A Survey of Parallel Programming Tools

Doreen Y. Cheng

Report RND-91-005, May 1991

RND Branch
NAS Systems Division
NASA Ames Research Center
Mail Stop 258-6
Moffett Field, CA 94035-1000

1 Computer Sciences Corporation, NASA Contract NAS 2-12961, Moffett Field, CA 94035
Introduction

This survey examines thirty-nine parallel programming tools. Focus is placed on those tool capabilities needed for parallel scientific programming rather than for general computer science. The tools are classified with current and future needs of NAS in mind: in particular, existing and anticipated NAS supercomputers and workstations, operating systems, programming languages and applications. They are divided into four categories: suggested acquisitions, tools already brought in, tools worth tracking, and tools eliminated from further consideration at this time.

Section 1 lists the tools that are suggested acquisitions. They have been ranked according to the functions provided, maturity at the time of this survey, and suitability of building new functions on top of them. Some of them have overlapping functions. They have been listed in order, most preferred first; price was not considered in this ranking.

No ordering has been placed on the listing for the remaining sections. Section 2 presents the tools that have already been brought to NAS, but which are not yet available to the users. They were obtained either because they were in the public domain, or some of their functionality was desired by NAS parallel tool developers. Of these tools, only Schedule, Force, ParaScope, Axe/Aims, Parti, Hypertask, and CRAY/fpp are of current interest to NAS.

Section 3 lists the tools that are of interest to NAS, but not suggested acquisitions at this time. Some of them are in an early, but active, research stage. Others have been discarded after being evaluated at NAS; they may become very attractive if new development directions are taken. It would be prudent to keep track of their development.

The tools in Section 4 are eliminated from further consideration at the present time. Some of these tools are eliminated because they are not suitable for NAS applications or NAS platforms. Others are eliminated because it is difficult to contact and collaborate with their developers.

The functions and maturity of each tool are evolving as well as the interests of NAS users and parallel tool developers. Except for Axe/Aims, the information collected in this survey reflects the status up until February, 1991. Changes in Axe/Aims between February, 1991 and May, 1991 have been included.

To make it easier for readers to find a specific tool in the report, an alphabetized look-up table is included at the beginning. The table indicates the usefulness of a tool to NAS as well as the page and section where the tool is described.

Four tables are appended at the end of the report. These tables compare the features and status of the tools that are of immediate interest to NAS (i.e. those in Sections 1 and 2 and some in Section 3). The first table lists the functions of the tools that transform a sequential program into a parallel one. The second table covers the functions that help users write parallel programs. The third table lists the functions that assist in parallel debugging and performance optimization. The last table lists the availability and system requirements of the tools.

In this report, OS means operating system and GUI represents graphic user interface.
<table>
<thead>
<tr>
<th>Name</th>
<th>Useful to NAS</th>
<th>Page Number</th>
<th>Section Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axe/Aims</td>
<td>Y</td>
<td>20</td>
<td>2.5</td>
</tr>
<tr>
<td>C-Linda</td>
<td>Y</td>
<td>13</td>
<td>1.9</td>
</tr>
<tr>
<td>Caper</td>
<td>N</td>
<td>30</td>
<td>3.5</td>
</tr>
<tr>
<td>Code/Rope</td>
<td>N</td>
<td>35</td>
<td>3.10</td>
</tr>
<tr>
<td>CRAY/fdp</td>
<td>Y</td>
<td>25</td>
<td>2.10</td>
</tr>
<tr>
<td>Dino</td>
<td>Y</td>
<td>31</td>
<td>3.6</td>
</tr>
<tr>
<td>E/SP</td>
<td>Y</td>
<td>7</td>
<td>1.3</td>
</tr>
<tr>
<td>Express</td>
<td>Y</td>
<td>6</td>
<td>1.2</td>
</tr>
<tr>
<td>FATCAT</td>
<td>Y</td>
<td>9</td>
<td>1.5</td>
</tr>
<tr>
<td>Faust</td>
<td>Y</td>
<td>10</td>
<td>1.6</td>
</tr>
<tr>
<td>FLO</td>
<td>N</td>
<td>33</td>
<td>3.3</td>
</tr>
<tr>
<td>Force</td>
<td>Y</td>
<td>17</td>
<td>2.2</td>
</tr>
<tr>
<td>Forge/MIMDizer</td>
<td>Y</td>
<td>3</td>
<td>1.1</td>
</tr>
<tr>
<td>Fortran-Linda</td>
<td>Y</td>
<td>27</td>
<td>3.2</td>
</tr>
<tr>
<td>Hypertask</td>
<td>Y</td>
<td>24</td>
<td>2.9</td>
</tr>
<tr>
<td>Hypertool</td>
<td>Y</td>
<td>15</td>
<td>1.11</td>
</tr>
<tr>
<td>IC*</td>
<td>N</td>
<td>41</td>
<td>4.5</td>
</tr>
<tr>
<td>IPS-2</td>
<td>Y</td>
<td>11</td>
<td>1.7</td>
</tr>
<tr>
<td>Kali-Fortran</td>
<td>Y</td>
<td>32</td>
<td>3.7</td>
</tr>
<tr>
<td>KAP/CRAY</td>
<td>N</td>
<td>36</td>
<td>3.11</td>
</tr>
<tr>
<td>Monmacs</td>
<td>Y</td>
<td>14</td>
<td>1.10</td>
</tr>
<tr>
<td>OACIS Tools</td>
<td>N</td>
<td>34</td>
<td>3.9</td>
</tr>
<tr>
<td>Olympus</td>
<td>N</td>
<td>38</td>
<td>4.2</td>
</tr>
<tr>
<td>Paragraph/PICL</td>
<td>Y</td>
<td>23</td>
<td>2.8</td>
</tr>
<tr>
<td>ParaScope</td>
<td>Y</td>
<td>18</td>
<td>2.3</td>
</tr>
<tr>
<td>ParaScope/Debugger</td>
<td>Y</td>
<td>28</td>
<td>3.3</td>
</tr>
<tr>
<td>Parti</td>
<td>Y</td>
<td>22</td>
<td>2.7</td>
</tr>
<tr>
<td>Pat</td>
<td>N</td>
<td>19</td>
<td>2.4</td>
</tr>
<tr>
<td>Pie</td>
<td>N</td>
<td>21</td>
<td>2.6</td>
</tr>
<tr>
<td>PISCES 2</td>
<td>N</td>
<td>37</td>
<td>4.1</td>
</tr>
<tr>
<td>Poker</td>
<td>N</td>
<td>29</td>
<td>3.4</td>
</tr>
<tr>
<td>PPD</td>
<td>Y</td>
<td>26</td>
<td>3.1</td>
</tr>
<tr>
<td>Schedule</td>
<td>Y</td>
<td>16</td>
<td>2.1</td>
</tr>
<tr>
<td>Simple/Care</td>
<td>N</td>
<td>43</td>
<td>4.7</td>
</tr>
<tr>
<td>Strand88</td>
<td>X</td>
<td>12</td>
<td>1.8</td>
</tr>
<tr>
<td>Topsys</td>
<td>N</td>
<td>40</td>
<td>4.4</td>
</tr>
<tr>
<td>Triplex</td>
<td>N</td>
<td>39</td>
<td>4.3</td>
</tr>
<tr>
<td>VecPar 77</td>
<td>Y</td>
<td>8</td>
<td>1.4</td>
</tr>
<tr>
<td>VMMP</td>
<td>N</td>
<td>42</td>
<td>4.6</td>
</tr>
</tbody>
</table>
1. Suggested Acquisitions

