

The Solaris[™] Network Cache and Accelerator

A Technical White Paper



Table of Contents

Introduction1
Solaris NCA Overview	1
Solaris NCA Chronology	2
Why Is Solaris NCA Important?	2
Using the Solaris Network Cache and Accelerator3
Solaris NCA Architecture	3
How Does Solaris NCA Work?	3
When to use the Solaris NCA	4
Web Servers Supported	4
Planning and Implementation5
System Requirements for Solaris NCA	5
Installation and Configuration 5	
Libraries	5
How to Configure Web Servers to Use Solaris NCA	5
How to Disable Caching of Web Pages 6	
How to Enable or Disable Solaris NCA Logging	7
Tuning Solaris NCA for Maximum Performance	7
General System Tuning	9
Performance Data10
Conclusion11
References12

Chapter 1

Introduction

The growth in network traffic and high-bandwidth connectivity has placed increased demands on today's Internet data centers to push more HTML content out to the Internet. This acceleration has particularly impacted Web serving.

To meet these new requirements, operating systems and Web servers have incorporated significant improvements in scalability and performance in recent releases. As a result, for the same computing capacity, today's Web server systems can sustain much higher user loads.

Solaris™ Network Cache and Accelerator (NCA) technology enables the increased server capacity required by today's Web centers.

Solaris NCA Overview

The Solaris NCA software helps increase Web server performance by maintaining an in-kernel cache of Web pages that have been recently accessed by HTTP requests.

When a user requests an HTML page contained in the Solaris NCA cache, the page is served directly from the Solaris NCA cache with no Web server involvement. This results in large performance improvements, since a Web page served from the cache requires less instructions to be processed.

Solaris NCA also incorporates other design changes that improve performance when a page needs to be processed by the Web server.

Solaris NCA Chronology

- Solaris NCA was first introduced in the Solaris 8 Operating Environment (OE) with a DOORs-based interface to the Web server.
- In Solaris 8 OE, Update 7/01, a sockets interface to Solaris NCA was introduced. This new interface allows Solaris NCA to run with the most popular Web servers. The Solaris 8 Update 7/01 release also introduced the `sendfilev()` system call, which allowed an application to send multiple vectors or data and file descriptor to Solaris NCA.
- Further refinements, including multiple-server instance support, are planned for a future release of the Solaris 9 OE.

Why Is Solaris NCA Important?

Solaris NCA provides:

- Increased server capacity. A Web server using Solaris NCA can process many more user requests per second than the same Web server that is not using Solaris NCA.
- Better return on investment (ROI): The increased server capacity translates to a lower cost per transaction, and reduces the number of systems required to do the same amount of work.

Chapter 2

Using the Solaris Network Cache and Accelerator

Solaris NCA is fully implemented in the Solaris OE. It does not require additional hardware.

Solaris NCA Architecture

Solaris NCA consists of the following components:

- Solaris NCA technology-enabled Web server: httpd
- Kernel module: ncakmod
 - The kernel module ncakmod maintains the cache of Web pages in system memory. It also communicates with a Web server, httpd, through a sockets interface (family type AF_NCA).
 - The kernel module also provides a logging-facility, which logs all HTTP cache hits. Solaris NCA logging writes HTTP data to the disk in binary format and provides the utility for converting binary log files to common log format (CLF).

How Does Solaris NCA Work?

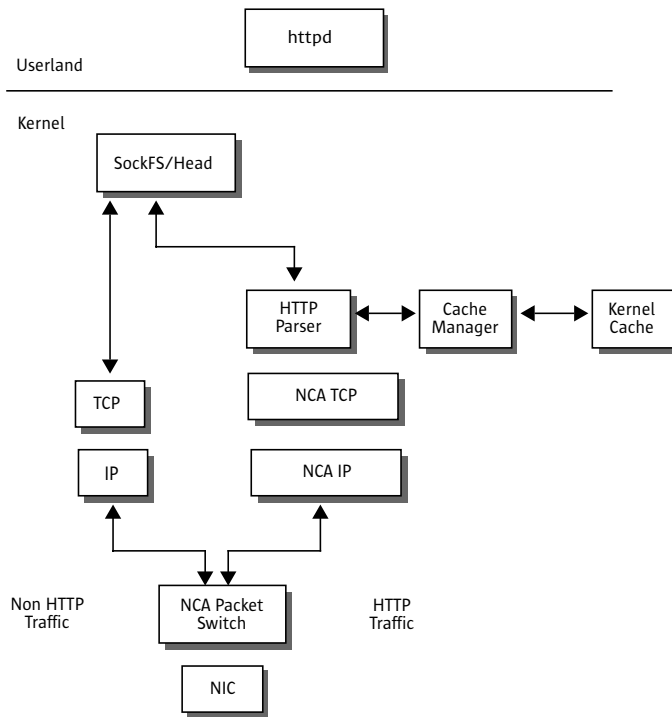
In the absence of Solaris NCA, the data flows from the Network Interface Card (NIC) to the Web server through the IP and TCP layers.

