/>

S.Cojocaru, A.Colesnicov, L.Malahova,

Received May 15, 2002

Svetlana Cojocaru,
Institute of Mathematics and Computer Science,
Academy of Sciences of Moldova,
str. Academiei, 5, Chişinău, MD 2028, Moldova.
E-mail: *sveta@math.md* *kae@math.md* *mal@math.md*