SciDAC Application Computing Needs Survey

This survey is designed to understand the computing needs of the SciDAC Applications to help meet your technical objectives. Responses to this survey will help guide the resource management plan for SciDAC. In particular, the answers will provide a first cut at:

- *Identifying critical communications among the applications, Applied Math, Computer Science and Collaboratory groups in SciDAC*
- *Helping to understand and detail requirements for the use of SciDAC hardware, software, and connectivity.*

Feel free to answer "I don't know" or "not applicable," or to provide other information that you think may be helpful. Partial answers are more helpful than no answers at all.

Results of the survey will be available at the SciDAC PI meeting in September, and implications of the data gathered from the survey will be discussed at that time.

The survey should be completed for each SciDAC project (one survey per project) and returned electronically to romine@er.doe.gov or by FAX to Chuck Romine, SC-31 at (301) 903-7774 before 8 September 2001 to allow time to collate and analyze the results prior to the SciDAC PI meeting.

Questions regarding the survey can be directed to Chuck Romine (romine@er.doe.gov or (301) 903-5152).

Project Name: Particle Physics Data Grid Collaboratory Pilot

Responding Contact: Ruth Pordes, Chair PPDG Steering Committee

A: Understanding the primary goals and objectives of the application

The purpose of this section is to help understand the project deliverables that are most critical to make your project successful. Subsequent sections will link your computing needs to these deliverables, so we can determine which technologies will be needed and when they need to be in place.

1. Give a brief description of your application. Include a URL to a project page if you have one.

The Particle Physics Data Grid Collaboratory Pilot extends, integrates and deploys grid middleware for and in collaboration with the participating High Energy and Nuclear Physics experiments – ATLAS, BaBar, CMS, D0, STAR and TJNF. Well defined Project Activities, collaborative developments between an Experiment and one of the Computer Science Groups on the project – Condor, Globus, SRNB,

SRM - develop and extend the software to facilitate the computational and data grids as part of the Experiment data handling system.

Project web Page: http:://www.ppdg.net

2. Identify major project milestones and the target completion date for each

Each Experiment Team Lead is developing PPDG project milestones aligned with thos of the experiment itself, to ensure maximal likelihood that the deliverables from PPDG will be integrated with and used by the Experiment. PPDG milestones for the first .5->1 year are prepared and are being made available in the next 2 weeks as part of the Year 1 Project Plan and project overview.

3. Provide any information that you feel will clarify what you expect to accomplish in this application.

The goals of the project are to acquire, develop and deploy reusable grid middleware that is used by all the experiments that are part of the PPDG Collaboratory Pilot. The project facilitates cross-experiment sharing of requirements and operational experience and stimulates "Cross-cut" activities involving all the Experiments and Computer Science groups towards the development of common and shared software and Grid middleware services.

B: Understanding the initial state of the computational application

The purpose of this section is to understand the starting point for your code development efforts. This information will also be used to track the influence that SciDAC has on the advancement of simulation capabilities.

1. Identify major computational modules of the current code system. Are they currently running as pieces of one large system, or will they need to be merged/integrated as a part of the application development?

Each of the six experiments has or will have a comprehensive data handling and data processing model. These systems are developed and supported by much Experiment specific teams than are participating in the PPDG Collaboratory. (see flower picture at http://www.ppdg.net). Each of the Computer Science Groups in PPDG represents the effort of a larger team doing research and development of key Grid middleware which is used by a much larger community than that in PPDG.

2. Identify the programming language in which each of these computational modules is written. What parallel programming models (if any) are currently used?

Each experiment and computer science group is responsible for the modular code developed. One of the goals of PPDG is to promote and define well defined interfaces between the modules and provide applications that test the integration of the set of modules in the Experiment data handling systems. Thus a mix of

programming languages: C, C++, Python, perl, sh, (maybe even Fortran?!), and Java are used.

