

# Sun Chili!Soft ASP Scalability and Performance

---

## Abstract

This paper takes a look at the technical architecture of Sun Chili!Soft ASP and its performance oriented features, and examines the horizontal and vertical scalability of Sun Chili!Soft ASP.

---



901 San Antonio Road  
Palo Alto, CA 94303  
1 (800) 786.7638  
Sun Microsystems, Inc.  
1.512.434.1511

Copyright 2000 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.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Solaris, and Solstice PC-Admin 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.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**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 NONINFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

# Table of Contents

---

<b>Executive Summary</b>	<b>4</b>
<b>Sun Chili!Soft ASP Architecture</b>	<b>5</b>
<i>Multi-Process or Multi-Threaded</i>	5
<i>ASP Page Pre-Compilation</i>	6
<i>ASP Page Caching</i>	6
<i>Object Caching</i>	6
<i>Database Connection Pooling</i>	7
<b>Horizontal Scalability Options</b>	<b>8</b>
<i>Software Solutions</i>	8
<i>Hardware Solutions</i>	10
<b>Vertical Scalability Options</b>	<b>11</b>
<b>Conclusion</b>	<b>13</b>
<i>About ASP</i>	13
<i>About Sun Chili!Soft ASP</i>	13

## Executive Summary

---

When Sun Chili!Soft ASP moved the Active Server Pages framework from the Windows NT/2000 world to the UNIX and Linux world, it was clear that some changes had to be made. The architectural model that performed the best on Windows was not necessarily the best for UNIX or Linux. With the newest version of Sun Chili!Soft ASP, the architecture of the core engine has been enhanced to maximize its flexibility to suit the differing needs of highly varied computing environments.

Sun Chili!Soft ASP supports an architecture that provides a multi-process engine for servicing requests for ASP pages. The engine automatically spawns new threads to meet demand, and a task manager controls how requests are sent to different threads. Sun Sun Chili!Soft ASP also supports an architecture that allows the Sun Chili!Soft ASP engine to run on a separate machine from the Web server. Finally, Sun Chili!Soft ASP also supports such features as page pre-compilation, page caching, object caching and database connection pooling.

Sun Chili!Soft ASP supports various models for horizontal scalability and load balancing, including both software- and hardware-based solutions. For example, Sun Chili!Soft ASP can support software-based solutions such as round-robin DNS or load balancing software. Sun Chili!Soft ASP also can be used with hardware-based scalability and load balancing products. Each of these type of products support the concept of “sticky sessions”, necessary for session-based applications to function properly.

Sun Chili!Soft ASP can also scale vertically on a single machine, along with the underlying operating system. Sun Chili!Soft ASP can fully exploit the scalability and performance improvements that are available through different operating systems. In test's on a multi-processor Solaris box, for example, Sun Chili!Soft ASP's performance scaled to within 5% of linear as additional processors were added.

# Sun Chili!Soft ASP Architecture

---

Sun Chili!Soft ASP offers a number of architectural elements to enhance its scalability and performance. These include:

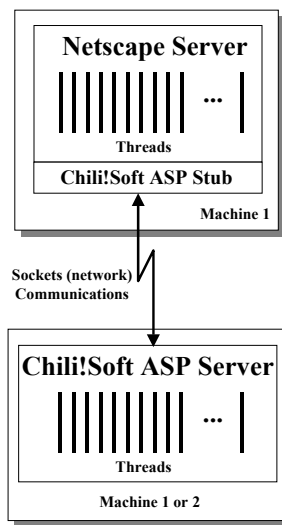
- Multi-threaded engine,
- Page pre-compilation,
- Page caching,
- Object caching, and
- Database connection pooling.

Let's take a closer look at each of these different elements.

## Multi-Threaded Engine

The Sun Chili!Soft ASP architecture provides a powerful multi-threaded engine. The Sun Chili!Soft ASP Task Manager is responsible for routing new ASP page requests to the appropriate thread.

Sun Chili!Soft ASP is able to spawn new threads for servicing page requests as the load increases, up to a configurable maximum number. A diagram of the multi-threaded iPlanet/Netscape server running against a multi-threaded Sun Chili!Soft ASP engine configuration appears to the



left. In this configuration, the Sun Chili!Soft ASP “stub” (an NSAPI server extension) uses network communications in order to pass ASP page requests to the Sun Chili!Soft ASP engine for processing. Sun Chili!Soft ASP can also run in-memory communications between the Web server extension and the Sun Chili!Soft ASP Server.

Sun Chili!Soft ASP’s multi-threaded engine realizes significant performance improvement of ASP pages that make heavy use of Application objects. An Application object can be used to maintain information that is common to all users accessing a defined set of ASP pages (an ASP “application”).