1.1 Forge/MIMDizer

Function of Forge:

- Program instrumentation
- Dependency analysis
- DO-loops parallelization guided by run time profiling
- Parallel/vector directives insertion
- Queries about the program (use-def, call-tree, common use, routine interface consistency, timing variable trace, etc)
- Code restructuring (invert loop, split loop, merge loop, collapse nested loops, unroll loop, inline routine, remove secondary induction variables, etc)
- A database on which new tools can be developed

Function of MIMDizer:

- All functions of Forge available
- Language extension to Fortran for programming message-passing machines
- Code frames generated for specification of communication and synchronization between code blocks written in Fortran (specification automatically transformed into message passing)
- Consistency checking for message-passing, variable definition, and caller-callee argument passing
- Automatic decomposition of arrays and static distribution of loops onto processors
- Automatic and dynamic repartitioning of arrays when needed
- Monitoring facilities for performance tuning and debugging
Evaluation of Forge by Doreen Cheng through use:

- The interactive nature provides convenient access to the tools and to information about the program.
- Preliminary evaluation showed the user response time is faster than that of other tools with similar functionality.
- The database is designed for program analysis. The size of the Forge database is only 3 to 4 times the size of the source code. Its speed is noticeably faster.
- Many bugs were encountered but PSR fixed them quickly.
- Forge flags a converted loop as parallel by simply using a DO ALL compiler directive. This has resulted in significant performance degradation which cancels the potential speedup by parallelization.
- Forge lacks the capabilities to guide dependence elimination, code transformation and parallelization.
- Vectorization and parallelization functions do not use a common user interface and do not allow mixed usage.

Evaluation of NfIMDizer by Doreen Cheng through literature search and phone contacts with the vendor:

- Extensive set of tools for programming message-passing machines.
- Alpha release.
- Lacks automatic load balancing.

Platform for Forge: Y-MP, X-MP, CRAY2, Sun, IRIS, NEC (negotiating)
Platform for MIMDizer: iPSC, Sun, IRIS
OS: UNIX
Language: Fortran
GUI: X-Windows, Sunview
Cost*: $28,050/workstation; $4,500/yr maintenance
Facility license**: $93,500
Unlimited license***: $158,950, $18,000/yr maintenance
Supplier: Pacific Sierra Research
Contact: John Levesque, (916) 621-1600

* Numbers quoted are for NASA Ames Research Center.
** One of two options available:

(1) Software available for running on up to 10 workstations.
(2) X-Window and/or terminal version for running on a single-host communicating with as many workstations (running X-Windows) or terminals emulating VT100.

*** Unlimited workstations and unlimited hosts running X-Windows and/or terminal version.

See reference 1 2 3
1.2 Express

Function:

- Library calls provided for parallelization
- Program instrumentation
- Hardware configuration management
- Automatic loop parallelization
- Automatic data distribution and domain decomposition
- Run time profile used for guidance
- Dynamic load balancing
- Parallel I/O
- Interactive memory access visualization
- Post-mortem communication and event analysis
- Communication and event monitoring
- Parallel algorithm monitoring
- Source level debugger
- Deadlock detection
- Converts Fortran 90 source code to Fortran 77
- A database used for run time information

Evaluation by Doreen Cheng through literature search and phone contacts with the vendor:

- Provides extensive set of tools for message-passing machines (debugging, performance monitoring, load balancing and parallelization).
- Covers a broad range of hardware, operating system, and languages.
- Debugger, profiler, parallel I/O, graphics are not available on Y-MP, and may not be in the future.
- Lacks support for interactive dependency analysis.

Platform: iPSC, Y-MP (beta), NCUBE, Sun, PC, Macintosh
OS: UNIX, DOS, Macintosh
Language: Fortran 77, Fortran 90, C, C++
GUI: X-Windows, Sunview, Postscript
Cost: $3,000 per Intel iPSC/860
       $15,000 per Y-MP
       $1,500 for network of Suns
       20% maintenance fee per year
Supplier: ParaSoft
Contact: Adam Kolawa, (818) 792-9941

See reference 4
1.3 E/SP:

Function:

- Program instrumentation
- Dependency analysis
- DO-loop parallelization
- Rough estimate of performance
- Parallel directive insertion
- Graphic editor for constructing new parallel programs
- A database for developing new tools

Evaluation by Doreen Cheng through literature search, observing demo, and phone contacts with the vendor

- The interactive nature provides a convenient access to tools and information about the program.
- For 1000 lines of code, 1 Mbytes is required for the database generated.
- It lacks support for vectorization.
- It lacks the capability to guide dependence elimination, code transformation and parallelization.

Platform:

- Sun
- In the process of negotiating with Kuck Associates for CRAY code generation
- Port to IRIS possible if desired

Language: Fortran

OS: UNIX

GUI: Sunview, X-Windows

Cost: TBD

Supplier: Scientific and Engineering Software Inc.

