Preliminary Panda Architecture

Peter Reiher

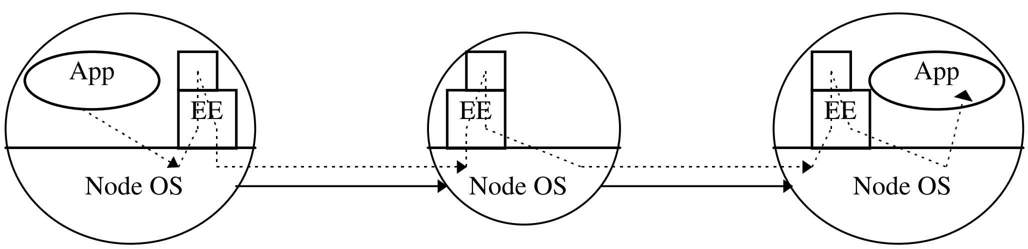
August 12, 1998

1. Introduction

The Panda system will provide middleware support for applications using active networks. The main goal of the system is to provide such support for applications that are unaware of the existence of the active network and are thus unable to make use of such a network's capabilities directly. A secondary goal is to provide useful services for applications that are able to interact with active networks. Further, the project operates under some constraints applied by the overall active networks program. In particular, the active network architecture they are developing sets certain limits on what is possible in Panda.

This document describes a preliminary architecture for the Panda system. This preliminary architecture is meant purely to serve as a starting point for more serious design of the Panda system. Other than the goals and constraints described above, all other aspects of Panda described here are little more than placeholders, and are subject to change when we discover better ways to do things.

The diagram below shows a simple version of the current active network architecture. One of the most fundamental questions that the Panda project must decide early on is how Panda software fits into this architecture. The diagram shows three active network nodes participating in support for a communication. One node creates and sends the data, one node is purely a middle point that uses active network technology to get the data to the destination effectively, and the third node receives the data and delivers it to an application for consumption.



Speaking roughly, a message created by the sending application goes to the Node OS, which determines which of several execution environments should handle the message. The execution environment in turn may choose to select an agent or other piece of code to handle the message. Once the agent completes the handling, the execution environment calls the Node OS to request actual physical transmission along some network link. The process repeats in the middle node, though generally no portion of the user application itself will be resident there. At the destination end, after the EE and agent have done their work, instead of requesting further transmission, the EE requests delivery to the destination object.

Some of these steps may be short circuited or optimized in various ways in some active network implementations.

2. Panda Design Assumptions

The following assumptions guide this preliminary design:

A. Panda must be able to capture messages sent by applications unaware of Panda's existence.

B. Panda must interoperate with non-Panda active network uses and with non-Panda, non-active network uses.

C. Panda should be as independent of particular execution environments and node OSes as possible.

D. Panda wants to make use of as many already available execution environment and node OS services as possible.

E. Panda must be able to deploy its own adaptation agents at any active network node involved in a stream of data Panda is trying to support. However, Panda must not fail when some active network node (or non-active network node) is unable or unwilling to host Panda software. Instead, the system should degrade gracefully.

F. If possible, Panda itself should inject no code into the stream of application messages being sent over the active network. Panda-selected adapters may inject code into that stream, but the underlying Panda framework should avoid introducing overheads into normal operations.

G. Panda should be able to make use of adaptation agents written by others, with no knowledge of Panda.

3. Preliminary Panda Design

Panda will consist of three components. The Panda Interception Component (PIC) will trap messages coming out of applications. The Panda Adaptation Component (PAC) will handle interactions with active network technologies. The Panda Updating Component (PUC) will monitor vital quality of service parameters that might

The Panda Interception Component (PIC) will capture all messages sent by applications very shortly after those messages leave the application's user space and are made the responsibility of the operating system. For the first message of a new stream of data, Panda will attempt to identify the type of the data. If Panda cannot identify it, Panda will step out of the picture and make no attempts to assist this message or further messages in this data stream. If Panda can identify the stream type and has the ability to plan adaptations for that type properly, it will convert the message to an active network message and keep track of the fact that it is adapting this stream of data. All subsequent messages sent in the same stream will be captured and converted into proper active network messages.

The active network header added to the first message will instruct the local execution environment on the sending application's node that this message should be delivered to the other Panda component, the Panda Adaptation Component (PAC). This component is associated with a particular execution environment on the active network node. The Interception Component will re-route the altered message through the node OS to cause the message to be delivered to the proper execution environment.

The execution environment will recognize that this message should be passed to the Panda Adaptation Component. From the point of view of the EE, the PAC is simply an agent. The EE treats it like any other agent, giving it the same privileges and services available to other agents. The PAC requires the ability to request this EE to deploy new agents locally, and also requires the ability to request the local EE to send active network messages to other active network nodes.

When the PAC receives the message, it invokes Panda planning code suitable for the type of stream in question. Typically, this code will deploy one or more local agents that will take responsibility for applying any necessary local adaptations for this stream. The PAC will also decide where to route the message next, and will create a new active network message to be sent to the next active node. This message will contain the actual data for the first message in the transmission, with an active network header that will cause the active node receiving the message to deliver it to the EE that hosts the PAC on that machine. The message will also contain information about which actual adaptation agents have been deployed at which active nodes in support of the stream. Further, it will contain information about the important network and node characteristics that might affect the kinds of adaptations that Panda might want to perform on the stream.

At each node on the path from source to destination, this initial message for the stream will get delivered to the node's PAC, which will use the accumulated information plus local information to deploy local agents and determine if the already deployed agents at other nodes fit into a sensible adaptation plan for this stream. Assuming that all is well, the process will continue until the message reaches its destination. At this step, in addition to deploying agents required to deal with earlier adaptations that have been applied, the PAC will ensure that the message is routed through a Panda adapter that removes any traces of its active network origins that might not be manageable by the destination application.

If a PAC determines that some adapters already deployed are not suitable for the global conditions, it will send active network messages to the PAC handling the unsuitable agents to instruct those PACs to shut the existing agents down and replace them with more suitable ones. Care will be required to handle adapted data already in transit.

Each node running Panda will run the PUC. The PUC will be a separate application-level process that monitors quality of service issues relevant to Panda. When these change significantly, the PUC will send the local PAC an active message informing it of the changes. The PAC will keep information on all active streams, and will decide if the new message from the PUC requires any changes in plans for any of those streams. If so, it instructs the local EE to make the local changes and notifies other nodes' PACs of alterations to their agent deployment.

The PIC will be resident within the Node OS. The PAC will be an EE environment agent that requires persistence. The PUC will be a user level (but possibly privileged) application. Typical messages in the middle of a stream will be intercepted by the PIC, but probably will not be delivered to any PAC. Instead, they will visit the actual adaptation agents deployed on their behalf by PACs. No user stream message will ever interact with the PUC.