As shown in Figure 2-1:, once Solaris NCA is enabled, the request flow between the client and the Web server happens as follows:

1. An HTTP request is made from the client to the Web server.
2. Packet from the client hits the Solaris NCA Packet Switch.

3. The Solaris NCA Packet Switch looks at the port number, and decides if the incoming packet is meant for NCA(dport=80) or IP.
4. Solaris NCA IP (which is optimized for Solaris NCA), processes the packets.
5. Solaris NCA TCP (which is optimized for Solaris NCA), looks up the connection or creates a new one, and passes the packet to the HTTP Parser.
6. The HTTP engine parses the packet, while the Cache Manager checks with the Kernel Cache to find out if it is a static request and if the page is cached.
7. If the page is in cache, the in-kernel cache Web page is returned.
8. If the page is not in cache, the request goes to the Web server to retrieve or update the page.
9. Depending on the HTTP protocol semantics used in the HTTP response, the page is cached or not, and then returned to the client. If the Pragma: No-cache header is included in the HTTP request, the page is not cached.

Figure 2-1: The conventional flow of data when Solaris NCA is enabled



When to use the Solaris NCA

The Solaris NCA should run on a dedicated Web server. Simultaneously running large CPU- and memory-intensive processes on the same server that is running the Solaris NCA may cause problems.

Web Servers Supported

The Solaris NCA uses a sockets interface through which Web servers can communicate with minimal modifications. It works with the most popular socket-based Web servers, such as the Sun™ ONE Web Server, Apache, AOL, and Zeus, without protracted modifications. These modifications include only changes in the configuration files, not source code changes.

Chapter 3

Planning and Implementation

This section describes the process of planning for and implementing Solaris NCA in the Solaris OE.

System Requirements for Solaris NCA

To support Solaris NCA, the system must meet these requirements:

- 256-Mbytes RAM
- Solaris 8 OE, update 7/01 or later

Installation and Configuration

Libraries

If a Web server does not use a native Solaris NCA sockets interface (that is, a socket of type `AF_NCA/PF_NCA,ncad_addr.so`), it can be preloaded. Add a line such as `LD_PRELOAD=/usr/lib/ncad_addr.so/usr/sbin/httpd` in the Web server startup script.

How to Configure Web Servers to Use Solaris NCA

1) Register the interfaces

Enter the names of each of the physical interfaces in the `/etc/nca/nca.if` file (See the `nca.if(4)` man page for more information).

```
# cat /etc/nca/nca.if
hme0
hme1
```

For each interface, there must be an accompanying `hostname.interface-name` file and an entry in `/etc/hosts` file for the contents of `hostname.interface-name`. To bring up the Solaris NCA feature on all interfaces, place an asterisk (*) in the `nca.if` file.

2) Enable the `ncakmod` kernel module

Change the status entry in `/etc/nca/ncakmod.conf` to enabled.

```
# cat /etc/nca/ncakmod.conf
#
# NCA Kernel Module Configuration File
#
status=enabled
httpd_door_path=/var/run/nca_httpd_1.door
ncad_status=disabled
nca_active=disabled
See the ncakmod.conf(4) man page for more information.
```

3) Enable NCA logging

Change the status entry in `/etc/nca/ncalogd.conf` to enabled.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=enabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
The location of the log file can be changed by changing the path indicted by the
logd_path_name entry. See the ncalogd.conf(4) man page for more information.
```

How to Disable Caching of Web Pages

1) Disable the `ncakmod` kernel module

Change the status entry in `/etc/nca/ncakmod.conf` to disabled.

