

FGCS Project Evaluation Report

Sten-Ake Tärnlund

1 VISION AND PLANS

The research paradigm of the FGCS project is logic programming [2], a major breakthrough in computer science and artificial intelligence from the early seventies going back to some principal notions of resolution logic [3] in the sixties, see professor Robinson's historical account [4] in the FGCS-92 proceedings. Dr. Fuchi, early to envision the potential of logic programming, gave together with several colleagues a research vision [5] based on this paradigm that enthused an entire world. He also worked out a practical research plan [6] for the FGCS project itself focusing on parallel inference computing. I shall do my best to evaluate the FGCS project according to this plan and not the vision that I, however, expect will inspire researchers for many years to come, as the Hilbert program [7], from almost hundred years ago, still inspires mathematicians.

2 MAJOR ACHIEVEMENTS

My evaluation will follow the presentations of the FGCS project given at the conference and in the proceedings.

2.1 Parallel Inference Machine (PIM)

The various PIM machines, are the centerpiece of the FGCS project. They differ somewhat in architecture, which opens up possibilities for experiments. They are all MIMD machines. A PIM has clusters of processing elements connected in a network of clusters. It is scalable at the three levels. A prototype machine PIM-m is used for demonstrations. They are scalable farther than thousand processors. They run the PIMOS operating system and the KL1 language. The performances of a PIM is about 100 MLIPS, although these LIPS may not quite be the original LIPS, they show that the FGCS project has more than reached the original bold performance target. More to the point, we can expect performances beyond the 100 MLIPS up to a GLIPS and beyond. This performance and these machines are brilliant results and could not have been taken for granted when the project started. On the contrary, after this project started other machine ideas have been proposed e.g., the Connection machine [8] in the US. In contrast, it is a SIMD machine, its network is less scalable and more limited by communication speed, these factors make it a more special machine for particular problems. So, the project could have gone wrong, but it didn't and it produced a

unique general machine that cannot be seen anywhere else in the world. This is an outstanding result of its designers. In fact, the PIM itself justifies the focus of the FGCS project on parallel inference machines for logic programming.

One could, of course, argue whether or not the project itself should have used more hardware and software available on the market or develop its own as it mostly did. In general, I think both alternatives are justifiable. For Japan, being so strong on hardware, this project with logic programming as its key idea, is just an excellent place for various advanced hardware studies and experiments on machine prototypes. On the other hand, it would be possible to obtain several of the interesting results with more market hardware and software. This would show that these results are repeatable - a good scientific criterion. This line of research may lead to more procedural software, and that could impose a constraint on massive parallel processing, leading to more complicated programming and thus more expensive software development.

In general, machines of this structure [9], but with more market hardware and software, is likely to become general purpose parallel machines on the market in the near future. So as a bonus from this project, Japanese manufacturers could go from these PIM prototypes to competitive massive parallel computers with market software - a pleasant spin-off from the FGCS project.

2.2 Parallel Inference Software (PIMOS)

Pimos is a parallel operating system for a PIM and is written in the concurrent language KL-1 that includes control facilities e.g., for resource management. The concurrent languages GHC [10] and KL1 [11], are among the most prominent results of the FGCS project. GHC is an elegant successor of Parlog [12] and Concurrent Prolog [13]. It is famous in the world. KL1 has clearly been very useful for developing most of the software for the PIM:s. Nobody has done an operating systems project like this elsewhere, and it adds significantly to the success of the FGCS project. The operating systems principles of the PIMOS experience and the methodology of KL-1 comprehend new interesting concurrent programming ideas. We are keen to learn more about them and they deserve wider publicity.

At this point, a comment on the concurrent versus declarative logic programming paradigms [14] may be appropriate. The invention of concurrent languages by Clark & Gregory, Shapiro and Ueda & Chikayama has stimulated a great interest in solving concurrent problems that occur frequently in operating systems. Although, this approach departs from the declarative (semantic) idea of logic programming, these languages provide interesting techniques for concurrent problems e.g., processes, committed-choice, stream and-parallelism, and communications with terms containing variables and bounded-buffers. Taking up Quine's point [15] a solution in a concurrent language can, however, be regimented to a solution in logic. Furthermore, concurrent computa-

tions are not identical to parallel computations or always related to better performance of a parallel inference machine. In general, I view the ideas and the experiments on concurrency as brilliant and a very successful part of the FGCS project. The PIMOS system itself is unique!

