Supporting Broad Internet Access to TACOMA*

Dag Johansen[1]
University of Tromsø
N-9037 Tromsø
Norway

Robbert van Renesse[2]
Cornell University
Ithaca, New York
U.S.A.

Fred B. Schneider[3]
Cornell University
Ithaca, New York
U.S.A.

1. Introduction

The TACOMA system [JRS95] provides operating-system support for mobile processes, or *agents*, that traverse the hosts of a network in accomplishing some task. A TACOMA agent executing on one host moves to another host by using TCP to communicate with TACOMA software at the destination host. The presumption is that TACOMA software has been installed at any host that might launch or be visited by an agent. This presumption is questionable:

- System managers are reluctant to install software that allows arbitrary imported software, like agents, to be executed on their hosts. And, there is good reason for this reluctance.
- Hosts are autonomous. So, even were security not a concern, arranging for installation of a piece of non-critical software on a significant fraction of the hosts in a large network is impractical.

The network-software installation-problem is not unique to agents or to TACOMA. Any provider of software will be faced with this problem if that software must be installed on autonomous sites of a large network. The speed with which WWW software spread through the Internet was an exception; architects of distributed-application support-software should not plan on having their systems disseminated in such a fashion. But, these architects better have some plan for disseminating their systems, since the utility and success of large networks will depend on offering new services in a timely way.

This paper reports experiences in addressing the network-software installation-problem for TACOMA. However, we believe that the techniques employed have utility in other situations as well. The next section gives an overview of TACOMA and the applications it currently supports. Section 3 describes a WWW-based scheme for avoiding software installation at all sites that might launch TACOMA agents; section 4 discusses an email-based scheme. Section 5 concludes with a summary and some observations.

2. The TACOMA System

Over the last three years, we have built a series of TACOMA (Tromsø And COrnell Moving Agents) system prototypes. TACOMA version 1.2, which is in the public domain (see http://www.cs.uit.no/DOS/Tacoma/), is in daily use as a production platform and runs under HP-UX, Solaris, BSD Unix, and Linux. Agents may be written in Tcl/Tk, C, Scheme, Perl, or Python; support for Java is now being added. Distributed applications have already been implemented using TACOMA to gather and visualize Arctic weather data [JH95], to provide matching between service providers and potential clients, to communicate and interact with users (i.e. active documents), and to manage software installation in a network.

TACOMA agents store data in *folders*. A subset of the folders are identified with individual hosts and collected in the *file cabinets* managed by those hosts; the remaining folders comprise a *briefcase* that is moved from host to host along with the computation.

---

*Contact author: Dag Johansen, Department of Computer Science, University of Tromsø, Tromsø, Norway. Email: dag@cs.uit.no.

[1]Supported by NSF (Norway) Grant No. 17543/410 and 111034/410.

[2]Supported in part by ARPA/ONR Contract N00014-92-J-1866.

[3]Supported in part by ARPA/NSF Grant No. CCR-9014363, NASA/ARPA grant NAG-2-893, and AFOSR grant F49620-94-1-0198.

A TACOMA agent can cause another agent to be executed by invoking the **meet** operation and naming a *target agent* and a briefcase. The effect of the operation is to terminate the agent invoking the **meet** and then start executing the target agent with the specified briefcase. Thus, transfer of control in TACOMA from one agent to another is similar to the transfer of control enabled by using continuations in Lisp-like languages. We have found a variation of **meet**, which supports remote-procedure call semantics, is a convenient abbreviation for structuring some applications.

Surprisingly, no additional abstractions are required to implement our basic computational metaphor. Services for agents—communication, synchronization, and so on—are provided directly by other agents. For example, an agent moves from one host to another by **meet**ing with the *tac_firewall* agent on the destination machine. Once the **meet** is called, it opens a TCP connection with the *tac_firewall* agent at the destination site found in the HOST folder of the briefcase. That *tac_firewall* agent logs the incoming briefcase to disk, performs some security checks, and then **meet**s with whatever agent is named in the CONTACT folder. For example, if the specified agent is a compiler or interpreter, then the code found in the appropriate CODE folder is extracted and executed.

Related Work

Java [CW96] is probably the best known system for launching computations into the Internet. Java programs, called applets, can be downloaded from web servers and executed by Java-able web browsers. Typically, an applet will enhance the graphical user-interface of a web browser for use with some specific application.

TACOMA agents and Java applets are complementary. Rather than being downloaded into a browser like applets, TACOMA agents are sent out into the internet to do work on behalf of the user. Java applets can launch TACOMA agents (using a CGI [CGI96] front-end to TACOMA), and TACOMA agents can interact with a user through applets.[1] Because any results of an agent's execution can be returned to a user as an applet, there is no reason to support running TACOMA agents in browsers—applets do that job nicely. In fact, TACOMA agents may be written in the Java programming language if so desired.

Other related work includes the World Wide Web Consortium's new web server, Jigsaw [J96], which will allow applets to be run on servers, and recent work by General Magic on their Telescript system [T96].