Contact: James C. Browne, (512) 474-4526

See reference 5
1.4 VecPar_77

Function:

- Dependency analysis
- Code transformation for vectorization (merge loops and if blocks, eliminate dependence, unroll loop, permute loops, etc)

Evaluation by Doreen Cheng through literature search and phone contacts with the vendor and users:

- Support for Fortran 90 is being discussed.
- Has a command-line oriented textual user interface.
- It lacks the capability to guide dependence elimination, code transformation and parallelization.
- Vectorization and parallelization tools are two separate tools.
- For large subroutines, large amounts of memory are required.

Platform: Sun, Y-_MP

OS: UNIX

Language: Fortran

GUI: None

Cost: $2,500/workstation; $19,500/Y-MP
      Unlimited license (use and machine) $33,510/yr
      Perpetual: $78,250

Supplier: Numerical Algorithms Group Inc.

Contact: Sheila Caswell, (708) 971-2337

See reference 6
1.5 FATCAT

Function:

- Dependency analysis

Evaluation by Doreen Cheng through literature search and phone contacts with the vendor and users:

- Able to analyze dependencies in recursive calls.
- Ported onto Y-MP in much shorter time than university products.
- Difficult to interpret the output (too much data, little guidance).
- The company is looking for funding for further development.
- For large subroutines, large amounts of memory are required.

Platform: Sun, CRAY2, X-MP, Y-MP

OS: UNIX

Language: Fortran

GUI: None

Cost: $50,000 for source license
      Executable not for sale

Supplier: New Jersey Advanced Technology Inc.

Contact: David Klappholz, (201) 420-5509

See reference 7
1.6 Faust:

Function:

- Dependency analysis
- Interactive variable tracing
- Simulator to discover the parallelism in a program
- Performance monitoring
- Interactive graphic user interface

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Tools could be integrated into NAS environment.

Platform: CRAY
OS: UNIX
Language: Fortran
GUI: X-Windows
Cost: $100 for source
Supplier: CSRD of Univ. of Illinois
Contact: David Hammerslag, (217) 244-0277

See reference 8
1.7 IPS-2

Function:

- Performance monitoring, analysis, and visualization
- Critical path analysis and visualization

Evaluation by Doreen Cheng through literature search and e-mail with the supplier:

- Second implementation reduces program intrusiveness and storage requirements, and adds a graphic user interface.
- Tools could be integrated into NAS environment.

Platform: Y-MP (will be ready by summer 1991), Sequent Symmetry

OS: UNIX

Language: Fortran, C

GUI: X-Windows (X11)

Cost: $300 for source

Supplier: Univ. of Wisconsin

Contact: Barton P. Miller, (608) 263-3378

See reference 9
1.8 Strand88

Function:

- Parallel language that calls code blocks written in Fortran or C
- Tools that monitor processor and communication load, and visualize the data

Evaluation by Doreen Cheng through literature search and attending seminar:

- The prolog-like language may be difficult for scientists to accept.

Platform: iPSC, Sun, Y-MP (available in spring 1991), Sequent, Encore

OS: UNIX

Language: C, Fortran

GUI: X-Windows

Cost: $22,000 for iPSC/860; maintenance $3,375/yr
$3,000/Sun; maintenance $900/yr
Site license range: $30,000 to $40,000
NAS can get 30% government discount

Supplier: Strand Software Technology Inc.

Contact: Timothy G. Mattson, (503) 690-9830

See reference 10 11
1.9 C-Linda

Function:

- Language extensions for parallel programming
- Source level debugger
- Program execution monitoring and visualization
- Consistency checking for tuple space usage
- Monitors message traffic and moves Linda run time library to reduce the traffic.
- Tuple space usage visualization

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Supported on a broad range of hardware platforms.
- Fortran routines can be called.

Platform: iPSC/860 (will be available shortly), Y-MP (not fully debugged), Sun, IRIS, IBM RS6000 Apollo, Encore, Sequent

OS: UNIX

Language: C, Fortran (Fortran is not directly supported, can be called from C.)

GUI: X-Windows (for debugger)

Cost:

$7,000/10 workstation
$20,000 for iPSC/2
Site license for ARC: $90,000

Supplier: Scientific Computing Associates Inc.

Contact: Ellen Smith, (203) 777-7442

See reference 12 13
1.10 MONMACS

Function:

- Language extensions for parallel programming
- Post-mortem performance analysis

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Has non-standard extensions.

Platform: iPSC/860

OS: UNIX

Language: C, Fortran (available in March, 91)

GUI: X-11

Cost: Public domain

Supplier: Argonne National Lab

Contact: Ewing Lusk (708) 972-7852

See reference 14
1.11 Hypertool

Function:

• Automatic partitioning of a program for message-passing machines
• Inserts communication and synchronization primitives
• Automatic scheduling and load balancing

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

• Requires the user to call all routines that will be executed concurrently from the main routine.
• Retargetable compilers are planned.
• Is a research product, still buggy.

Platform: iPSC/2
OS: UNIX
Language: C
Cost: University distribution
Supplier: U.C. Irvine
Contact: Daniel D. Gajski (714) 856-4155

See reference 15
2. Tools Already in Ames Research Center:

2.1 Schedule

Function:

- Language extension to express dependencies between code blocks written in Fortran
- Performance monitoring
- Memory access visualization
- Program execution visualization
- Critical path determination
- Program profiling
- Task scheduling
- Dynamic load balancing

Evaluation by Doreen Cheng through literature search and observing demo:

- Helps in writing new parallel programs.
- For functional parallelization.
- Loops need to be transformed to subroutine calls for parallel execution.
- No help in dependency analysis.
- Static specification for parallelism only.
- Versions work for message-passing machines will be available at the end of summer, 1991.

Platform: CRAY2
OS: UNIX
Language: Fortran
GUI: X-Windows
Cost: Public domain
Supplier: Univ. of Tennessee
Contact: Jack Dongarra, (615) 974-8295
Local contact: Doreen Cheng, (415) 604-4361

See reference 16 17
2.2 Force

Function:

- Fortran extension for parallel programming

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Good performance has been reported for structure analysis problems.
- Uses static partitioning, one level fork-join parallelism only.

Platform: Y-MP, CRAY2, Encore, Sequent, Convex, Alliant

OS: UNIX

Language: Fortran

GUI: None

Cost: Public domain

Supplier: Univ. of Colorado

Contact: Harry Jordan, (303) 492-1411

Local access: Doreen Y. Cheng (415) 604-4361

See reference 18 19 20
2.3 ParaScope

Function:

- Dependency analysis
- Code transformation
- DO-loop parallelization

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Supplier plans to integrate performance visualization tools of SCHEDULE into ParaScope.
- Debugging facilities are in development. These tools may require the program to be written in PCF Fortran.
- Supplier plans to develop tools to allow a user to annotate a program written for shared-memory machines and automatically translate it to message-passing.

