Center for Scalable Application Development Software: Management Plan

1.1 Overview

Management of the Center will be the overall responsibility of the Principal Investigator/Project Director, Ken Kennedy of Rice University. In carrying out that responsibility he will be assisted by an *Executive Committee* chaired by the PI and consisting of all the co-investigators:

Rice University:
- Ken Kennedy, Lead PI and Chair
- John Mellor-Crummey
- Keith Cooper

Argonne National Laboratory:
- Peter Beckman
- William Gropp
- Ewing Lusk

University of California, Berkeley
- Katherine Yelick

University of Tennessee, Knoxville
- Jack Dongarra

University of Wisconsin Madison
- Barton Miller

The Executive Committee will be responsible for developing and implementing the vision for all Center programs in outreach, research, and infrastructure development. The investigators have a 17-year track record of successful collaboration following this model, beginning with the Center for Research on Parallel Computation.

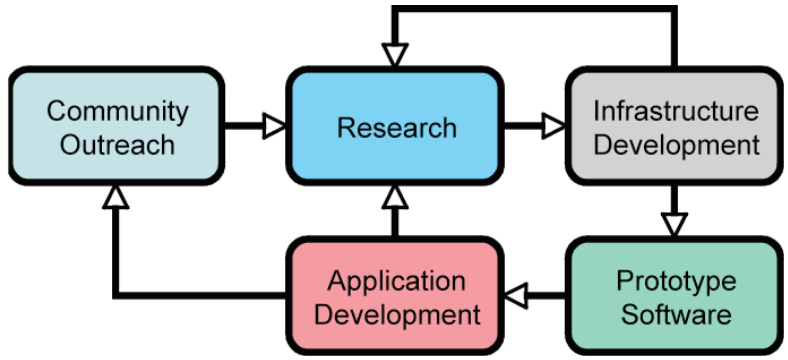
The community outreach and vision-building efforts will be the responsibility of the *Outreach Steering Committee*, which includes all of the Summer Institute Program Directors (Beckman, Dongarra, Kennedy, Miller, Yelick). Argonne National Laboratories, under the direction of the *Outreach Program Director*, Peter Beckman, will handle day-to-day management of the outreach activities.

Each of these committees will meet regularly by teleconference and they will meet jointly at least once a summer during a full project meeting co-located with the Summer Institute. The Executive Committee will meet at least once every two weeks at a regularly scheduled time (currently Friday at 3pm Central time).

Rice University will be responsible for the management of the infrastructure building efforts, which will be carried out by a staff located at Rice. The *Software Development Manager,* reporting to the Project Director, will oversee the infrastructure-building activities.

1.2 Interrelationships Among Center Activities

The diagram below illustrates the relationships among the Center activities. The flow of ideas originates from two sources: the community outreach and vision-building workshops and direct collaboration with application development. These activities focus research efforts on important problems. In turn, research drives the infrastructure development by identifying capabilities that are needed to support the long-range vision. Infrastructure feeds back into the research program, but it can also lead to prototype software tools and programming systems that support further application development. Finally, the developers report their experiences and difficulties with the available tools and systems through the Summer Institutes, and the cycle starts over again.



1.3 Metrics for Success

An important part of the management of CScADS is establishing measures of success without sacrificing the flexibility to rapidly respond to the needs of the community. Although there is an extensive list of milestones and deliverables detailed in Section 1.5, we expect there will be some changes in the plan over the five years of the project. At an abstract level, success will be measured by the answers to the following questions:

1. How well are the Summer Institute workshops functioning as a mechanism for two-way exchange of information: (1) as a way of familiarizing technology consumers with new developments and (2) as a driver for change in the research and development plans?
2. How effectively is the research adapting to the needs of the developer community while keeping quality at the highest possible level?
3. How effectively is the research effort directly interacting with application developers to understand their problems and using the solution strategies to influence future directions?
4. How effectively is the infrastructure development effort supporting the entire HPC software research community, both within and outside CScADS?
5. How effective is CScADS at spinning out intermediate results as prototype tools and software for end users and at influencing commercial software products?
6. How well is CScADS facilitating scalability of applications on the DOE Leadership Class Facilities?