Another important detail for Sun Chili!Soft ASP’s performance and scalability is the option to run Sun Chili!Soft ASP on a physically separate machine. It is possible to configure an environment with the Web server(s) running on a different machine than the Chili!Soft ASP engine. In this case, the Sun Chili!Soft ASP extension or Task Manager

communicates with one or more other machines running Sun Chili!Soft ASP via network communications. This configuration provides an option for adding additional Sun Chili!Soft ASP servers to handle additional demand.

### **ASP Page Pre-Compilation**

A performance feature built into Sun Chili!Soft ASP is pre-compilation of ASP pages. When an ASP page is placed on the server, it is not compiled by the Sun Chili!Soft ASP engine until it is requested for the first time. When a request comes in, Sun Chili!Soft ASP compiles the page into bytecode that can be more quickly processed by Sun Chili!Soft ASP when subsequent requests for the page comes in. Pages in either VBScript or JScript are compiled into bytecode.

### **ASP Page Caching**

After the ASP page has been compiled, it can be cached by the Sun Chili!Soft ASP engine. Sun Chili!Soft ASP can be configured to cache all requested pages, cache a set number of pages, or cache no pages at all. Typically, Sun Chili!Soft ASP is configured to cache all of ASP pages that are requested. This is optimal for small sites, or sites that have a small number of heavily-used pages. However, as the number of pages grows, the Sun Chili!Soft ASP engine may end up caching (and storing in memory) a lot of pages that are rarely requested. To avoid this, Sun Chili!Soft ASP may be configured to only cache the most recently used pages. By specifying the number of pages to cache, large, busy sites can control the memory usage of the Sun Chili!Soft ASP server (improving performance), while still ensuring that the most popular pages are kept in the cache.

### **Object Caching**

Sun Chili!Soft ASP offers the ability to store objects with either session scope or application scope. Session scope maintains an object that is specific to a particular user, while application scope maintains an object that is particular to an ASP application. An ASP application is a group of files and directories that make up a logical application. By maintaining frequently used objects in memory, Sun Chili!Soft ASP is able to limit the overhead necessary with creating and destroying objects as they are used. Sun Chili!Soft ASP also efficiently allocates and releases the resources used by objects to ensure that unused objects do not use up valuable memory.

## **Database Connection Pooling**

One of the most expensive operations in typical Web applications is accessing database information. Web application environments such as Perl or CGI will open a connection to the database, retrieve the data, and close the connection. Each subsequent request will do the same thing, repeatedly opening and closing the database.

The solution to this problem is database connection pooling. Sun Chili!Soft ASP is able to open a number of connections to the database that are shared across all of the users accessing the Web application. Instead of opening and closing a database connection for each request, Sun Chili!Soft ASP simply uses one of the connections that is already open for fulfilling the request. This dramatically improves the performance of applications that rely heavily on database operations.

## Horizontal Scalability Options

---

The classic model for providing horizontal scalability is to simply add additional servers to an overall “farm” of servers. The addition of user sessions, however, adds an element of complexity to the horizontal scalability picture. In order for ASP to maintain session information for a specific user, the user’s requests must be consistently routed back to the same machine with which the initial session was created. This is called “session-aware load balancing,” and can be done using either software or hardware solutions. Session-aware load balancing offers several distinct benefits:

- Applications can use Sun Chili!Soft ASP’s standard session management capabilities, and the Session object.
- The server managing the session is also storing the session information, making access to the session information very fast.
- State information can consist of simple variables or complex objects. (Most session management techniques are limited in the information they can store.)
- Session management is distributed across all of the servers in the Web farm, and servers under heavy load can be taken out of the rotation.

There are a few disadvantages to using session-aware load balancing to achieve horizontal scalability, such as:

- User-specific information is difficult to consolidate across all servers in the farm, without the use of a database server.
- User requests may not always be balanced evenly across all servers. (The “Law of Averages” typically takes care of this, however.)
- Sun Chili!Soft ASP’s Application object can only be used for read-only information if a single logical application is running across several different physical machines in the farm.

### Software Solutions

There are two primary techniques for handling session-aware load balancing, using only software: Round-Robin DNS and Clustering software.

#### ***Round-Robin DNS***

The Round-Robin DNS approach is implemented by setting up two or more servers that service a common logical-DNS host name. Requests to

DNS then resolve this single host name into one of the names of a server in the collection. For example, the host name “www.chilisoft.com” might be resolved into one of several web servers with the host names “web1.chilisoft.com”, “web2.chilisoft.com”, ..., up to “webX.chilisoft.com”. By routing incoming requests to each of these servers in round-robin fashion, the load is distributed across all of them.