Platform: SUN
OS: UNIX
Language: Fortran
GUI: X-Windows
Supplier: Rice Univ.
Contact: Ken Kennedy, (713) 285-5186
Local contact: Doreen Cheng, (415) 604-4361

See reference 21 22 23
2.4 PAT

Function:

- Program instrumentation
- Performance analysis
- Parallel debugger
- Static code analysis
- Interactive parallelization

Evaluation by Kathi Flecher and Doug Pase through use:

- Has serious bugs.
- Lacks many claimed functions.

Platform: CRAY

OS: UNIX

Language: Fortran

GUI: X-Windows

Supplier: Georgia Institute of Technology

Contact: Kevin Smith, smith@boa.gatech.edu

Local contact: Doreen Cheng, (415) 604-4361

See reference 24
2.5 Axe/Aims

Function:

- Axe provides a *Parallel Program Behavior Description Language* for describing interactions between processes.
- Axe predicts the performance of the program.
- Aims instruments a Fortran program and collects the run time information.
- A visualization tool allows viewing the behavior and the run time information.

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- About to be released for use.

Platform: iPSC/860

OS: UNIX

Language: Fortran

GUI: X-Windows

Supplier: NASA Ames Research Center

Contact: Jerry Yan, (415) 604-4381

See reference 25
2.6 PIE

Function:

- Language extensions for parallel programming
- Performance predictor
- Performance trace and visualization

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Non-standard language extensions
- Has potential to be more than a performance tuning tool.
- Difficult to port to UNIX

Platform: Sun, VAX, Encore

OS: Mach (distributed UNIX)

Language: C, Fortran, Ada

GUI: X-Windows

Supplier: Carnegie-Mellon Univ.

Contact: Zary Segall, (412) 268-3736

Local contact: Ann Patterson-Hine, (415) 604-4178

See reference 26 27 28
2.7 Parti

Function:

- Provides a set of procedures to be called from C or Fortran program that will translate read/write into send/receive (transform shared-memory programs into message-passing).
- Schedules the processes onto iPSC/860.

Evaluation by Doreen Cheng through phone contacts with the supplier:

- Load balancing must be done by the user.
- Supplier plans to extend this by adding higher level tools.

Platform: iPSC/860

OS: UNIX

Language: C, Fortran

Cost: Public domain

Supplier: NASA Langley ICASE

Contact: Joel Saltz, (804) 864-2197

Local access: Doreen Y. Cheng (415) 604-4361
2.8 Paragraph/PICL

Function:

- PICL generates execution profile of a parallel program on message passing machines.
- Paragraph allows visualization of the data collected.

Evaluation by Doreen Cheng through e-mail with the supplier:

- Tools could be integrated into our environment.

Platform: Intel and Ncube hypercubes, Symult/Ametek, Cogent

Language: C, Fortran

GUI: X-Windows

Cost: Public domain

Supplier: Oak Ridge National Lab

Contact: Michael T. Heath, mth@indigo.EPM.ORNL.GOV

Local contact: Doreen Cheng, (415) 604-4361
2.9 Hypertask

Function:

- Provides comment-directives for writing C programs for Intel hypercube machines.
- Provides library calls for dynamically resizing arrays by a factor of 2 each call.
- Automatically divides arrays and loop iterations among all nodes in a cube of any size at run time.
- Data locality is considered.
- Inserts message passing directives.
- Plots flops/processor in 3D (Sunview only).

Evaluation by Doreen Cheng through literature search and e-mail with supplier:

- The current version contains known bugs.

Platform: iPSC/860
OS: UNIX
Language: C
GUI: X-Windows, Sunview
Cost: Public domain
Supplier: Intel
Contact: Marc Baber, marc@isc.intel.com
Local contact: Doreen Cheng, (415) 604-4361

See reference 29
2.10 CRAY/fpp

Function:

- Automatic DO-loop parallelization
- Code transformation to take advantage of CRAY architecture

Evaluation by Doug Pase and Katherine Fletcher through use:

- The code generated for NAS benchmarks on Y-MP is 90% parallelized or less.
- User interface is batch oriented.
- Uses static program analysis only.

Platform: CRAY machines

OS: UNICOS, COS

Language: Fortran

GUI: None

Supplier: Cray Research

See reference 30 31
3. Tools Worth Tracking

3.1 PPD

Function:

- Event trace and post-mortem analysis for debugging
- Race condition detection
- Data flow and dependency analysis for debugging

Evaluation by Doreen Cheng through literature search and e-mail with the supplier:

- Part of it will be available at the end of summer 1991.

Supplier: Univ. of Wisconsin
Contact: Barton P. Miller, (608) 263-3378

See reference 32 33 34
3.2 Fortran-Linda

Function:

- Language extensions for parallel programming

Evaluation by Doreen Cheng through e-mail with the supplier:

- It is a proof-of-concept project not for distribution.

Platform: Encore

Contact: Nick Carriero, carriero-nicholas@CS.YALE.EDU
3.3 ParaScope/debugger

Function:

- Dependency analysis for debugging
- Static analysis of potential race conditions
- Instrumenting the code for race condition detection

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Not available yet

Supplier: Rice Univ.

Contact: Robert Hood, (713) 285-5182
3.4 Poker

Function:

- Graphic extension to C for parallel programming
- Trace and visualization of instruction execution through an emulator
- Visualization of data for debugging (Voyeur)

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Scaling is a problem since the process topology can only be defined statically.
- Poker project is replaced by a follow-up project ORCA which will incorporate the lessons learned.

Platform: iPSC (buggy), Sequent, Sun

OS: UNIX

Language: C

GUI: X11

Cost: $100

Supplier: Univ. of Washington

Contact: Larry Snyder, (206) 543-1695

See reference 35 36
3.5 CAPER

Function:

- Graphic language extension to C for parallel programming on message-passing machines
- A library of parallel algorithms
- A library of functions that convert distributed/parallel data structures into a target form
- Automatic generation of code to handle I/O
- Simple debugging facility

Evaluation by Doreen Cheng through phone contacts and e-mail with the supplier

- Not released yet
- Supplier will port to iPSC if required.