Any effective measurement strategy must provide the data to answer these questions. For the first question, we will rely on three mechanisms: (1) surveys of the workshop attendees, which will be summarized in the *Summer Institute Report*; (2) a review of the recommendations of each workshop to be included in the *CScADS Annual Report*, along with a proposed set of changes in research direction within the Center; and (3) a review at the annual meeting, one year later, of the impact of the previous years' recommendations on research within CScADS.

For the second question, CScADS will produce an annual *Research Summary* that will detail the changes in direction in response to both the workshop reports and the direct interactions with applications (part of question 3). We will assess research quality using the traditional academic measures of papers produced, students graduated, and awards received. Both of these measures will be documented in the CScADS Annual Report.

The third question will be answered in a *Collaboration Report* that will discuss the ongoing interactions with application developers and other technology consumers and how those interactions (1) directly helped the application and (2) led to new technology to overcome barriers for future applications. In most cases, the latter goal will receive a higher priority than the former, because it is critical that computer science efforts focus on solving problems for whole classes of applications rather than simply helping one application at a time.

The fourth question will be addressed by a survey of the projects using the CScADS Open Software Suite along with a summary of how well the suite is meeting the needs of the research community and of how flexibly the development effort is responding to evolving requirements. Note that the Open Software Suite will be released to the community under open-source licenses, consistent with DOE SciDAC policies.

The fifth and sixth questions will be answered by a *Technology Transfer Survey* of prototype tools, their usage (including installations at SciDAC leadership sites), and the number of commercial products influenced by the research and development. Once again, this will be a standard part of the Annual Report.

In summary, answers to the six questions will be continuously monitored and published each year as a part of the CScADS Annual Report to DOE. In addition, all of the outputs from the Summer Institutes will be included in a separate report produced by the workshop leaders and edited by Outreach Steering Committee under the leadership of the Outreach Program Director.

In a sense, the questions above are all designed to ask how well CScADS is executing the influence diagram in Section 1.2 below. In the end, that is the true determination of whether CScADS is achieving its goals.

1.4 Baseline

At the start of our activities we have a number of prototype software systems and tools that can serve as a basis for our work. Here is a partial inventory:

- *HPCToolkit*, a performance measurement and analysis system based on sample-based profiling.
- *Dyninst,* a program analysis and instrumentation suite. We will leverage the current Dyninst effort to spin-off components that support capabilities such as stack walking and other related logic.
- The *D System*, a compiler infrastructure for source-to-source transformation based on dependence analysis. Currently it is used in our work on HPF compilation, transformations and automatic tuning for complex memory hierarchies.
- The *Open64 Compiler Infrastructure* originally released by SGI, and widely used in the community. It is used as basis for the Berkeley UPC and Rice Co-array Fortran compilers.
- The *Telescoping Languages Infrastructure* for precompilation and specialization of libraries. The current version includes a constraint-based type analysis system for Matlab and supports libraries and scripts written in Matlab. However it is completely general and can be used for other high-level languages.
- *KOJAK*, a toolkit for performance analysis of parallel applications. KOJAK components support instrumentation, postprocessing of performance data, and result presentation. KOJAK places particular emphasis on automating the transformation of performance measurements into a high-level view of performance behavior. A key part of the software is an integrated event-trace analysis environment for MPI and OpenMP applications.
- *PAPI*, a cross-platform interface to hardware counters available on modern microprocessors. PAPI consists of an application programming interface for accessing hardware counters, and support for a standard set of performance metrics for application performance evaluation.
- The *Berkeley UPC compiler,* a portable compiler translates UPC into C with calls to the GASNet communication layer. Is uses the Open64 compiler infrastructure and performs source-to-source optimizations in addition to translating the code.

- The *Rice Co-array Fortran compiler*, a portable compiler translates Co-array Fortran (CAF) into Fortran 90 with calls to the GASNet or ARMCI communication layers. Is uses the Open64 compiler infrastructure and performs source-to-source optimizations in addition to translating the code.
- The *General Code Optimization (GCO)* framework extends the idea of code generation and search-based optimization used in ATLAS and FFTW to more general code sequences and a larger set of transformations. Since the software segment to be optimized is not known in advance, GCO requires compiler technology (which is based on the ROSE system from Lawrence Livermore National Laboratory) to analyze the source code and generate the candidate implementations.

