HEP Software Foundation

GridPP input on purpose, scope, organisation etc.

Introduction

As has been mentioned by others, this initiative is very timely, both because of the decreasing levels of effort being funded in this area and the increasing diversity of hardware platforms that are becoming available. The last fifteen years of standardisation on the x86 architecture have been an anomaly which is coming to an end. This means that large volumes of code will need to be rewritten over the next few years with fewer people. It is therefore important both that the code they produce runs efficiently and the process by which they produce it is itself efficient.

This foundation should help to provide the environment and tools to make the production of efficient HEP software, often by geographically separate developers, as easy as possible. It should also foster interoperability between different HEP software activities. This interoperability will often come through the adoption of open standards.

This foundation would also provide a mechanism for interacting with other communities, however these interactions must be void of the arrogance that some members of our community have shown in the past and must be genuine collaborations.

The development of efficient HEP code is carried out by skilled developers. It is important that these people see a career path that is accessible to them. All too often these people do not see such a path and despite being interested in the challenges that HEP computing provides, take less challenging (if better remunerated) roles outside of HEP. The foundation should do what it can to help to raise the profile of HEP developers and so help to funders to understand the importance of providing career paths for them.

Goals

- To recognise the strengths of HEP software but to avoid the arrogance that some members of our community have shown in the past.
- To achieve interoperable pieces of HEP software. This should involve component and interoperability testing.
- To promote the use pre-existing open standards wherever possible.
- To promote the use of pre-existing pieces of existing open source software whenever appropriate. If it exists, even if it is from outside of HEP we don't need to write it ourselves.
- The code written by member projects of the foundation must be open source.
- The Foundation can suggest coding and documentation standards, but there adoption or otherwise is dependent on the individual projects.
- To provide a supportive environment for new projects. While a lot of code development is carefully planned in advance a lot of the best ideas come from somebody thinking “What would be really useful would be a tool that does ...”. To this end the barrier to entry should be low.
- Failure of a project to be adopted should not be viewed as a problem and should not carry stigma. If there are two projects doing similar things and one is adopted in preference to the other, or there are simply not enough interested parties to make a

tool worth maintaining then this should be viewed as OK and the project simply dropped with the developers moving on to new projects. Only writing bad code or failing to share should carry a stigma.

- The Foundation should provide a forum for people with ideas to be able to look for potentially interested collaborators.
- To build a community in which it is easy to write efficient code efficiently. To this end it could provide the tools to make this as easy as possible. The use of these tools would not be obligatory and would be there to aid projects not to constrain them. These might include:
    - Code repositories
    - Automatic build and test suite
    - Community building and communication tools (wikis, email lists etc)
    - Binary repositories for different Operating systems
    - etc.
- To also recognise the importance of users outside of the individual projects willing to stress test either individual components or combinations of components.
- To be a public face of HEP software where communities outside of HEP can look for potential software collaborators or products that they want to use. This would perhaps help people gain recognition for HEP software activities in the world outside.

Governance and membership

The governance of such a foundation should be as lightweight and transparent as possible. This foundation has no right to be able to tell people what they can develop or what they cannot develop. This is between the projects and their funders. The individual projects must retain the control over their work. It is inevitable that their be some governance but this should come from the individual development teams as they are probably best placed to understand how the foundation's activities best achieve the stated goals (perhaps also including those willing to stress test products).

Membership should be open to any HEP software project willing to share their code and collaborate with other HEP software projects.