

# Evaluating the *state of the art* of parallel programming systems

Technical Report KIS No. 1 / 2005

Michael Süß, Claudia Leopold  
University of Kassel  
Research Group Programming Languages / Methodologies  
Wilhelmshöher Allee 73, D-34121 Kassel, Germany  
{msuess, leopold}@uni-kassel.de

February 28, 2005

## Abstract

This paper describes our plans to evaluate the present state of affairs concerning parallel programming and its systems. Three subprojects are proposed: a survey among programmers and scientists, a comparison of parallel programming systems using a standard set of test programs, and a wiki resource for the parallel programming community - the *Parawiki*. We would like to invite you to participate and turn these subprojects into true community efforts.

## 1 Introduction

The world of parallel programming today is diverse and complex. A large number of new systems have been developed during the last years, yet few are in practical use. Many of the new systems show great promise, but do not scale well to real world applications. Others never leave the testing state. The few systems that seem to be used for real applications are mostly more than 5 years old (e.g. *MPI*, *OpenMP*), and are usually based on languages that are even older (*C*, *C++*, *Fortran*).

The main goal of our project is to generate an up-to-date overview of the field, thus to find reasons for the described state, by answering questions such as:

- What parallel programming systems are used in real world applications?
- Which parallel programming systems have been developed in the past, and which are being developed at the moment?
- What are the strengths of these various systems, and what are their problems?
- How do the systems compare to each other?
- Which application areas and architectures are the systems useful for?

To get answers, three subprojects are being pursued. They are described in detail in the following sections. The first subproject, considered in section 2, is a survey that we want to take in order to find out how popular the various systems are at the moment. The next subproject in section 3 is a comparison of parallel programming systems based on a standardized set of applications. As third subproject, section 4 proposes a community effort based on wiki technology to generate a lasting resource for all questions related to parallel programming. We close this report with a short outlook in section 5.

## **2 A survey on current usage of parallel programming systems**

To gather more information about the present usage of parallel programming systems in *real* projects, we have set up a short online-poll. We are hoping to gain several insights through this survey, such as:

- Which programming languages are used for parallel programming ?
- Which extensions / libraries for these languages are used ?
- Which parallel programming systems are known in the community ?
- Which hardware and operating system platforms are in use ?
- What shortcomings do the current parallel programming systems have ?

The poll consists of 9 questions. They are mostly multiple choice, with an option of adding more answers when the given ones are not sufficient. To get an impression, here is question 1 as an example (the layout is different in the poll):

1. How often have you used the following programming languages as a basis for your parallel applications during the last 3 years?

(N/K=unknown language – 1=never – 2=sometimes – 3=about half of the time – 4=often – 5=for every parallel application)

- C
- C++
- Fortran
- Java
- functional language(s) (if yes, which?)
- logical language(s) (if yes, which?)
- other programming languages(s) (if yes, which?)

With this poll, we are hoping to reach people from different backgrounds: scientists, application programmers, and anybody else with some experience in the field of parallel programming. If you consider yourself a member of these groups, please take the time to fill out our survey at [Süß05]. The results will be made available to the public on the project website [Eva].

### 3 An evaluation of selected parallel programming systems

Nothing beats real programming (or *getting ones hands dirty*) when it comes to evaluating software. Taking this to our hearts, we have set up a project, where a group of students are programming a standard set of parallel applications with different parallel programming systems. This set has first been proposed by Wilson in [WI95] and is known as the *Cowichan problems*.

It consists of 14 short applications (where short means that a sequential version can be written in an afternoon) that are defined by their inputs, outputs, and a

general mission statement. No concrete algorithms are enforced, to account for the different capabilities, strengths and weaknesses of the parallel programming systems to be tested. Note that Wilson published two versions of the Cowichan problems. The first version turned out to be too big, as the problems presented there took as long as a term for a student to implement. We refer to the second version, for which also a `Pthreads` implementation has been provided in [Wil].

