Fahad's proposal

MAGGIE-NIIT

IEPM
"WHAT WE INTEND TO ACHIEVE"

By Fahad Ahmed Satti
IEPM (Internet End Host Performance Monitoring) project aims to develop and implement an end-to-end WAN Monitoring infrastructure by employing sophisticated measurement and analysis software.
IEPM provides an installation source that is used to configure monitoring hosts. These Monitoring hosts then apply different tests on selected monitored nodes. These tests and further analysis based on them, give us a fair idea of how the link to an end host is performing.

For the last six months or so I, have been working with SLAC (Stanfordlinear acceleratorCenter) to improve the presentation of IEPM data.

This task then led me to come up with certain improvements that if made on existing IEPM structure, can facilitate better visualizations. In this context, after having discussions with Dr. Les Cottrell, Mr. Umar Kalim, and my colleagues, I have come up with 5 improvements that may be made in the existing IEPM structure: # An automated IEPM installation package.

1. Visualization of IEPM data.
2. Increasing Levels of data Storage by introducing an Archive Server.
3. Providing Scripts to make Improve IEPM database after Installation.
4. Re-Implementation of IEPM in a higher level language (JAVA)

Automated IEPM Installation Package:

This task involves making an installation package which helps administrators and other users to install and configure IEPM with very little work to do. To remove dependencies and provide a source that provides all other software's necessary for IEPM installation.

This task can further be subdivided in these four tasks:

a. Provide an IEPM database creation Tool

b. IEPM support TOOL Automated Installer

c. IEPM User Control and Directory Structures Creator

d. A Control Script to perform all above tasks.

Archive Servers:

The current version of IEPM stores data on monitoring host about the nodes being monitored by that particular host. We intend to remove this restriction by inserting one more level to IEPM system. Scripts would be made which will regulate data between monitoring hosts and Archive servers. This data will act as a central repository and can then be used for further analysis and sharing. This task also has other benefits like after an archive server is built it can then be used to perform the level 3 and level 4 task (these tasks are related to analysis of data collected by monitoring hosts in order to determine its short and long term affects on internet system.). This will also reduce load on monitoring hosts. The methodology will be to write a server side code that is always running on archive server, and a client side code that is on the monitoring hosts. Now there are two different methodologies each of which has its own benefit. One method is to use a push model by which a client will push data once it is updated. The second methodology is the pull model by which the server will check for new data every few minutes or so. This data will be placed in a DB on the archive server. That DB will be accessible to perform more mature long term tasks. When I say long term and mature I mean tasks like ARMA/ARIMA and event detection. These are tests which give us a future insight to network performance. The DB will also be used to provide data for Visualization purposes.

*Visualization of IEPM data:*This task involves making a user friendly view of data that is stored on IEPM database on monitoring hosts. Providence of Archive server is meant to enable a central storage of data from where this data can then be used to visualize it in a better way. The concept I have in mind involves a map of earth appropriately a 3d map of earth on which different nodes are plotted then, a line with arrow to show the direction of tests being performed. Then on clicking a particular node graphs should open up. These graphs can be the ones already in use or data can be used to plot dynamic plots using JFREE CHARTS or some other JAVA API for plotting graphs. This also includes the task of changing the outlook of web pages, showing graphs and other data. *IEPM Database Improvement Tool:*The current IEPM database has a lot of redundancy and extra data that is either not used or is not causing any impacts on any analysis or test. We intend to provide a tool or scripts that can check database identify such data and remove it prefera

*Re-Implementation of IEPM in a higher level language (JAVA):*The current implementation is such that its extension is a non-trivial task. What I would like to perform as a major task would be to re implement iepm using a higher level language like java. This should make the code easier to understand and reproduce or change in the end. Debugging would be easier and iepm would become more scalable. Additionally, iepm would also become platform independent.

Comments from SLAC

Following on from reading the proposal synopsis from Fahad Ahmed Satti at the meeting, from the SLAC end it looks very interesting and we really do need a new iepm package, including automated installtion, archive servers and new front end. However the focus on Java is counter to our directions needs. The concerns are security at SLAC has a problem with Java, it is not really blessed. We are moving in the direction of perfSONAR and though the Europeans have done implementations in perl that is not the US direction. It is felt that Java implementations are hard to deploy and maintain in a production environment. Thus the US collaboration is requiring that most of perfSONAR is in perl. Less important is that SLAC has lots of experience in perl and not much in Java.

We also noted in the meeting that iepm is divided into a back end (data measurements, archiving) and a front end (visualization). Asif has been working very hard and made a lot of progress on implementing the iepm front end (still neeeds installing in production environment) and redesigning the back-end with a view to making it fit better into the perfSONAR framework. This back end (to be in perl) includes scheduling, archiving (later adding lookup service as it becomes available). The front end is in Javascript and HTML.

Thus it appears that a fruitful way to proceed is for Asif to work with Fahad to continue the design and development. The idea of an automated IEPM installation package is also very important. Given the direction of a new iepm it does not appear to be very fruitful to provide an installation package for it. If many sites were to install the old pacakge we would be run ragged trying to support them since it is not really production quality. Rather focus on the iepm backend, deploy and get it into production (harden it) at 2 sites (NIIT and SLAC), then build a installation package. So to be more specific Fahad we recommend that he should start to work with Asif to implement the design ideas for the iepm backend (in particular the Measurement point and Measurement archive). Following this the front end, hardening, deployment, installation automation etc.

One thing that might be useful for the current version of iepm (version 3) is to enhance it so that it can report back as a Measurement archive in response to perfSONAR requests.