Platform: Sun, HPC (Bell Lab's machine)
OS: UNIX
Language: C
GUI: X-11
Cost: None (may change in the future)
Supplier: Bell Labs
Contact: Binay Sugla, (201) 949-0850
3.6 DINO

Function:

- C extension for data parallel programming
- Single program multiple data (SPMD)

Evaluation by Doreen Cheng through literature search and e-mail with the supplier:

- How well it works for CFD applications is not yet clear.
- No plan so far for Fortran.

Platform: Intel Hypercubes
OS: UNIX
Language: C
Cost: University distribution cost
Supplier: Univ. of Colorado
Contact: Bobby Schnabel, bobby@lupine.Colorado.EDU

See reference 37 38
3.7 Kali-Fortran

Function:

- Parallelizing compiler
- Using directives to express parallelism on message-passing machines

Evaluation by Doreen Cheng through phone contacts with the supplier:

- Prototype not completed yet

Supplier: NASA Langley ICASE
Contact: Joal Saltz, (804) 864-2197
3.8 FLO

Function:

- A graphic parallel programming language for scientific applications

Evaluation by Doug Pase:

- Proprietary design by Floating Point Systems
- Very attractive conceptually
- No compiler available

Supplier: None

Contact: Martin Waugh (503) 629-7651 (Designer of FLO)
3.9 OACIS Tools

Function:

- A graphic parallel programming language for scientific applications based on ELGDF.
- Generates code for C-Linda and Strand88.
- When target machine topology and the task graph of a program is entered as input, a schedule of the run-order of each task is produced.
- Analyzes a sequential program and saves the analysis in a database.

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Still in research prototyping stage.
- Using graphic user interface for Linda and Strand88 can be more intuitive for scientists than the languages' original user interface.
- Currently the tools are on Macintosh only.

Platform: Macintosh
Supplier: OASIS
Contact: Tony Capitano, capitano@sunny.oacis.org

See reference 39
3.10 Code/Rope

Function:

- Hierarchical graphic language to specify dependencies and firing rules of program components written in C or Fortran
- Insertion of parallel directives

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Is a research project.
- Could be used to learn the pros and cons of graphic parallel programming languages.

Cost: $200

Supplier: Univ. of Texas, Austin

Contact: James C. Browne, (512)-471-9579

See reference 40 41
3.11 KAP/CRAY

Function:

- Automatic DO-loop parallelization
- Code transformation to take advantage of CRAY architecture

Evaluation by Doug Pase and Katherine Fletcher through use:

- The code generated for NAS benchmarks on Y-MP is 90% parallelized or less.
- User interface is batch oriented.
- Uses static program analysis only.

Platform: Y-MP, X-MP, Sun, Vax

OS: UNICOS, COS, UNIX, Ultrix

Language: Fortran

GUI: None

Cost: First copy $7,500/yr
Add'l copy $3,750/yr
Site license $15,000/yr

Supplier: Kuck and Associates

Contact: Davida Bluhm, (217) 356-2288

See reference^{31}
4. Tools Eliminated

4.1 PISCES 2

Function:

- Extensions to Fortran for parallel programming
- Performance monitoring

Evaluation by Doreen Cheng through literature search and phone contacts with the supplier:

- Not on the platforms we have or anticipated having
- No plans for further development

Platform: FLEX/32 (shared-mem)
OS: UNIX
Language: Fortran
Supplier: University of Virginia
Contact: Terrence W. Pratt, (804) 982-2229

See reference 42 43
4.2 OLYMPUS

Function:

- Graphic extension to C for parallel programming

Evaluation by Doreen Cheng through literature search:

- Is an experimental system.
- Not on the platforms we have and anticipated having.
- Based on RPC.

Platform: Sun

OS: UNIX

Language: C

GUI: Sunview

Supplier: Univ. of Colorado

Contact: Garry J. Nutt

See reference 44
4.3 Triplex:

Function:

- Program instrumentation and visualization

Evaluation by Doreen Cheng through literature search:

- Not on the platforms we have and anticipated having

Platform: Sun, NCUBE
OS: SunOS
Languages: Unknown
GUI: Sunview
Cost: $250
Supplier: Tufts Univ.
Contact: David Krumme (author of the reference)

See reference 45
4.4 TOPSYS

Function:

- An object-oriented parallel programming tool
- Parallel debugging
- Performance monitoring
- Dynamic load balancing
- Process-processor mapping
- Program animation

Evaluation by Doreen Cheng through literature search:

- Unable to reach the supplier
- Out of US, difficult to collaborate

Platform: iPSC2
OS: Unknown
Language: C, Fortran
GUI: X11
Supplier: Univ. W. Germany
Contact: Thomas Bemmerl, bemmerl@lan.infomatik.tu-muenchen.dbp.de

See reference 48
4.5 IC*

Function:

- Specification language for parallel/distributed systems

Evaluation by Doreen Cheng through literature search:

- The language is difficult to use.
- The system is very slow (requires special purpose hardware).
- Designed for communication protocols.

Platform: Unknown
OS: UNIX
GUI: Unknown
Supplier: Bell Communication Research, Morristown, NJ
Contact: E. Jane Cameron (author of the reference)

See reference 47
4.6 VMMP:

Function:

- Language extension for programming on both shared-memory and message-passing machines
- Dynamic load balancing

Evaluation by Doreen Cheng through literature search:

- Has a routine call user interface.
- High efficiency claimed.
- Not able to contact the supplier.
- Out of US, difficult to collaborate.