3. Identify major numerical algorithms currently employed.

PPDG does not include the development of numerical algorithms

4. What programming environment tools (editors, source code control, debuggers, profilers, etc.) were employed in the development of these modules?

The programming tools used for development in PPDG are the responsibility of the individual development groups. We use CVS for source code and documentation control. Leak checkers such as purify and insure are used on invidual code modules but are not procured or supported as part of the PPDG project.

5. On what computing platforms do the current code modules run (computer systems, operating systems, compilers…)? Please specify the size of problems you can run, execution time required, I/O needs, and mass storage use.

All PPDG code is developed for cross-platform portability. It must run on the independent experiment systems which are decided on outside of the Collaboratory Pilot project. In general codes run on Solaris, AIX, IRIX and Linux

6. Do you have any special needs for wide-area network access to run your current application code?

Yes. All experiment participants in PPDG are global collaborations which require wide area distribution of the data and the application code. This is done using existing tools, and through experiment policy and organization today. It is hoped that this is done through the Grid middleware tools and through automation of the experiment policies in the future.

Part C: Understanding your code development plans for meeting your objectives

The purpose of this section is to understand the major code development efforts planned for your project and how they relate to your project objectives. In answering these questions, please try to link the various code development activities, so we can better understand your timetable for various needs.

1. Identify all of the major computational modules you expect to have in the final code system. Please explain what kind of revisions you expect to make to existing ones and what capabilities you want in new ones.

We will supply this but not before September 8th. Suffice it to say that all layers in the Data Grid architecture will be included in the Experiment applications codes. These modules are defined by the indiviual Experiment data handling system architectures. The following papers describe them:

ATLAS - http://atlassw1.phy.bnl.gov/magda/info

http://www.ihep.ac.cn/~chep01/paper/10-013.pdf

CMS - http://kholtman.home.cern.ch/kholtman/cmsreqs.pdf

BaBar - http://www.ihep.ac.cn/~chep01/paper/4-021.pdf

D0 - http://www.ihep.ac.cn/~chep01/paper/10-037.pdf

STAR - http://www.ihep.ac.cn/~chep01/abstract/10-017.htm

JLAB - http://www.ihep.ac.cn/~chep01/paper/4-016.pdf

2. Do you expect to have to deal with multi-language integration issues as a part of this project? If so, what are they? For those modules written for parallel execution, what parallel programming model(s) do you expect to use?

Yes.
C, C++, perl, java, sh, python.
HENP applications implement a coarse grained parallelism at the event or file level. Parallelism is achieved at the data file level through parallel job execution using batch systems such as LSF, Condor etc and event/file level parallelism using parallel analysis tools such as PIAF, PROOF.

3. For each new computational module, please identify major numerical algorithms you expect to need.

this is not applicable to PPDG.

4. What programming environment tools (editors, source code control, debuggers, profilers, etc.) would help you the most in the development or revision of these modules?

PPDG looks to the standard off the shelf public domain utilities for programming tools.

However, the support for large scale software development and distribution activities is lacking. Traditionally HENP experiments have developed their own software build, packaging and distribution tools tailored to their particular environment. While BaBar, CDF and D0 use the same build tool "Software Release Tools", CMS and Atlas use different and independent ones (SCRAM and CMT respectively). Several initiatives in the field to have a common toolkit have not succeeded.

Recently Globus has sponsored a project to package its software. It is a reflection on the state of the field that they developed their own set of perl and sh scripts to do this – and did not use any of the publicly available utilities.

We would like to think that the deliverables from the Software Carpentry project http://software-carpentry.codesourcery.com/ would be useful for the community and are concerned by the lack of information and progress reports on implementation of the winning entries.

5. On what computing platforms will you need the final system to run (computer systems, operating systems, compilers…) to meet your programmatic objectives? Be as specific as possible about the size of problems you need to run, execution time required, I/O needs, and mass storage use.