```
# cat /etc/nca/ncakmod.conf
# NCA Kernel Module Configuration File
#
status=disabled
httpd_door_path=/var/run/nca_httpd_1.door
ncad_status=disabled
nca_active=disabled
See the ncakmod.conf(4) man page for more information.
```

2) Disable NCA logging

Change the status entry in `/etc/nca/ncalogd.conf` to disabled.

```
# cat /etc/nca/ncalogd.conf
#
# NCA Logging Configuration File
#
status=disabled
logd_path_name="/var/nca/log"
logd_file_size=1000000
See the ncalogd.conf(4) man page for more information.
```

3) Reboot the server

How to Enable or Disable Solaris NCA Logging

Solaris NCA logging can be turned on or off as needed once the program is enabled. (See “How to Enable Caching of Web Pages” for more information.)

1) Change Solaris NCA logging

To permanently disable logging, the status in `/etc/nca/ncalogd.conf` to disabled must be changed, and the system rebooted. See the `ncalogd.conf(4)` man page for more information.

To stop logging

Type the following command:

```
# /etc/init.d/ncalogd stop
```

To start logging

Type the following command:

```
# /etc/init.d/ncalogd start
```

2) How to Load the Solaris NCA Socket Utility Library

Follow this process only if the Web server does not provide native support of the `AF_NCA` socket.

Add a line to the Web server startup script that causes the library to be preloaded. The line should look something like this:

```
LD_PRELOAD=/usr/lib/ncad_addr.so /usr/bin/httpd
```

Tuning Solaris NCA for Maximum Performance

Tuning Solaris NCA for maximum performance is recommended only if the Web server software runs on a dedicated Web server machine. This tuning allocates more memory for caching the pages. The variables should be set in `/etc/system`.

nca:nca_conn_hash_size

Controls the hash table size in the Solaris NCA module for all TCP connections.

Adjusted to nearest prime number.

```
minimum: 0
maximum: 201326557
default: 383
```

When to Change:

This should be changed when the Solaris NCA TCP hash table is too small to keep track of the incoming TCP connections, many TCP connections are grouped together in the same hash table entry. This situation is indicated when Solaris NCA is receiving many TCP connections and performance is decreasing.

nca:nca_conn_req_max_q

The maximum number of pending TCP connections for Solaris NCA to listen on.

```
minimum: 0
maximum: 4,294,967,295
default: 256
```

When to Change:

This should be changed when Solaris NCA closes a connection immediately after it is established because it already has too many established TCP connections. If Solaris NCA is receiving many TCP connections and can handle a larger load, but is refusing additional connections, increase this parameter to allow it to handle more simultaneous TCP connections.

nca:nca_conn_req_max_q0

The maximum number of incomplete (three-way handshake not yet finished) pending TCP connections for Solaris NCA to listen on.

```
minimum: 0
maximum: 4,294,967,295
default: 1024
```

When to Change:

This should be changed when Solaris NCA refuses to accept more TCP connections because it already has too many pending TCP connections. If Solaris NCA is receiving many TCP connections and can handle a larger load, but is refusing additional connections, increase this parameter to allow Solaris NCA to handle more simultaneous TCP connections.

nca:nca_ppmax

Maximum amount of physical memory (in pages) used by Solaris NCA for caching the pages.

Should not be more than 75 percent of total memory.

```
minimum: 1% of physical memory
maximum: limited by amount of physical memory
default: 25% of physical memory
```

When to Change:

This should be changed when using Solaris NCA on a system with a great deal of memory (more than 512 MB). If a system has physical memory that is not being used, increase this parameter so that Solaris NCA will efficiently use memory to cache new objects and increase the system's performance. *nca_ppmax* should be increased in conjunction with *nca_vpmax*, unless the system has more physical memory than virtual memory (a 32-bit kernel having greater than four gigabytes of memory). Use `pagesize(1)` to get the page size on the system.

nca:nca_vpmax

Maximum amount of virtual memory (in pages) used by Solaris NCA for caching pages. Should not be more than 75 percent of the total memory.