## 4 The Parawiki - a resource for parallel programmers

How does one choose one of the available parallel programming systems for a particular project? Where can one go hunt for information about features, reliability and *real* user experiences (not supplied by the programmers of the system in question) ?

Of course, the most valuable resources in a field like this are personal experiences over several projects and years. But many people do not have these, and still need to take educated decisions about the tools for their next project. These projects could be student term assignments or multi-million dollar projects in companies.

Of course, one can go ask on one of the available mailing lists or forums (should one be able to find one that fits), or try his luck at one of the available link collections (like [dmo]). Search engines might be of help, too, but they require much time to hunt through a lot of hits. Books such as [Leo01] may be a nice starting point, but are unfortunately never really up to date, especially when it comes to research projects. For these, one can look into conference proceedings or at the home page of the project, but the information supplied there is rarely unbiased.

All of these methods have one thing in common: They are very time consuming and not likely to lead to reliable and objective information. Believe us, the authors have been through this more than once.

The solution we are proposing is not new, but has, to our knowledge, not yet been attempted for this particular field. We are proposing a web-portal called the *Parawiki* [Theb], where a vast amount of information will be collected about parallel programming systems. Topics include the history of a system, whether or not the system is still actively maintained, features and caveats, as well as links to resources on the web and maybe books describing the system. General information about parallel programming and parallel architectures, for instance definitions of basic terms used in the field are included as well.

So far this sounds much like one of the existing portals about programming lan-

languages such as the *Language Guide* [Thea] or one of the available online dictionaries [Fre, Com]. This is true to a certain extent; in fact these websites even inspired the authors to try something similar for the topic of parallel programming, for which there seems to be no such site.

A crucial difference between our resource and these sites is that in the Parawiki, the entries will not be written and kept up to date entirely by our students or us, but by volunteers from the parallel processing community (might that be experienced students, application programmers or scientists). Of course, we are encouraging the authors of parallel programming systems to put up an initial review of their systems, but we are as well hoping to attract comments from the actual users of these systems.

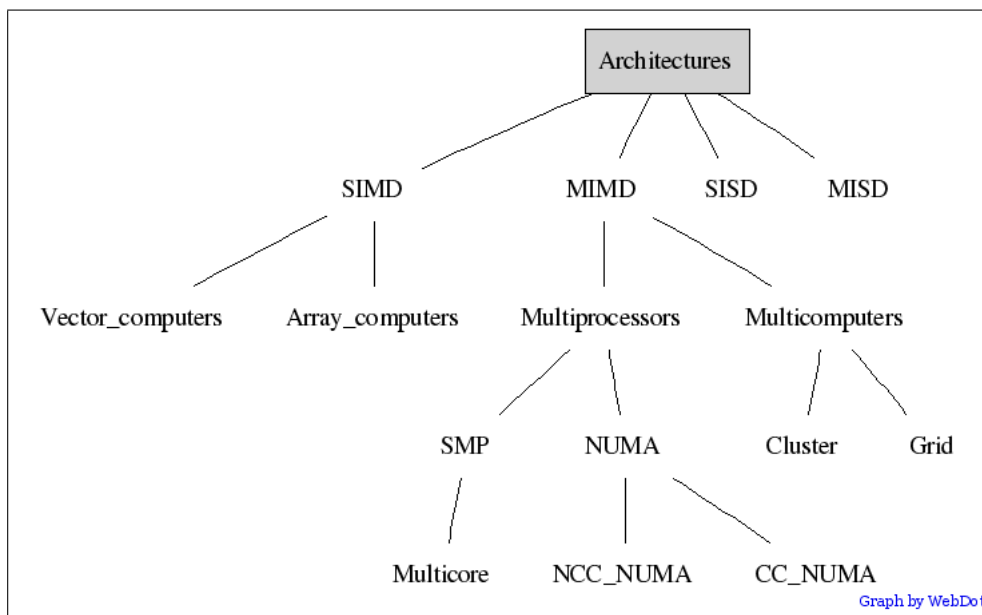
The flexibility of this approach is best served by a Wiki, which will be the central part of our portal. All readers who are not convinced that a community is willing and able to build and support an undertaking of this size, we can only point to the Wikipedia [Wik] and several other large and successful wikis and their participants. Several hundreds of thousands of pages are maintained in these communities with great success.