3. TACOMA Through WWW

One way to avoid installing software at a site is to use software that has already been installed there. The ubiquity of web browsers makes them an obvious target of opportunity. We, therefore, have developed a scheme so that agents can be launched from a site on which a web browser, but no TACOMA software, has been installed. The scheme basically works as follows.

(1) A user at some host constructs an agent.

(2) Then, using a web browser, the user transmits that agent to a TACOMA server located elsewhere on the network. The *tac_web* agent, a CGI-bin script at the server, **meet**s with the local *tac_firewall* agent, which launches the user's agent.

(3) The *tac_web* agent (ultimately) returns to the originating web browser an HTML page containing results of the agent's execution.

No longer are all hosts equivalent. Some hosts have TACOMA software installed and execute the *tac_firewall* agent; some do not. However, by virtue of WWW, those that do not execute the *tac_firewall* agent are capable of being clients to hosts that can launch agents. A web browser and HTTP, software that is likely to be installed at all hosts, is used to support the additional client-server interaction.

The scheme we are employing is an instance of a general method for implementing a distributed application comprising a set of peers:

---

[1]Try this out for yourself. Point your web browser to http://www.cs.uit.no/DOS/Tacoma/ and follow the demonstration links from there. It will allow you to download forms and applets that can launch agents at remote TACOMA servers. If you install a TACOMA server locally then you will be able to launch a TACOMA agent there that can travel to other TACOMA sites on its own.

Relax the requirement that the system comprise a collection of peers, and support peers that function as gateways for arbitrary clients. Use extant support to implement the interaction between the client and gateway, thereby avoiding the need to install software at hosts executing clients.

Notice that this architecture of gateways and servers accommodates hosts with limited capabilities especially well. The architecture is thus well suited for the use with "network appliances" that promise to provide Internet access for the masses.

Our use of a web browser in order to submit TACOMA agents for launching also provides support for users who are unwilling or unable to build agents for themselves. Such a user would employ the web browser and select an agent that has been advertised as accomplishing the task at hand. The pool of available agents might be maintained by the TACOMA gateway itself. Or, the gateway might simply certify and sign those agents that it is willing to launch, allowing other hosts to provide these predefined task-specific agents. Certification and signing enables TACOMA to prevent the launching of malicious agents. This is necessary because TACOMA agents are not restricted in what sites they can visit or what actions they can take, the way Java programs are. For TACOMA, other methods (currently under investigation) are employed to guarantee host integrity in the presence of malicious agents.

4. TACOMA by Mail

Email provides an extremely simple facility for a user at some host to allow visits to that host by TACOMA agents. With this facility, a visiting agent is permitted to do anything at the host that the user can do. This limits agents to interacting with and using only those resources that are accessible to the user and, thus, limits the damage that a hostile agent can inflict.

The scheme we employ is quite simple:

(1) The user defines a .forward file that causes a copy of every incoming piece of email to be processed by a filter.[2] The filter forwards any email marked as being a TACOMA agent to a local *tac_firewall* agent.

(2) The **meet** command is modified to communicate using email rather than TCP.

Notice that this scheme is one that can be installed by an arbitrary user at a host.

Email also allows agents to be submitted to a TACOMA server for launching, in much the same way as described above for web browsers. The author of an agent employs a regular mailer to send email containing that agent to a user id having a suitable .forward file.

Email is attractive for agent communication and movement because email that cannot be sent immediately is spooled. This spooling is useful for environments where connectivity between sites is intermittent or network partitions are frequent. Agents are well suited to such environments, because an agent sent to a host can proceed asynchronously and out of contact with the host launching that agent. And, such environments will become increasingly important as forms of mobile computing become more prevalent.

5. Final Remarks

To many, system management is something dealt with in isolation by staff supporting each individual host. In a world of networked computers, this view is inadequate, because providing network-wide services will require some means to install network-wide software. Such an installation might involve a significant number of hosts, each controlled by a separate organization. Thus, support for system management is about to become a critical issue in the evolution of networks. It is an issue ripe for discussion.

For TACOMA, we have been able to avoid some software-installation problems simply by adapting WWW facilities. But adding gateways is not the same as adding new servers, so our WWW solution is not entirely satisfactory in that regard. Our other solution, subverting email to carry agents, actually involves a user installing software—at best, a partial solution.

---

[2]This "user" may be a fictitious user-id created specifically for processing TACOMA agents.

References

[CGI96] WWW URL http://hoohoo.ncsa.uiuc.edu/cgi/.

[CW96] Campione, M. and K. Walrath. *The Java Tutorial: Object-Oriented Programming for the Internet*. Addison-Wesley, 1996.

[J96] WWW URL http://www.w3.org/

[JH95] Johansen, D. and G. Hartvigsen. Architectural issues in the StormCast system. *Theory and Practice in Distributed Systems*. Lecture Notes in Computer Science Vol. 938, Springer-Verlag, Heidelberg, 1995, 1-16.

[JRS95] Johansen, D., R. van Renesse, and F.B. Schneider. Operating system support for mobile agents. *Proc. Fifth Workshop on Hot Topics in Operating Systems (HOTOS-V)*. IEEE Press (May 1995), 42-45.

[T96] WWW URL http://www.genmagic.com/Telescript/index.html