```
minimum: 1% of virtual memory
maximum: limited by amount of virtual memory
default: 25% of virtual memory
```

When to Change:

This should be changed when using Solaris NCA on a system with a great deal of memory (more than 512 MB). If a kernel has virtual memory that is not being used, increase this parameter so that Solaris NCA will efficiently use memory to cache new objects and increase the system's performance. This should be increased in conjunction with *nca_ppmax* and should be approximately the same as *nca_vpmax*, unless the system has more physical memory than virtual memory.

General System Tuning

Apart from Solaris NCA tuning, general system tuning can also improve performance. When using Sun Gigabit Ethernet cards, set it in interrupt mode for better results.

sq_max_size

The depth of the syncq (number of messages) before a destination streams queue generates a QFULL.

```
minimum: 1
maximum: 0 (unlimited)
default: 2
```

When to Change:

When Solaris NCA is running on a system with a great deal of memory, increase this parameter to allow drivers to queue more packets of data. If a server is under heavy load, increase this parameter so that modules and drivers may process more data without dropping packets or becoming backlogged.

ge:ge_intr_mode

Enable the GE driver to send packets directly to the upper communication layers rather than queueing.

```
minimum: 0
maximum: 1
default: 0 (Queue packets to upper layers)
```

When to Change:

When Solaris NCA is enabled, set this parameter to 1, so that the packet is delivered in interrupt mode for faster processing.

For example, a system with four gigabytes of memory and booted under 64-bit kernel should have the following setting in `/etc/system`. (Note: `pagesize(1)` can be used.)

```
set sq_max_size=0
set ge:ge_intr_mode=1
set nca:nca_conn_hash_size=82500
set nca:nca_conn_req_max_q=100000
set nca:nca_conn_req_max_q0=100000
set nca:nca_ppmax=393216
set nca:nca_vpmax=393216
```

Chapter 4

Performance Data

Independently published results show that the performance of a commercial Web server on a 900-Mhz Sun Fire™ 280R with the Solaris NCA is 73 percent better than a one-gigahertz Intel-based Hewlet Packard system with Linux that is running the same commercial Web server. Both of these systems were tested without the LSO network interface cards (NIC). These results are posted at:

<http://www.spec.org/osg/web99/results/web99.html>.

Chapter 5

Conclusion

The Solaris NCA can improve the performance of most popular Web servers, making it easier to efficiently manage resources and provide a higher level of service. It allows service providers to manage Web services across a heterogeneous network, without having to port to multiple Web servers. Solaris NCA also reduces the number of systems required to do a particular task, enabling better utilization of existing resources and resulting in lower cost of ownership.

Chapter 6

References

The following sections cover the files and components that are need to use Solaris NCA. Also, specifics about how Solaris NCA interacts with the Web server are included.

1. `/etc/hostname.*` Lists all physical interfaces configured on the server.
2. `/etc/hosts` Lists all host names associated with the server. Entries in this file must match with entries in `/etc/hostname.*` files for Solaris NCA to function.
3. `/etc/init.d/ncad` Solaris NCA door server startup script run when a server is booted.
4. `/etc/init.d/ncakmod` Solaris NCA server startup script run when a server is booted.
5. `/etc/init.d/ncalogd` Solaris NCA logging startup script run when a server is booted.
6. `/etc/nca/nca.if` Lists the interfaces on which Solaris NCA is run (see the `nca.if(4)` man page).
7. `/etc/nca/ncakmod.conf` Lists configuration parameters for Solaris NCA (see the `ncakmod.conf(4)` man page).
8. `/etc/nca/ncalogd.conf` Lists configuration parameters for Solaris NCA logging (see the `ncalogd.conf(4)` man page).
9. `/usr/bin/ncab2clf` Command used to convert data in the log file to Common Log File format (see the `ncab2clf(1)` man page).
10. `/usr/lib/net/ncaconfd` Command used to configure Solaris NCA to run on multiple interfaces during boot (see the `ncaconfd (1M)` man page).
11. `/var/nca/log` Holds log file data; in binary format, so do not edit this file.

Copyright 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, Sun Fire and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, Sun Fir et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, CA 94303-4900 USA Phone 800 786-7638 or +1 512 434-1577 Web sun.com



We make the net work.

Sun Worldwide Sales Offices: Africa (North, West and Central) +33-13-067-4680, Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India-Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +82-2-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-45-15-000, New Zealand-Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China-Beijing +86-10-6803-5588; Chengdu +86-28-619-9333; Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-596-9900, Sweden +46-8-631-10-00, Switzerland-German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888, Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44-1-276-20444, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800