These are the responsibility of the experiments and not the PPDG project itself

6. If you will run your application across multiple computers and/or sites, what are your networking requirements in terms of bandwidth and latency to each site? What is your current external network connection speed and ISP/Network (Abilene, ESnet, MCI, AT&T…)? Will you have any special needs for wide-area network access (beyond current connectivity) to run your application? Estimate the peak bandwidth you will require. What areas drive your network requirements (e.g., file transfer, remote collaboration, remote visualization and/or computing…)? What network services (e.g., authentication…) are important?

HENP experiments have significant network resource requirements and is actively involved in several working groups both internal to the field and with other networking initiatives. Useful information is available from
http://lexus.physics.indiana.edu/griphyn/henp_issues.txt
http://www.usatlas.bnl.gov/computing/mgmt/lhccp/henpnet/
We are participating in the Internet2 HENP Working Group, as an appropriate framework for this group whose goals is to coordinate with the Internet2 Applications Strategy Council; and the Internet2 E2E Initiative.
The identified Transatlantic Network bandwidth requirements are in Mbits/sec (HN, L. Price)

	2001	*2002*	*2003*	*2004*	*2005*	*2006*
CMS	100	200	300	600	800	2500
ATLAS	50	100	300	600	800	2500
BaBar	300	600	1100	1600	2300	3000
D0	400	1600	2400	3200	6400	8000

7. What are principal locations in your team that require network services? Be specific with regard to location within a facility. (bldg number, point of contact, phone number….).

This is too numerous to list here. We refer you to each of the experiments participating in this project for this list.

8. What visualization requirements will you have for your application (if any)? Include visualization for each component (e.g., creation of input data sets, computational steering and monitoring of simulations, visualization and interpretation of output).

Visualization is the responsibility of the individual experiments working on the PPDG collaboratory pilot. While they could, we believe, benefit from common tools and interesting work is being done that is interesting to PPDG, no programmatic work of this nature is part of this project:

Part D: Identifying the most critical applied math and computer science needs for meeting your key objectives

The purpose of this section is to identify the critical applied math and computer science needs for achieving your project objectives. *In essence, where would you like to see us invest the most energy and resources to help make you successful?* There are seven major Integrated Software Infrastructure Centers (ISICs) funded under SciDAC. Below is a very brief description of each center. For each one, please do the following:

- Indicate possible interactions (if any) with this center that would help you meet your objectives.
- Relate these interactions back to your implementation plan and deliverables
- As best you can, give some indication of the priority of this particular interaction, in comparison to the others (either in this center, or other centers).

1. *Solvers ISIC:* This ISIC focuses on developing and implementing optimal or near-optimal schemes for PDE simulations and closely related tasks, including optimization of PDE-constrained systems, eigenanalysis, and adaptive time integration. The ISIC will research, develop and deploy and integrated toolkit of open source, (nearly) optimal complexity solvers for the nonlinear partial differential equations that arise in many Office of Science application areas, including fusion, accelerator design, global climate change, and reactive chemistry. These algorithms, primarily multilevel methods, aim to reduce computational bottlenecks by one to three orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible. Along with usability, robustness, and algorithmic efficiency, an important goal will be to attain the highest possible computational performance in its implementations by accommodating to the memory bandwidth limitations of hierarchical memory architectures.

The above ISIC is not directly related to the current work of PPDG.

2. *Locally Structured Grid Methods ISIC:* The goal of this ISIC is to develop a high-performance algorithmic and software framework for solving partial differential equations arising from three important mission areas in the DOE Office of Science: magnetic fusion, accelerator design, and combustion. This framework will provide investigators in these areas with a new set of simulation capabilities based on locally structured grid methods, including adaptive meshes for problems with multiple length

scales; embedded boundary and overset grid methods for complex geometries; efficient and accurate methods for particle and hybrid particle/mesh simulations; and high performance implementations on distributed-memory multiprocessors.

The above ISIC is not directly applicable to the current work of PPDG.