When a request comes into one of the physical hosts, the server must redirect the request back to itself using its physical hostname (“web3.chilisoft.com”) instead of the logical hostname of the Web farm (“www.chilisoft.com”). As long as all of the Web pages in the site using relative Web links (<A HREF="/some/page.htm">) instead of absolute links (<A HREF="http://www.chilisoft.com">), the browser will issue all requests to the physical hostname. As long as the user keeps returning to the same physical host, their session information will be maintained.

This approach may seem rather restrictive by requiring developers to only use relative URLs in their Web applications. It also requires a web site to be meticulous in keeping the pages across the site consistent, it does not distribute load across servers in an intelligent fashion, and it can cause users difficulty in bookmarking a site (if the server they originally hit goes down, for example). However, the simplicity of this solution, its ease of implementation, and its session-aware abilities, can make it attractive.

### ***Clustering Software***

Clustering software can address many of the shortcomings of the round-robin DNS approach. Clustering software, like round-robin DNS, relies on a Web farm of servers running the same copies of software, and consisting of the same content. But clustering software is also able to make much more intelligent decisions on how to route new requests. Clustering software can:

- Automatically route requests to the proper server for session management,
- Measure server response time in order to distribute new requests to “underutilized” servers,
- Monitor servers for poor performance or shutdown, redirecting traffic away from them, and restarting them automatically
- Assist with maintaining consistency of content across all of the servers in the farm.

Much of the value in the clustering software products is in their ability to implement load-balancing algorithms based on a wide variety of metrics, and even to allow the user to define their own metrics and algorithms.

## Hardware Solutions

There are numerous hardware solutions on the market today for providing session-aware load balancing. These products can be referred to as “intelligent routers”.

Intelligent routers allow a group of servers to appear as a single, logical server. The IP address and host name of the intelligent router is registered with DNS (as, for example, “www.chilisoft.com”), while the IP addresses and hostnames of the servers in the group remain unpublished. When the intelligent router receives a request, it routes the request to one of the servers in the group. In this manner, a group of servers can appear to a user as a single server. Intelligent routers support fail-over methods and provide different options for distributing load.

A critical element for making intelligent routers “intelligent” is the ability to work with session-aware web applications. Most routers offers “sticky” sessions that basically ensures that the same client gets the same server for the duration of the session. If the “sticky” session is set to 5 minutes, all requests from a specific client will be routed to the same server until the client is inactive for a period of 5 minutes. Developers using Sun Chili!Soft ASP with an intelligent router should make sure that the intelligent router’s “sticky” time interval is the same or greater than Sun Chili!Soft ASP’s session timeout value.

## Vertical Scalability Options

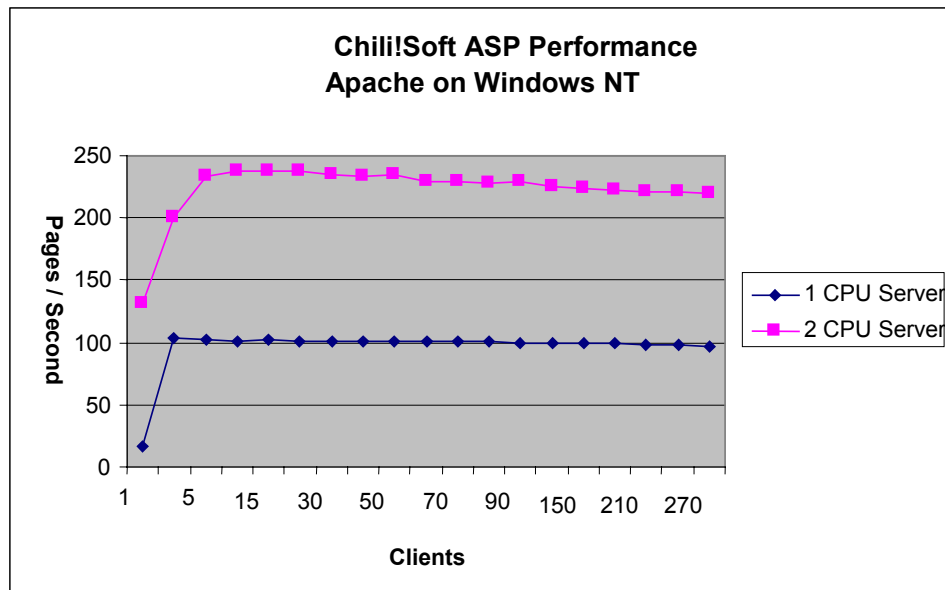
---

Horizontal scalability has appeal where the Web serving environment relies on relatively inexpensive hardware and software. In this environment, adding a new box to the Web farm is fairly inexpensive and easy to accomplish. However, horizontal scalability approaches do have their drawbacks. For example, they require an increasing number of boxes that have to be individually configured and maintained, which can become a significant burden. As the number of machines increases, the overall environment can become significantly more complex.