2.3 Knowledge Base Management Software (KBMS) and Basic Artificial Intelligence

The methodology and general principles as well as the application experiments used in the FGCS project are very interesting. In particular, the application experiments have been useful in the field of artificial intelligence. Several special purpose languages have been developed for various applications e.g., CIL for natural language processing. There is also more basic AI research e.g., on theorem proving, hypothetical reasoning, analogy and non-monotonic reasoning. These are areas where results from logic programming have made several interesting contributions to the AI field recently. The results of the FGCS project make this progress even stronger.

2.4 Knowledge Programming Software and Programming Methodology

Beside the development of the concurrent languages there are also several other important results on programming methodology mainly building on results in the areas of: constraint programming [16], partial evaluation [17], meta programming, and program transformation. Several of these research results are superb, and have positioned Japanese research at the frontier. Some of these results could also play an important role for future software engineering. In fact, this methodology is particularly significant for the FGCS project since it could become a bridge between (declarative) logic programming and efficient parallel inference machines e.g., by automatically transforming logic specifications into parallel logic programs.

2.5 Experimental Parallel Inference Software

The project shows several fascinating choices of applications e.g., in legal reasoning that are impressive. The applications often demonstrate the parallel power of a PIM. Some of them have good potential of becoming blossom applications. This would be an interesting result in itself, but the methodology of developing such applications is also very interesting. With the PIMs and the craftsmanship of logic programming methodology at hand the researchers at ICOT are well placed to develop extraordinary applications. In fact, to take advantage of logic programming and its problem solving competence, and the parallel inference machines for efficient computations, may also be a good subject for a sequel project.

3 SUMMARY OF RESULTS

I will sum up my evaluation in six parts.

3.1 Research

Principal results:

- the PIM prototypes and their architectures;
- performance of PIM of 100 MLIPS and potential for more;
- concurrent languages, GHC and KLI, and their methodology;
- logic programming methodology on: concurrent software, constraint programming, partial evaluation, meta programming, and program transformation;
- several interesting applications e.g., legal reasoning.

A superb result, and an outstanding achievement. Consequently, I would rank the ICOT work on par with the best in the world.

The successful advance and experiments of the FGCS project validates some essential aspects of the logic programming paradigm - declarative logic programming for (economical) software development and parallel inference machines for efficient computing.

3.2 Research education and training

Several of the researchers work for Japanese companies. They have been visiting researchers at ICOT and will return to their companies. This leads to a large scale training of researchers at the Japanese companies that should yield significant spin-off effects in the future.

3.3 Project planning

ICOT has had a 10-year working period. This may impose an earlier commitment to a research idea than some researchers may want, on the other hand a commitment gives the opportunity to come up with concrete prototypes e.g., a PIM.

3.4 Scientific responses

We all know the impact of the FGCS project around the world. Japan has given a distinguished mark on computer science research in general and logic programming in particular. I shall only mention a few responses that I have seen: ESPRIT in Europe; MCC and the strategic computer project in the US; and in Sweden research programs

by STU and the creation of SICS. In addition, private companies have made significant efforts e.g., IBM.

3.5 Economics

As pointed out above, Japanese manufacturers have got the opportunity to go from the PIM prototypes to competitive massive parallel computers with more market hardware and software. Although such a move involves more than technical aspects, this economical opportunity is a pleasant spin-off from the FGCS project.

3.6 Public relations

We have seen and will see many press reports on the FGCS project. Some may be somewhat superficial as the article in the International Herald Tribune, the other day. Some will be more sophisticated as the two consecutive editorials in the New York Times, some time ago, interpreting the FGCS project not only as a success but recommending that the US should learn from this project and launch a new industrial policy for its computer industry. This recommendation is, of course, equally valid for Europe. In general, the results of the FGCS project are not yet fully understood and appreciated, and we would benefit from a better understanding of the achievements of the FGCS project.

4 A FEW RECOMMENDATIONS

Japan has set the stage for future research based on the success of the FGCS project. I shall only briefly mention some potentials for an international cooperation based on the Japanese results and the international results, in particular, from the many responses to the FGCS project around the world.

There are several interesting subject matters e.g., from parallel inference machines to advanced AI via programming methodology. Logic programming would be helpful to strengthen the coherence of such an effort. This research paradigm was considered the most powerful for the FGCD project. I do not think it has been replaced, new projects would energize the original vision and be a challenge for the new millenium.

Some principal areas would be:

- parallel inference machines;
- software methodology e.g., specifications, transformations and verifications;
- software development methodology;
- artificial intelligence methodology;
- advanced applications.