Over time, we are hoping to merge this effort with the one presented in section 3. The Cowichan Problems can be used as a starting point for comparing programming languages to each other. We are interested in programs and experiences from both capable programmers with deep insight into a particular system, as from newcomers who are playing around with different systems. Contributions may refer to the Cowichan problems or other applications.

Furthermore, a voting system can be implemented into or on top of the portal, so that the data gathered through the survey described in section 2 can be continuously gathered, processed and presented inside the Parawiki.

In addition to the functionalities that wiki systems generally support, the Parawiki incorporates classifications. Usually, wiki systems have a flat structure, that is, all pages are located at the same level, and connections are provided through hyperlinks. On the other hand, it is often observed that classifications help to organize a complex field. With classifications, it is easier to grasp the overall picture and, especially for a new and diverse field such as parallel programming, classifications can point to opportunities in the design space that have not been explored so far.

The Parawiki invites the community to invent and improve classification schemes. We have defined a syntax to input tree-like categorizations that may provide a quick overview over certain subtopics and additionally serve as tables of contents. New systems can be inserted into existing trees, and these can be reorganized according to changing times and tastes by any user. Many categorizations can



**Figure 1:** A tree visualizing parallel architectures

live side by side, as of course ordering by different properties is possible and encouraged. See figure 1 for an example.

We would like to close this section by encouraging you to take part in this undertaking and support the creation of a place where the parallel programming community can learn from, discuss and work with each other - please support the Parawiki [Theb]!

## 5 Concluding remarks and perspectives

A difficult question has not been answered by us yet: How does one actually compare parallel programming systems? Phrases like *applicability*, *usability* and *performance* come to our minds, yet to clearly define these terms and put them into key figures remains complicated. We are planning on writing a short catalog of characteristics that can be applied to the parallel programming systems and the programs written with them. This catalog will be based on the measures and terms coined in the world of sequential computing (where there is a lot more research going on concerning these terms).

This work can only succeed if the proposed subprojects (and especially the Parawiki) turn into true community efforts. We have provided the infrastructure for the Parawiki and some initial content, and now are hoping to raise enough interest in

the parallel processing community to take an active role in its future development. Let us therefore repeat ourselves once again: If you consider yourself a member of the parallel programming community, please take the time to fill out our survey [Süß05] and join the Parawiki [Theb] !

## Acknowledgments

We are grateful to Beliz Senyüz, who contributed most of the initial content to the Parawiki as a part of her master thesis. Furthermore, thanks to Holger Bischof for his thorough review of the parallel programming survey.

## References

- [Com] Computer dictionary online. <http://www.computer-dictionary-online.org/>.
- [dmo] dmoz - open directory project. <http://dmoz.org/>.
- [Eva] Evaluation of parallel programming systems. [http://www.plm.eecs.uni-kassel.de/plm/?eval\\_pps](http://www.plm.eecs.uni-kassel.de/plm/?eval_pps).
- [Fre] Free on-line dictionary of computing. [www.foldoc.org](http://www.foldoc.org).
- [Leo01] Claudia Leopold. *Parallel And Distributed Computing*. John Wiley & Sons, 2001.
- [Süß05] Michael Süß. Parallel programming survey. <http://www.plm.eecs.uni-kassel.de/parasurvey/>, February 2005.
- [Thea] The language guide. <http://www.engin.umd.umich.edu/CIS/course.des/cis400/>.
- [Theb] The Parawiki. <http://parawiki.plm.eecs.uni-kassel.de>.
- [WI95] G. Wilson and R. Irvin. Assessing and comparing the usability of parallel programming systems, 1995.
- [Wik] Wikipedia the free encyclopedia. <http://www.wikipedia.org/>.
- [Wil] G. Wilson. The ANSI C implementation of the cowichan problems.