Addendum to the paper "Isolation with Flexibility: A Resource Management Framework for Central Servers,"

published in: *Proceedings of the 2000 USENIX Annual Technical Conference*
(San Diego, California), USENIX Association, 2000, pp. 337-350.

David G. Sullivan, Margo I. Seltzer
Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138
`{sullivan,margo}@eecs.harvard.edu`

Alternate Solution to the Lower-Limit Problem

In Section 3.3 of the paper, we describe how currencies—like all mechanisms for providing isolation—impose lower limits on resource allocations. These limits prevent lottery scheduling from fully supporting the semantics of UNIX's `nice` utility. To solve this problem, we developed a utility that allows users to transfer some of their user currency's funding to another currency (the "nice" currency) that is then used to fund a resource-intensive task (see Section 4.6 and Figure 4 of the paper).

One problem with this original solution is that the transferred resource rights are never restored. When all resource principals funded by the "nice" currency exit, the currency, and all of its backing tickets, are destroyed. However, because the kernel cannot know that these backing tickets originally belonged to the user currency, the user currency's funding remains at a reduced level. As a result, an additional utility would be needed to allow users to regain their currency's original funding.

To overcome this problem, we developed an alternate utility that succeeds in performing a *temporary* transfer of resource rights. It uses a ticket exchange (Section 3.2 of the paper) to take away a small percentage of the user currency's resource rights and give them to the "nice" currency. As Figure A1 shows, the fact that we use a ticket exchange means that the user currency's original backing tickets are not modified. Instead, the user currency receives *negative* tickets that reduce its resource rights by the amount being transferred (see Section 3.2.2 of the paper). When all principals funded by the "nice" currency exit, the currency is destroyed as before. But because its backing tickets are part of an exchange, the exchange is undone and the negative tickets held by the user currency are also destroyed. The user currency thereby regains the funding that it held before the transfer of resource rights took place.

Our source distribution (`ftp://ftp.eecs.harvard.edu/pub/vino/vino-usenix2000`) includes both versions of the utility. In the directory `src/utils/vino/usr.bin/vnice`, `vnice.c` contains the source for the new solution, and `vnice.old` contains the source for the original solution.

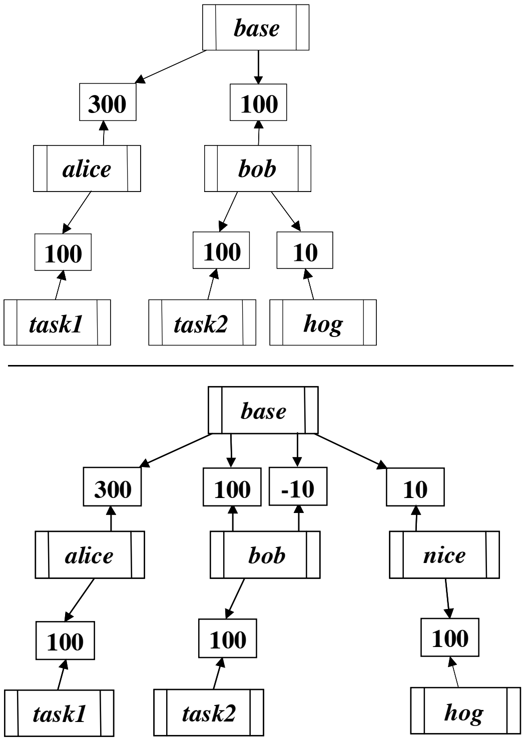

Figure A1: Using a Ticket Exchange to "Nice" a Task. The user *bob* tries to lower the priority of *hog*, a CPU-intensive process, by giving it only 10 tickets (*top*). However, if *task2* becomes idle or exits, *hog* will still receive all of *bob*'s resource rights. The *vnice* utility uses a ticket exchange to temporarily transfer a small percentage of *bob*'s resource rights to a new currency, *nice*, and the new currency is used to fund *hog* (*bottom*). When the *nice* currency is destroyed, the tickets involved in the exchange will be revoked, and the *bob* currency will regain its original funding.