These tools are, for the most part, research prototypes and not yet suitable for use by end users. Furthermore, none has been applied at petascale.

In five years we plan to have a collection of software tools called the *CScADS Open Software Suite*, which will include mature versions of these infrastructures and a collection of tools based upon them that are robust enough for end-user use. This collection will support scaling across all three dimensions described in the proposal: (1) high-level language to efficient uniprocessor implementation, (2) uniprocessor to large-scale parallel configuration, and (3) between different architectures, including heterogeneity.

The HPC community is just beginning to grapple with the complexity of petascale computing. The machines contemplated for leadership-class facilities feature significantly different approaches to design. We expect these differences to become more pronounced in the future. In five years, we plan to have programs in place, through our series of workshops and summer institutes, that will help the application development community understand and effectively utilize these systems at scale.

Finally, we are limited in our understanding of scalability itself, much less scalability to the extremes required over the coming years. We plan to address this limitation by focusing on applications and what is needed to make them scale. This will, in turn, illuminate our research on tools by showing us what we need to automate to make petascale application development easier.

1.5 Milestones

- Year 1:
    - Prototype implementation of the telescoping languages infrastructure. Demonstrate on a small application written in Matlab.
    - Release HPCToolkit performance tools as part of the CScADS Open Software Suite. This release will incorporate any new measurement capabilities developed as part of the PERI Institute augmented with a new user interface for call path based analysis of execution performance.
    - Define requirements for performance analysis tools including analysis product data, instrumentation interfaces, stack-walk facilities, and code mapping data.
    - Release plan for CCA-based optimizing component integration framework.
    - Develop plan and prototype mapping of Fortran 90 to a multicore chip used in one of the leadership-class facilities with a focus on choreographing the movement of data on and off chip and the partitioning of computation across the on-chip cores.
    - Release Rice Co-array Fortran compiler and the Berkeley UPC compiler as part of the CScADS Open Software Suite.
    - Port prototype CScADS Open Software Suite to leadership machines at ANL, ORNL, and NERSC.

- Collect performance information for key benchmarks and applications to provide baseline for improvements.
  - Explore the design space of interactive environments, such as Matlab, Python, and Mathematica, and the consequences of particular design choice.
  - Summer Workshops: Improve connection between application teams and computer scientists; introduce tool sets; understand immediate scaling problems.
- Year 2:
  - Demonstrate performance analysis of multicore chips with HPCToolkit, including metrics for understanding interactions between multiple cores competing for memory bandwidth.
  - Define standard formats and classes for code analysis products, both from source and binary analysis. These products will include abstract assembly code, basic blocks, control flow graphs, call graphs, and data flow information. The classes will include the ability to export and import such products.
  - Define standard interfaces for code instrumentation for both source and binary.
  - Refine parallel Matlab implementation and demonstrate on small benchmarks.
  - Create simplified tool environment for leadership-class machines.
  - Release toolkit for analysis of distribution-based global view languages. This release will contain a more robust version of research prototype software developed for the Rice dHPF compiler.
  - Produce a study of the scalability of applications written with the CScADS Software suite on the leadership machines. Explore scalable parallelization of scripting languages such as Matlab and Python.
  - Explore dynamically making decisions on how to solve the user's problem by coupling the cluster state information with knowledge of the particular application.
  - Release new version of CScADS Open Software Suite.
  - Summer Workshops: Explore next steps to extreme scalability, analyze current roadblocks; cross-connect with other SciDAC centers and institutes; explore multi-core code generation.
- Year 3:
  - Demonstrate compilation-based integration on a CCA-compliant component library (e.g., Trilinos) on a simple application.
  - Demonstrate advanced compiler optimizations for multicore chips including techniques for exploiting on-chip parallelism and improving cache cache reuse.
  - Stress-test tool set, and compare baseline machine and application performance against improvements.
  - Develop a study of and optimizations for global address space applications written with fine-grained sharing (e.g., those developed for the Cray X1 or MTA machine) to allow them to run on a broader class of architectures.
  - Explore algorithm detection at runtime.
  - Investigate search space optimization, automatic search space generation, and expand to heterogeneous clusters.
  - Release new version of CScADS Open Software Suite.
  - Summer Workshops: Explore initial performance of newly deployed peta-scale machine upgrades, scalability challenges, and future plans for applications. Present compiler and language tool sets.