Platform: Sun, MMX, IBM ACE

OS: UNIX, MACH

Language: C

Supplier: Tel-Aviv Univ., Israel

Contact: Eran Gabber (author of the reference)

See reference 48
4.7 Simple/Care

Function:

- Discrete event simulator
- Code instrumentation and visualization

Evaluation by Doreen Cheng through literature search:

- For object-oriented languages
- For hardware system design

Platform: Unknown
OS: Unknown
Language: Common Lisp
Cost: Public domain
Supplier: Stanford Univ.
Contact: Nakul Saraiya (author of the reference)

See reference 49
Acknowledgements

I would like to thank Doug Pase, Hal Barraclough, Horst Simon, and Russell Carter for helping with preparing this report.
References

4. Parasoft, Express Information Package.


<table>
<thead>
<tr>
<th>Functions of Parallel Programming Tools (Transform Sequential to Parallel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forge/MIMDizer</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Dependency Analysis</td>
</tr>
<tr>
<td>DO-Loop Parallelisation</td>
</tr>
<tr>
<td>Functional Parallelisation</td>
</tr>
<tr>
<td>Vectorisation</td>
</tr>
<tr>
<td>Directive Insertion</td>
</tr>
<tr>
<td>Automatic Code Transformation/Optimisation for Target Architecture</td>
</tr>
<tr>
<td>User Directed Code Transformation</td>
</tr>
<tr>
<td>Query about Program Info</td>
</tr>
<tr>
<td>Run Time Info Used for Guidance</td>
</tr>
<tr>
<td>Guidance in Parallelisation/Vectorisation</td>
</tr>
<tr>
<td>Guidance in Code Transformation</td>
</tr>
<tr>
<td>Guidance in Dependence Elimination</td>
</tr>
<tr>
<td>Program Profiling</td>
</tr>
<tr>
<td>Performance estimate</td>
</tr>
<tr>
<td>Parallelism Estimate</td>
</tr>
<tr>
<td>Use a Database</td>
</tr>
<tr>
<td>Interactive</td>
</tr>
<tr>
<td>Window-based GUI</td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Transform Sequential to Parallel Continued)</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dependency Analysis</td>
</tr>
<tr>
<td>DO-Loop Parallelisation</td>
</tr>
<tr>
<td>Functional Parallelisation</td>
</tr>
<tr>
<td>Vectorisation</td>
</tr>
<tr>
<td>Directive Insertion</td>
</tr>
<tr>
<td>Automatic Code Transformation /Optimisation for Target Architecture</td>
</tr>
<tr>
<td>User Directed Code Transformation</td>
</tr>
<tr>
<td>Query about Program Info</td>
</tr>
<tr>
<td>Run Time Info Used for Guidance</td>
</tr>
<tr>
<td>Guidance in Parallelisation /Vectorisation</td>
</tr>
<tr>
<td>Guidance in Code Transformation</td>
</tr>
<tr>
<td>Guidance in Dependence Elimination</td>
</tr>
<tr>
<td>Program Profiling</td>
</tr>
<tr>
<td>Performance estimate</td>
</tr>
<tr>
<td>Parallelism Estimate</td>
</tr>
<tr>
<td>Use a Database</td>
</tr>
<tr>
<td>Interactive</td>
</tr>
<tr>
<td>Window-based GUI</td>
</tr>
</tbody>
</table>

* Reduce message passing by relocate Linda run time routines
* For Shared-memory only
Functions of Parallel Programming Tools (Transform Sequential to Parallel Continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Hypertool</th>
<th>Schedule</th>
<th>Force</th>
<th>Parascope</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency Analysis</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DO-Loop Parallelisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Functional Parallelisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vectorisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directive Insertion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Code Transformation / Optimisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Directed Code Transformation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Query about Program Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run Time Info Used for Guidance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Parallelisation / Vectorisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Code Transformation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Dependence Elimination Program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Profiling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallelism Estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use a Database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window-based GUI</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Transform Sequential to Parallel Continued)</td>
<td>Axe/Aims</td>
<td>PIE</td>
<td>Parti</td>
<td>Paragraph/PICL</td>
<td>Hypertask</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Dependency Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO-Loop Parallelisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Parallelisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vectorisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directive Insertion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Code Transformation /Optimisation for Target Architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Directed Code Transformation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query about Program Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run Time Info Used for Guidance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Parallelisation /Vectorisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Code Transformation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Dependence Elimination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Profiling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance estimate</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallelism Estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use a Database</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window-based GUI</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Functions of Parallel Programming Tools (Transform Sequential to Parallel Continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>CRAY/lpp</th>
<th>KAP/CRAY</th>
<th>Poker</th>
<th>PPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency Analysis</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DO-Loop Parallelisation</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functional Parallelisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vectorisation</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directive Insertion</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Code Transformation/Optimisation for Target Architecture</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Directed Code Transformation</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query about Program Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run Time Info Used for Guidance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Parallelisation/Vectorisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Code Transformation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance in Dependence Elimination</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Profiling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallelism Estimate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use a Database</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Interactive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window-based GUI</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
## Functions of Parallel Programming Tools (Writing New Programs)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Forge/MIMDiser</th>
<th>Express</th>
<th>E/SP</th>
<th>VecPar_77</th>
<th>FATCAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language for Parallel Programming</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Data &amp; Loop Distribution</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Load Balancing</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency Checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Fortran 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Dynamic Resizing Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Parallel I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Function</td>
<td>Faust</td>
<td>IPS-2</td>
<td>Strand88</td>
<td>C-Linda</td>
<td>MONMACS</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------</td>
<td>-------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>Language for Parallel Programming</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Automatic Data &amp; Loop Distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Load Balancing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Frames</td>
<td>Faux</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency Checking</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Support for Fortran 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Resizing Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Writing New Programs Continued)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>Hypertool</td>
<td>Schedule</td>
<td>Force</td>
<td>Parascope</td>
<td>PAT</td>
</tr>
<tr>
<td>Language for Parallel Programming</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Data &amp; Loop Distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Load Balancing</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency Checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Support for Fortran 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Resizing Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language for Parallel Programming</td>
<td>Axe/Aims</td>
<td>PIE</td>
<td>Parti</td>
<td>Paragraph/PICL</td>
<td>Hypertask</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Automatic Data &amp; Loop Distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Load Balancing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency Checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Fortran 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Resizing Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Writing New Programs Continued)</td>
<td>CRAY/fpp</td>
<td>·KAP/CRAY</td>
<td>Poker</td>
<td>PPD</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Language for Parallel Programming</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Automatic Data &amp; Loop Distribution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Scheduling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Load Balancing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code Frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consistency Checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Fortran 90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Resizing Arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel I/O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Performance Tuning &amp; Debugging)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Forge/MIMDiser</td>
<td>Express</td>
<td>E/SP</td>
<td>VecPar_77</td>
<td>FATCAT</td>
</tr>
<tr>
<td>Performance Monitoring Visualization</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory Access Visualisation</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Path Analysis and Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication and Event Monitoring and Analysis</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Level Debugging</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Race Condition Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
Functions of Parallel Programming Tools (Performance Tuning & Debugging Continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Faust</th>
<th>IPS-2</th>
<th>Strand88</th>
<th>C-Linda</th>
<th>MONMACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Monitoring Visualization</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Memory Access Visualization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y*</td>
</tr>
<tr>
<td>Critical Path Analysis and Visualisation</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication and Event Monitoring and Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Source Level Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Race Condition Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Tuple space usage
<table>
<thead>
<tr>
<th>Functions of Parallel Programming Tools (Performance Tuning &amp; Debugging Continued)</th>
<th>Hypertool</th>
<th>Schedule</th>
<th>Force</th>
<th>Parascope</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Monitoring Visualisation</td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Memory Access Visualisation</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Path Analysis and Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication and Event Monitoring and Analysis</td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Level Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Prediction</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Race Condition Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Deadlock Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Performance Tuning &amp; Debugging Continued)</td>
<td>Axe/Aims</td>
<td>PIE</td>
<td>Pariti</td>
<td>Paragraph/PICL</td>
<td>Hypertask</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Performance Monitoring Visualisation</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Memory Access Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Path Analysis and Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication and Event Monitoring and Analysis</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Level Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functions of Parallel Programming Tools (Performance Tuning &amp; Debugging Continued)</td>
<td>CRAY/fpp</td>
<td>KAP/CRAY</td>
<td>Poker</td>
<td>PPD</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Performance Monitoring Visualisation</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Memory Access Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Critical Path Analysis and Visualisation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication and Event Monitoring and Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Level Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Race Condition Detection</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Deadlock Prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadlock Detection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Parallel Programming Tools (Status)

