Plush: An Infrastructure for Managing and Visualizing Distributed Applications

Jeannie Albrecht, Ryan Braud, Darren Dao, John Jersin,
Christopher Tuttle, Alex C. Snoeren, and Amin Vahdat
{jalbrecht, rbraud, ctuttle, snoeren, vahdat} @ cs.ucsd.edu
{hdao, jjersin} @ ucsd.edu

1 Research summary

Installing, configuring, deploying, and monitoring a distributed application is currently a time-consuming and error-prone process. In any environment made up of hundreds or even thousands of physical machines distributed across the Internet, failures are inevitable . Hence, applications running across these platforms must be carefully monitored and controlled to ensure continued operation and sustained performance. Operators in charge of deploying and managing these applications face a daunting list of challenges: discovering and acquiring appropriate resources, distributing files, and appropriately configuring hosts (and reconfiguring them when operating conditions change). It is not surprising, then, that a number of tools have been developed to address various aspects of this process. Unfortunately, no solution has been presented that flexibly automates the entire task while gracefully handling the variety of failures that often present themselves in large-scale, wide-area deployments.

We present Plush, a framework of tools that, when taken together, provide a unified environment supporting the distributed application design and deployment life cycle. Plush users describe distributed applications using an extensible XML specification language that allows users to customize various aspects of the deployment life cycle to fit the needs of an application and its target infrastructure. This functionality is used, for example, to specify a particular resource discovery or allocation tool to use during application deployment. In addition, Plush provides extensive failure management support to adapt to failures in both the application and the underlying computational infrastructure.

Plush manages resource discovery and acquisition, software distribution, and process execution in a fully configurable fashion. Once an application is running, Plush monitors it for failures or application-level errors for the duration of its execution. Upon detecting a problem, Plush can perform a number of user-configurable recovery actions, such as restarting the application, automatically reconfiguring it, or even searching for alternate resources. For applications requiring wide-area synchronization, Plush provides several efficient synchronization primitives. In particular, Plush provides two new barrier semantics, which relax traditional barrier semantics for increased performance and robustness in failure-prone environments.

2 Poster and demo details

Our poster highlights the main components of Plush and focuses on some of the new features that have been recently added for increased usability. For example, Plush now exposes its API through an XML-RPC interface, allowing programmatic interaction in addition to the terminal interface. We show a screenshot of Nebula, a visualization application that uses this API to present the status of a PlanetLab application in a graphical manner. All stages of a distributed application's execution are visualized and manipulated with Nebula, making it easier for users to debug their programs and deal with failures quickly. Plush also now supports integration with virtual machine management infrastructures, allowing users to create and destroy virtual machines through Plush as part of their application's execution.

In conjunction with the poster, we present a demo of Plush in action. The demo focuses on how Plush is used to manage PlanetLab applications. In particular, we show Plush being used with Nebula. Nebula includes a tool that allows users to create their application specifications for Plush graphically, rather than having to manually edit XML files. After building the application specification, we show how to deploy our application on some subset of PlanetLab resources. We then show how we use Nebula to visually monitor the status of our application as it runs.

For more information and details regarding Plush, please visit `http://plush.ucsd.edu`.