- Year 4:
  - Demonstrate efficacy of telescoping languages on an AMR library.
  - Demonstrate autotuning of multi-loop application using at least five critical transformations.
  - Integrate analysis and run-time support for user-defined data distributions for HPCS global view languages developed under PMODELS-2 into CScADS Open Software Suite.
  - Demonstrate decomposition of F90 loop nests onto heterogeneous computing elements.
  - Analyze productivity and performance improvements; explore new techniques to simplify application tuning.
  - Develop optimizations overlapping and pipelining communication for the PGAS languages on the leadership machines.
  - Construct algorithm selection based on input data with heuristic choice.
  - Develop framework for self-adaptation of numerical libraries on clusters.
  - Release new version of CScADS Open Software Suite.
  - Summer Workshops: Explore HPCS technology; Evaluation of current applications using UPC, CAF, and telescoping languages.
- Year 5:
  - Report on relative performance of scripting languages, data parallel languages, and partitioned global array languages (all versus MPI) on leadership-class machines.
  - Demonstrate compilation for large-scale machines using both multicore chips and heterogeneous computing elements.
  - Demonstrate effectiveness of compiler-based component integration on a significant DOE application.
  - Implement design for self-adaptation of numerical libraries on clusters.
  - Report on the scalability of the PGAS languages on the leadership machines
  - Release new version of CScADS Open Software Suite.
  - Summer Workshops: Evaluate current state of petascale computing and tools for application development; develop roadmap for future productivity and tool needs.

1.6 Software

An important component of CScADS activities will be the development of software infrastructures to support research and to build usable tools for application and library developers. To support this activity, we have included budget for two software developers at Rice and one at University of Wisconsin, Madison, the University of California at Berkeley, and at the University of Tennessee (Argonne).

The main mission of these developers will be to support open-source software infrastructures that will be used to foster research in the community, prototype tool building, and standardization of tool interfaces. Among the infrastructures that we have selected for possible support are Rice HPCToolkit, Open64 Compiler Infrastructure, the Rice D System (a source-to-source transformation system), the Berkeley UPC compiler, the Rice CAF compiler and Palomar (the Rice Telescoping Languages infrastructure).

By policy, we will only support open-source infrastructures, so the intellectual property issues should be minimal.

1.7 Reporting

As described in Section 1.3 above, we plan to produce a detailed annual report on each year's activities, including research, meetings, outreach, and requirements analysis. Six months into each year, we will also provide an interim report to DOE on the status of activities during that period. Each year, the interim report will be incorporated into, and subsumed by, the annual report. The annual report will be delivered by October 1 each year, while the interim report will be delivered by April 1.

1.8 Relationships to Other Centers and Institutes

CScADS will seek to establish strong relationships with several other SciDAC Centers and Institutes. Here is a list of such and a brief statement on how we plan to interact.