3. *Terascale Simulation Tools and Technologies ISIC:* The primary goal of this ISIC is to develop technologies that enable application scientists to easily use multiple mesh and discretization strategies within a single simulation on terascale computers. The focus will be on high-quality hybrid mesh generation for representing complex (and possibly evolving) domains, high-order discretization techniques for improved numerical solutions, and adaptive strategies for automatically optimizing the mesh to follow moving fronts or to capture important solution features. Results of this effort will be encapsulated into software components with well-defined interfaces that enable different mesh types, discretization strategies, and adaptive techniques to interoperate in a "plug and play" fashion.

The above ISIC is not directly related to the current work of PPDG

4. *Performance ISIC*: The Performance ISIC will focus on how one can best execute a specific application on a given platform. The research results from this effort are expected to permit the generation of realistic bounds on achievable performance, and to answer three fundamental questions: 1) why do these limits exist; 2) how can we accelerate applications toward these limits; and 3) how can this information drive the design of future applications and high-performance computing systems.

The work done under the Performance ISIC will be of value for the resource management and scheduling aspects of the PPDG focus. We would like to receive the reports of results and research done as part of this ISIC.

5. *Common Component Architecture ISIC*: This ISIC will involve research into software component technology for high-performance parallel scientific computing to address problems of complexity, reuse, and interoperability for scientific simulation software. Research under this ISIC will address the following areas. First, it will extend the Babel language interoperability technology to support Fortran 90 as needed by SciDAC application collaborators. Second, it will develop component schema and communication protocols between the ISIC component repository and component tools developed by collaborators. Third, it will continue to investigate parallel data redistribution approaches and integrate that support into the Babel language interoperability framework. Finally, ISIC team members will work with SciDAC application groups to use component technology in large, sophisticated simulation codes.

Ideas developed under the Common Component Architecture could have potential benefit to the distributed job management and execution components of PPDG. While the work is not directly related to the needs of the HENP experiments that are part of the PPDG Collaboratory Pilot, we will be interested in learning about the progress in the field and protocols and algorithms developed to achieve interoperability, data distribution techniques and performance.

.

6. *Scalable Systems Software ISIC*: This ISIC will focus on the development of systems software and tools for the effective management and utilization of terascale computational resources. The center will use a multidisciplinary approach to the development of an integrated suite of machine independent, scalable systems software components needed for the SciDAC Program. The goal is to provide open source solutions that work from small to large-scale systems.

Ideas developed under the Scalable Systems Software have potential benefit to PPDG. Scalable software is an integral part of the requirements of the experiment data handling systems PPDG is servicing – e.g. with respect to the ability to manipulate the billion HENP event/data objects of a typical PPDG experiment over the lifetime of the project. We will follow closely the progress in software and tools for the management and usage of terascale compute systems.

7. *Data Management ISIC*: The goal of this ISIC is to provide a coordinated framework for the unification, development, deployment and reuse of scientific data management software. The ISIC will target four main areas that are essential to scientific data management, emphasizing efficient management and data mining of very large, heterogeneous, distributed datasets. In addition, four tier levels of data have been identified: storage, file, dataset, and dataset federation. This "area and tier" framework is the basis for a management structure that will ensure the vertical integration of technologies in each area over these tier levels, as well as cross-utilization of area technologies. The result will be efficient, well-integrated, robust scientific data management software modules that will provide end-to-end solutions to multiple scientific applications

Members of PPDG are also leading members of the Data Management ISIC. The work of this ISIC is directly relevant to the needs of the PPDG experiments for access to their scientific data sets. PPDG SDSC members are collaborating with JLAB on issues directly related to the work of the Data Management ISIC. We plan to continue active communication with this project and collaborate with the participants in areas of mutual interest. Additionally the broader Experiment data handling projects may collaborate directly with this ISIC – e.g. the "data set query" research being done by CMS and EDG WP2.