5 ACKNOWLEDGMENTS

I would like to thank the leadership of the FGCS project and all its members I have had the pleasure to meet and work with, for example, in several enjoyable workshops. In particular, I admire the works and monumental effort by Dr. Fuchi, Dr. Furukawa and Dr. Uchida to make this project such an outstanding success.

NOTES:

- [1] Address: Computing Science Department, Uppsala University,
P.O. Box 520, S-751 20 Uppsala Sweden.
Phone: + 46 010 222 4002.
Fax: +46 18 521270.
Email: stenake@csd.uu.se.
- [2] R. Kowalski, Predicate Logic as Programming Language, Proc. IFIP-74,
Stockholm, North-Holland. Colmerauer, A., Kanoui, H., Pasero, R., &
Roussel. P., 1972, Un Systeme de Communication Homme-Machine en
Francais, Univ. d'Aix-Marseille, Luminy, Marseille.
- [3] J., A., Robinson, 1965, A Machine-oriented Logic Based on the
Resolution Principle, J. ACM, New York.
- [4] J., A., Robinson, The Role of Logic in Computer Science and
Artificial Intelligence, Proc., FGCS-92, Tokyo.
- [5] T., Moto-oka, (ed.), Fifth Generation Computer Systems, JIPDEC:
North-Holland, 1982.
- [6] K., Fuchi., in T., Moto-oka, (ed.), Fifth Generation Computer Systems, JIPDEC:
North-Holland, 1982.
- [7] Hilbert, D., Mathematical problems. Lecture delivered before the International
Congress of Mathematicians at Paris in 1900. Bull. Amer. Math. Soc., vol 8,
1901-2.
- [8] Hillis, W.,D., The Connection Machine, MIT Press, Cambridge, 1985.
- [9] CM-5, iPSC/Delta, nCUBE etc.,
- [10] Ueda, K., Guarded Horn Clauses, in Logic Programming 85, E., Weda (ed.),
Lecture Notes in Computer Science 221, Springer-Verlag, 1986.
- [11] Ueda, K., and Chikayama, T., Design of the Kernel Language for the Parallel
Inference Machine, The Computer Journal, Vol 33, No 6, 1990.

- [12] Clark, K., and Gregory, S., PARLOG: A Parallel Logic Programming Language. Research Report DOC 83/5, Department of Computing, Imperial College of Science and Technology, 1984.
- [13] Shapiro, Y., A Subset of Concurrent Prolog and its Interpreter. Tech. Report TR-003, ICOT, 1983.
- [14] The concurrent versus the logic programming issue is an instance of an issue of philosophy of logics i.e., whether or not a problem e.g., a concurrent, modal or temporal problem should have its own logic or should be solved in standard predicate logic as argued by Quine see for example Word and Object, Wiley, 1960.
- [15] See the preceding footnote.
- [16] Colmerauer, A., Opening the Prolog III Universe: A new generation of Prolog promises some powerful capabilities, BYTE, August 1987. Jaffar, J., and Lassez J-L, Constraint Logic Programming, In 4th IEEE Symposium on Logic Programming, 1987.
- [17] Futamura, Y., Partial Evaluation of Computation Process: An Approach to a Compiler-Compiler. Systems, Computers, Controls 2,1971.

Sten-Åke Tärnlund
Uppsala Univ.

Fifth Generation Computer Systems (FGCS)

Logic programming

A major breakthrough in computer science!

Parallel inference

A very important computation method
for speed.

Project program

Vision - plans

cf The Hilbert program for mathematics.

Major Achievements

1. Parallel Inference Machine (PIM)

Adapted cluster HMM machine. Unique!
100 MIPS

cf Connection machine

Comments: General vs special hardware.

2. Parallel Inference Software (PIMOS)

Nobody has done this elsewhere

3. Knowledge Base Management Software (KBMS)

See my report

4. Knowledge Programming Software

5HC K2.1 concurrent versus declarative L2
experiment in CLP.

5. Experimental Software

Many interesting applications e.g. logical reasoning

Suggestions & Conclusions

Research results.

I would rank the best of the work
on par with the best in the world e.g.

PIM, GHC...

* performance 100 MIPS

* implicit methodology for software development.

* logic programming paradigms are successful

Scientific politics

Successful

Planning of the project good for concrete results.

Economical

Probably good prospects.

Politics

Herald Tribune may view the project as a
failure but New York Times viewed it
successful!

Finally, the FACS results should be continued!