As grid computing evolves into a practical, usable technology accessible to researchers across many fields and disciplines, the quality of support provided to grid users is of fundamental concern. A positive introductory experience to grid computing, a key part of which is reliable support services, is essential if researchers are to adopt the grid as a fundamental technology driving their work and the promise of grid computing is to be realized. At the same time, support providers require certain grid services, tools, and capabilities to efficiently and dependably serve these users. Such a support framework has not grown up at the pace of grid technology, however, and as users begin to assay production environments with real projects, the grid community has gained increased understanding of the enhancements needed.

My PACI REU internship afforded me the opportunity to engage head-on the nascent technology of grid computing and the resulting difficulties of user support. The document guiding my work was “Support Services and Tools Requirements,” a proposal submitted by Jim Giuliani to the Grid User Services Research Group of the Global Grid Forum. My work this summer involved implementing several of the technologies this document recommended, thereby demonstrating their potential usefulness when implemented successfully as well as exploring the changes in existing grid configurations necessary to make their implementation feasible. Through performing this work and presenting my tools to a working group at the Ninth Global Grid Forum, I gained a greater knowledge not only about grid computing but about the research process in computer science.

As a senior majoring in computer science and music, having past experience in music technology research and in software development, I was interested in this internship because, while my computer science knowledge supplied a good foundation for the work, the topic of grid computing was totally unfamiliar. My summer internship therefore began with an in-depth introduction to the grid and the Globus Toolkit software. I read through mountains of paper and online documentation, first those describing the grid in layman's terms, then eventually those designed for grid developers. I attended a two-day virtual workshop on Grid administration, which included a discussion of the future technology of Globus 3.0. I also gained hands-on experience using Globus and the grid through creating and submitting my own “jobs” to supercomputers and clusters at OSC. I arose from this experience with a solid understanding of the promise of grid computing as well as an appreciation of the needs my project would meet in the user support community.

After acclimating to Globus and the grid environment, I began to work toward accomplishing several of the goals outlined in “Support Services and Tools Requirements.” My first project involved creating a program to search for log file entries created within a given time window. It is inconvenient to manually scroll through logs tens of thousands of lines long, and grep and other search tools are not amenable to date range searches, especially for files with different formats. My program provided a faster and more convenient way to view a snapshot of the gatekeeper, job manager, and other
activity on a resource to track the source of a problem. This task proved to be quite challenging, because the program needed to be compatible with numerous log file formats, potentially including other formats not foreseen at the time of its design. With this in mind, I designed a script that searches through a log using a user-defined format template as its guide. The program also allows the user some control over the type of information included in the output. To make the script easy to use while maintaining this degree of flexibility, the program has several presets for what will probably be popular formats, such as for the Globus gatekeeper and PBS server logs.

The second programming project I took on was the creation of a search tool that could match Globus or PBS job IDs to detailed job information, and vice versa. Such a tool could be used to gather up the known history of a job, infer Globus or PBS IDs from each other, or provide the history of a user's activity. This project differed from the other in that the design constraints and requirements were more concrete. This software, while serving as a model for future tools on a variety of grid systems, needed only to consider the file formats, data logging procedures, and other peculiarities of Globus and PBS. The project was nonetheless challenging, because little formal documentation existed to codify these formats, procedures, and other behaviors. I had to rely on my own observation to determine what output would be reliably generated by these programs in a given situation, and therefore what data I could assume was available for my own software to use. This project was also significantly larger in scope, in the number of lines of code and number of distinct modules I chose to create. Once I had completed the code for a functioning information finding program, Jim Giuliani composed a python interface through which support personnel could easily use my software.

It was extremely rewarding to be able to present my work on these two projects to the User Services Working Group at the Ninth Global Grid Forum in October. Although there was small attendance at the working group session, the group members offered positive and constructive feedback on my work. I left the conference excited that, having had no experience with grid computing prior to June, my summer’s work had contributed software and knowledge of use to the grid community.

My sense of accomplishment also stems from broader skills and experiences I gained over the summer, which will be an asset in my future work even if I am not involved in grid computing specifically. I honed my abilities to search out obscure formal and informal documentation on software behavior and implementation, and in the process I was reminded of the importance of making my own software easy to understand and to modify by future developers. I was able to test my software until I was satisfied that it worked as it should. Because of this, I learned more about the testing process and about my own strengths and weaknesses as a programmer than I have in other projects, when I have not had the opportunity to test as rigorously. I also learned a great deal about the Perl language, and about the nuances involved in using it to implement large projects.

Furthermore, my summer position introduced me to an approach to computer science-related research to which I had not previously been exposed. Before this summer, most of my research in computer science centered around educational rather than technical issues. The technical research I had done was on small, independent projects spanning computer science and music, working alone and creating new music technology from scratch. This internship introduced me to the challenges and rewards of working on cutting edge technology within a large community of researchers. I saw firsthand the
difficulties that arise when there is not yet one established paradigm to cohesively guide work in a discipline. I also experienced the excitement central to potentially influencing future work in a new field. I expect to experience similar challenges and rewards in my future work in music technology, which is also a relatively recent field currently populated by enthusiastic researchers with divergent approaches to their work. The development, testing, and research skills I gained in my internship will undoubtedly benefit me as I pursue graduate studies next year in computer science-related music technology, and I sincerely appreciate NPACI allowing me the opportunity to gain these experiences.