<table>
<thead>
<tr>
<th>Platform</th>
<th>Forre/MIMDiser</th>
<th>Express</th>
<th>E/SP</th>
<th>VecPar_77</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y-MP</td>
<td>ipSC</td>
<td>Sun</td>
<td>Y-MP</td>
</tr>
<tr>
<td></td>
<td>X-MP</td>
<td>NCUBE</td>
<td>CRAY*</td>
<td>Sun</td>
</tr>
<tr>
<td></td>
<td>CRAY2</td>
<td>Sun</td>
<td>IRIS***</td>
<td></td>
</tr>
<tr>
<td></td>
<td>iPSC</td>
<td>PC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sun</td>
<td>Macintosh</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRIS</td>
<td>Y-MP**</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NEC*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating system</th>
<th>UNIX</th>
<th>UNICOS</th>
<th>UNIX</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Languages</th>
<th>Fortran</th>
<th>Fortran 90</th>
<th>C</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GUI</th>
<th>X-Windows</th>
<th>X-Windows</th>
<th>X-Windows</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>Sunview</td>
<td>Sunview</td>
<td>Sunview</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Postscript</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maturity</th>
<th>Bugs Fixed</th>
<th>Unknown</th>
<th>Unknown</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quickly</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost</th>
<th>$28,050/wkst</th>
<th>$3,000/IPSC/860</th>
<th>TBD</th>
<th>$2,500/wkst</th>
</tr>
</thead>
<tbody>
<tr>
<td>mnt:</td>
<td>$4,500/yr</td>
<td>$15,000/Y-MP</td>
<td>$19,500/Y-MP</td>
<td></td>
</tr>
<tr>
<td>yr:</td>
<td>$93,500/10 wkst</td>
<td>$1,500/Network of Suns</td>
<td>Site License</td>
<td>$33,510/yr</td>
</tr>
<tr>
<td>yr:</td>
<td>$93,500/CRAY</td>
<td>mnt:20% / yr</td>
<td>$78,250 Perpetual</td>
<td></td>
</tr>
<tr>
<td>yr:</td>
<td>$93,500/ipSC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yr:</td>
<td>Site License</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnt:</td>
<td>$158,950</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnt:</td>
<td>$18,000/yr</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supplier</th>
<th>PSR</th>
<th>ParaSoft</th>
<th>SES</th>
<th>NAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>John Levesque</td>
<td>Adam Kolawa</td>
<td>J. C. Browne</td>
<td>Sheila Caswell</td>
</tr>
<tr>
<td></td>
<td>(916) 621-1600</td>
<td>(818) 792-9941</td>
<td>(512) 474-4526</td>
<td>(708) 971-2337</td>
</tr>
</tbody>
</table>

wkst: workstation  
mnt: maintenance  
yr: year  
TBD: to be determined  
* in the process of negotiation  
** beta test  
*** port if desired
<table>
<thead>
<tr>
<th>Platform</th>
<th>FATCAT</th>
<th>Faust</th>
<th>IPS-2</th>
<th>Strand88</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRAY2</td>
<td>CRAY</td>
<td>Y-MP*</td>
<td>Y-MP</td>
<td></td>
</tr>
<tr>
<td>Y-MP</td>
<td>UNIX</td>
<td>UNIX</td>
<td>iPSC</td>
<td></td>
</tr>
<tr>
<td>X-MP</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Sequent</td>
<td></td>
</tr>
<tr>
<td>Sun</td>
<td>None</td>
<td>C</td>
<td>Encore</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sun</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating system</th>
<th>UNIX</th>
<th>UNIX</th>
<th>UNIX</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Fortran</td>
</tr>
<tr>
<td>GUI</td>
<td>None</td>
<td>X-Windows</td>
<td>X-Windows</td>
<td>X-Windows</td>
</tr>
<tr>
<td>Maturity</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Cost</td>
<td>$50,000/source</td>
<td>$100/source</td>
<td>$300/source</td>
<td>$22,000/IPSC/860</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mnt: $3,375/yr</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mnt: $3,000/Sun</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Site License:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$30,000-$40,000</td>
</tr>
<tr>
<td></td>
<td>Site License:</td>
<td></td>
<td></td>
<td>30% GOV dscnt</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supplier</th>
<th>NEAT</th>
<th>U. of Illinois</th>
<th>U.of Wisconsin</th>
<th>Strand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>David Klapphols</td>
<td>David Hammerslag</td>
<td>Barton Miller</td>
<td>Timothy Mattson</td>
</tr>
<tr>
<td></td>
<td>(201) 420-5509</td>
<td>(217) 244-0277</td>
<td>(608) 263-3378</td>
<td>(503) 690-9830</td>
</tr>
</tbody>
</table>