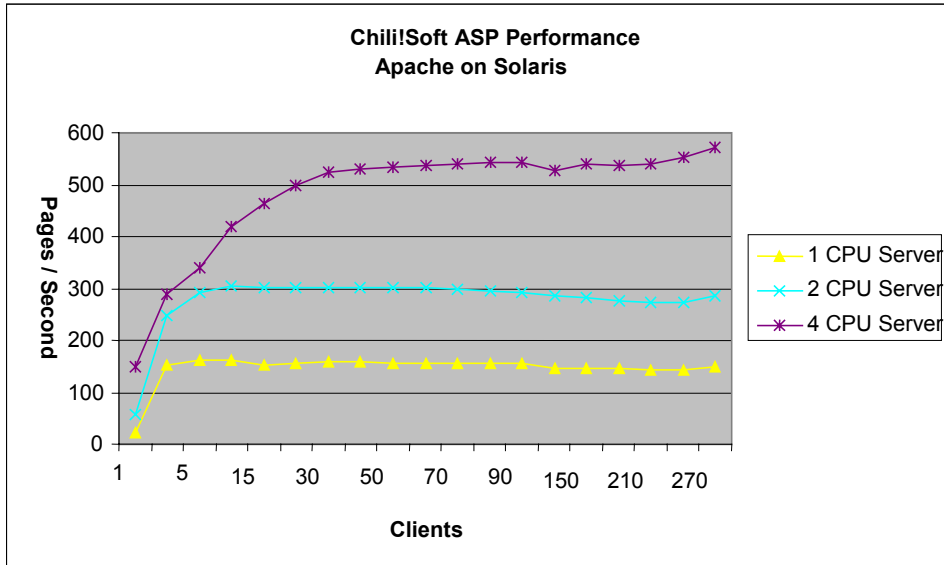
Furthermore, as we have seen, horizontal scalability options make it more difficult to efficiently consolidate information across servers, and in the case of ASP, the Application object's functionality can't be used if a logical application is running across several different physical machines.

So for some environments, vertical scalability is the end goal. Vertical scalability can be loosely defined as providing the ability for one machine to serve an increasing load, without adding additional machines. In the case of Sun Chili!Soft ASP, with the exception of the performance features mentioned in the opening section, vertical scalability is a function of the underlying operating system and hardware platform. The usual technique for providing scalability on a specific machine is to add hardware and memory.

In internal testing, Sun Chili!Soft ASP has shown the ability to fully exploit the scalability of the underlying operating system and platform. For example, the following chart displays performance results for a Windows NT system with 1 and 2 CPUs.



The chart above demonstrates that the throughput (pages/second) for Sun Chili!Soft ASP on Windows NT against 1 and 2 CPUs is actually better than linear. The next chart shows the results on a Sun Solaris box, running against 1, 2, and 4 CPUs.



This chart also demonstrates that Sun Chili!Soft ASP's throughput scales vertically to within 5% of linear as additional processors are added. Adding additional processors to an existing server installation provides a fast, easy, and relatively inexpensive option for accomplishing vertical scalability of Sun Chili!Soft ASP.

## Conclusion

---

Sun Chili!Soft ASP has a number of internal features that enable it to serve the needs of demanding Web environments. Furthermore, Sun Chili!Soft ASP is compatible with several methods for achieving horizontal scalability, and shows excellent ability to scale vertically along with the capabilities of the underlying platform.

## About ASP

ASP is an open, compile-free Web application environment that combines scripting, HTML custom server components, and database publishing to create dynamic Web-based business applications. With ASP, developers can build browser-independent Web solutions and publish databases to the Web using tools such as Microsoft's FrontPage™, Macromedia™ Dreamweaver UltraDev, or Adobe™ GoLive!.

## About Sun Chili!Soft ASP Software

Sun Chili!Soft ASP is the functional equivalent of Microsoft's ASP. While Microsoft ASP is available exclusively on Microsoft's Internet Information Server (IIS) web server on Windows NT/2000, Sun Chili!Soft ASP enables ASP on other leading Web servers and operating systems. Sun Chili!Soft ASP is supported by industry leading Web development tools such as Microsoft FrontPage, Macromedia™ Dreamweaver UltraDev, and Adobe™ GoLive!, and provides ASP functionality to iPlanet, Apache, Zeus and other Web servers.

For more information, please go to <http://www.chilisoft.com>.