- *Performance Engineering Research Institute (PERI).* Perhaps our strongest relationship is with PERI because there is some overlap of personnel and research goals. We plan to jointly hold workshops and conduct applications surveys. Some of the CScADS software, such as HPCToolkit and PAPI, are supported by PERI as part of their "triage" process to determine tuning opportunities. CScADS will collaborate with PERI in the area of performance tool infrastructure and autotuning.
- *Center for Technology for Advanced Scientific Component Software (TASCS).* Our work with TASCS will focus on use of the telescoping languages infrastructure as an efficient component integration system. We plan to have regular contact with the leadership and conduct at least one joint workshop.
- The *Applied Partial Differential Equations Center for Enabling Technologies (APDEC).* The APDEC project is focused on infrastructure for and applications of Adaptive Mesh Refinement technology (AMR). We plan to work directly with the APDEC team on the use of CScADS software for performance analysis, tuning, and scaling of AMR and will hold a joint workshop on the performance of AMR computations on large-scale machines.
- *Towards Optimal Petascale Simulations (TOPS).* The TOPS project is developing scalable solvers, including direct and iterative sparse matrix methods. TOPS and CScADS will hold a joint workshop on performance analysis, tuning, and scaling of sparse matrix solvers.
- *Center for Programming Models for Scalable Parallel Computing (PModels*). The PModels Center is a DOE base program project developing new programming models, including PGAS languages and libraries. CScADS and PModels will collaborate on the use of Open64 compiler infrastructure and will hold a joint workshop to introduce the applications community to advanced programming models.
- *Center for Interoperable Technologies for Advanced Petascale Simulations (ITAPS).* CScADS and ITAPS will hold at least one joint workshop to discuss problems of shared interest. Topics may include the use of telescoping language approaches to improve the performance of the ITAPS-defined interfaces and the creation and evaluation of challenge problems for compiler-based optimization, particularly for multicore platforms.
- *Combinatorial Scientific Computing and Petascale Simulations (CSCAPES).* CScADS and CSCAPES will hold at least one joint workshop on problems of common interest. There are both opportunities for joint research, such as exploring techniques for compilation and automatic differentiation, and complementary research, such performance tuning for sparse-matrix algorithms on multicore platforms.
- *Petascale Data Storage Institute.* CScADS will collaborate with the Petascale Data Storage Institute to explore opportunities for application of compiler-based tools and new programming models to expose opportunities for improving the performance of I/O-intensive applications. We will drive this collaboration by jointly working with a DOE application that exhibits performance issues in both computation and I/O.

1.9 Communication with Other Centers, Institutes, and Applications

Our main vehicle for communication with other Centers, Institutes and with application groups will be the workshops and summer institutes we plan to hold. By bringing application groups, library developers, and computer scientists together to discuss challenges, we can show how the work we have done applies to current problems and discover new directions for our research and development.

In addition to these, we plan to conduct a survey of application groups in a joint effort with PERI.

Finally, we will arrange direct collaborations with a few applications that will drive our early research. With these, we will have face-to-face meetings with the developers and regular telecoms to monitor progress in developing support tools.

We also plan to have telecoms as needed with other Centers and Institutes.

1.10 Workshop Plan

The Summer Institute workshops will be 3-5 days long each and will be of three general types:

Focused Tutorials: In these hands-on workshops, the speakers will be experts in the appropriate fields and the attendees will be application developers, particularly those who design and write code. Topics covered will be chosen to accelerate development of scalable applications in general. Some potential examples are:

- User of math libraries such as ScaLAPACK, PETSc, AMR libraries, meshing libraries, etc.
- Beyond MPI: implementation methodologies using OpenMP, Pthreads, PGAS languages, and in later years HPCS languages.
- Problems and solutions for large data: parallel file systems, strategies for dealing with petascale data management and processing.
- Software engineering for scientific computing

Application Area Conclaves: Attendees and speakers will be the same, drawn from practitioners in a particular application area in which problems and solutions can be shared. Some potential examples are:

- Climate modeling
- Combustion
- Astrophysics
- Materials science
- Biomolecular modeling
- Biological sequence analysis

Mathematics and Computer Science Futures: Attendees and speakers will be the same, drawn from expert mathematicians and computer scientists, with a specific mission to look beyond the next few years to help structure a long-range research agenda in a particular critical area. Some examples are:

- Programming models and languages
- Standard interfaces for tools

- New algorithm research
- Workflows for scientific computing
- Network-enabled applications
- Exoscale computing challenges in hardware and software
- Whither the Grid?
- Fault tolerance approaches in applications and systems

Other types of workshops will no doubt emerge with experience. We will endeavor to schedule these so that multi-week residences for attendees will make sense.

Finally, we are leaving open the possibility of focused workshops that will occur outside the Summer Institute to achieve specific goals. These workshops would typically be shorter and would usually involve some special topic on which we might want a high-bandwidth interaction with other centers, application groups or members of the computer science community. An example would be the one-day annual workshop on automatic tuning at the Santa Fe HPC meeting (previously the LACSI Symposium), with which several of the PIs have been involved in the past.

1.11 Broad-based Outreach

Our broad based outreach will be carried out through two main mechanisms. First we will maintain a Web Site that will have areas for application developers, library developers, computer scientists, and the CScADS team. This web site will also be used as a vehicle for communication with the general public and DOE program managers.

In addition, we will use our workshops and summer institutes for broad-based outreach.