mnt: maintenance  
yr: year  
dscnt: discount  

* Available in March 1991
<table>
<thead>
<tr>
<th>Platform</th>
<th>C-Linda</th>
<th>MONMACS</th>
<th>Hypertool</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-MP*</td>
<td>iPSC/860**</td>
<td>iPSC/2</td>
<td></td>
<td>CRAY2</td>
</tr>
<tr>
<td>IPSC/860**</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encore</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM RS/6000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apollo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>UNIX</td>
<td>UNIX</td>
<td>UNIX</td>
<td>UNIX</td>
</tr>
<tr>
<td>Languages</td>
<td>C</td>
<td>Fortran***</td>
<td>C</td>
<td>Fortran</td>
</tr>
<tr>
<td>GUI</td>
<td>X-Windows</td>
<td>X-Windows</td>
<td>X-Windows</td>
<td>X-Windows</td>
</tr>
<tr>
<td>Maturity</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Cost</td>
<td>$7,000/10wkst</td>
<td>None</td>
<td>Univ. Distribution</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>$20,000/ipSC2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Site License for ARC: $90,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supplier</td>
<td>SCA</td>
<td>ANL</td>
<td>UC Irvine</td>
<td>U of Tenn.</td>
</tr>
<tr>
<td>Contact</td>
<td>Ellen Smith (203) 777-7442</td>
<td>Ewing Lusk (708) 972-7852</td>
<td>Daniel Gajski (714) 858-4155</td>
<td>Jack Dongarra (815) 972-8295</td>
</tr>
</tbody>
</table>

wkst: workstation
mnt: maintenance
yr: year
dscnt: discount
TBD: to be determined

* not fully debugged
** will be available soon
*** not directly supported, can be called from C
**** available in March 1991
<table>
<thead>
<tr>
<th>Platform</th>
<th>Force</th>
<th>Parascope</th>
<th>PAT</th>
<th>Axe/Aims</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-MP</td>
<td></td>
<td>Sun</td>
<td>CRAY</td>
<td>iPSC/860</td>
</tr>
<tr>
<td>CRAY2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encore</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convex</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alliant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>UNIX</td>
<td>UNIX</td>
<td>UNIX</td>
<td>UNIX</td>
</tr>
<tr>
<td>Languages</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Fortran</td>
</tr>
<tr>
<td>GUI</td>
<td>X-Windows</td>
<td>X-Windows</td>
<td>X-Windows</td>
<td></td>
</tr>
<tr>
<td>Maturity</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Cost</td>
<td>None</td>
<td>Univ. Dist</td>
<td>Univ. Dist</td>
<td>None</td>
</tr>
<tr>
<td>Supplier</td>
<td>U. of Colo.</td>
<td>Rice U.</td>
<td>GIT</td>
<td>NASA ARC</td>
</tr>
<tr>
<td>Contact</td>
<td>Jerry Jordan</td>
<td>Ken Kennedy</td>
<td>Kevin Smith</td>
<td>Jerry Yan</td>
</tr>
<tr>
<td></td>
<td>(303) 492-1411</td>
<td>(713) 285-5188</td>
<td><a href="mailto:smith@boa.gatech.edu">smith@boa.gatech.edu</a></td>
<td>(415) 804-4381</td>
</tr>
<tr>
<td>Platform</td>
<td>PIE</td>
<td>Parti</td>
<td>Paragraph/PICL</td>
<td>Hypertask</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Sun</td>
<td>iPSC/860</td>
<td>iPSC</td>
<td>Neube</td>
<td>iPSC/860</td>
</tr>
<tr>
<td>Encore</td>
<td></td>
<td></td>
<td>Ametek</td>
<td></td>
</tr>
<tr>
<td>VAX</td>
<td></td>
<td></td>
<td>Cogent</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating system</td>
<td>Mach</td>
<td>UNIX</td>
<td>UNIX</td>
<td>UNIX</td>
</tr>
<tr>
<td>Languages</td>
<td>Fortran</td>
<td>Fortran</td>
<td>Fortran</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ada</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td>X-Windows</td>
<td></td>
<td>X-Windows</td>
<td>X-Windows</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sunview</td>
</tr>
<tr>
<td>Maturity</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Cost</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Supplier</td>
<td>CMU</td>
<td>NASA Langley</td>
<td>ORNL</td>
<td>Intel</td>
</tr>
<tr>
<td>Contact</td>
<td>Zary Segall</td>
<td>Joel Salts</td>
<td>Michael Heath</td>
<td>Marc Baber</td>
</tr>
<tr>
<td></td>
<td>(412) 268-3736</td>
<td>(804) 864-2197</td>
<td><a href="mailto:mth@indigo.epm.ornl.gov">mth@indigo.epm.ornl.gov</a></td>
<td>marc@isc,intel.com</td>
</tr>
<tr>
<td>Platform</td>
<td>CRAY/fpp</td>
<td>KAP/CRAY</td>
<td>Poker</td>
<td>PPD</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>CRAY</td>
<td>CRAY</td>
<td>Sun</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td>Sequent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vax</td>
<td></td>
<td>iPSC*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating System</td>
<td>UNICOS</td>
<td>UNIX</td>
<td>UNICOS</td>
<td>UNIX</td>
</tr>
<tr>
<td></td>
<td>COS</td>
<td>UNICOS</td>
<td>COS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ULTRIX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Languages</td>
<td>Fortran</td>
<td>Fortran</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>GUI</td>
<td></td>
<td></td>
<td></td>
<td>X-Windows</td>
</tr>
<tr>
<td>Maturity</td>
<td>Bugs Fixed</td>
<td>Bugs Fixed</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td>Cost</td>
<td>CRAY Compiler</td>
<td>1st copy: $7,500/yr</td>
<td>Univ. Dist</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add'l: $3,750/yr</td>
<td></td>
<td>Univ. Dist.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Site License: $15,000/yr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supplier</td>
<td>CRAY</td>
<td>K&amp;A</td>
<td>U. of Washington</td>
<td>U. of Wisconsin</td>
</tr>
<tr>
<td>Contact</td>
<td></td>
<td></td>
<td>Davida Bluhm</td>
<td>Barton Miller</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(217) 356-2288</td>
<td>(608) 263-3378</td>
</tr>
</tbody>
</table>

yr: year
Add'l: additional copy

* buggy