

# 1 3.0 IC WebServices

## 1.1 VMWare ESX 3.5



## 1.2 Introduction

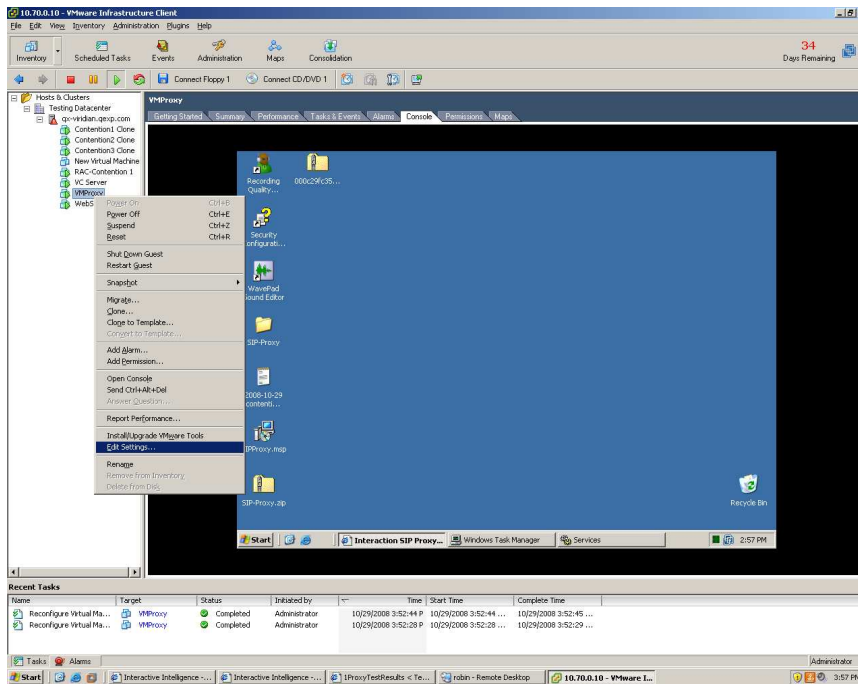
Interaction WebServices 3.0 is installed and used on a web server (such as IIS or Apache) to allow users connecting via a browser to send Interactions (such as chats) to IC and thus onto an interfacing IC Client. The best performance required a reservation on the virtual machine.

**NOTE:** At high CPU utilization (above 90%) and definitely at the 4800 user level the App pool gets overwhelmed and will throttle down to compensate. IIS 6.0 had the HTTP Keep-Alive specification enabled (that maintains client connectivity) and was also set to unlimited connections. Thankfully, with proper IIS configuration the Web Services default page is still reachable although slower to load.

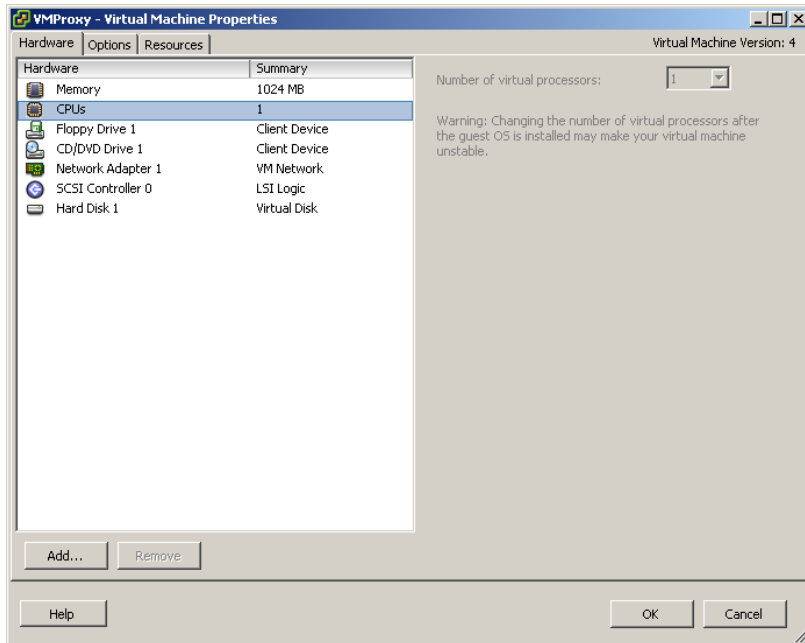
## 1.3 Setup and Configuration

Exhaustive technical documentation has been written on Web Server farms and guaranteeing a quality of Web Service in a robust web environment. In this test we are isolating IC WebServices 3.0 to a single virtual machine running a single CPU core.

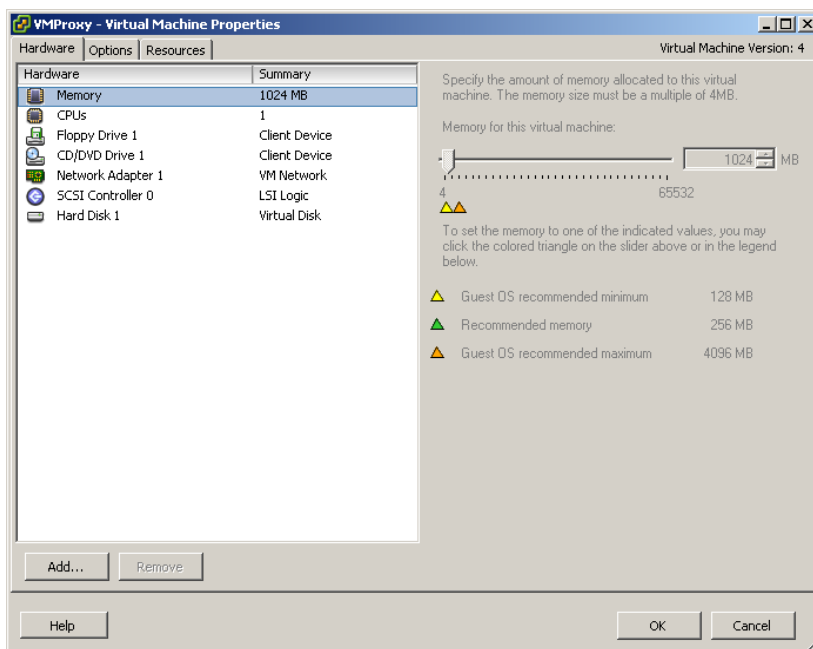
Use the VMWare Infrastructure Client to set the reservation for the Web Server running IC WebServices. Right click on the Server and select > Edit Settings.



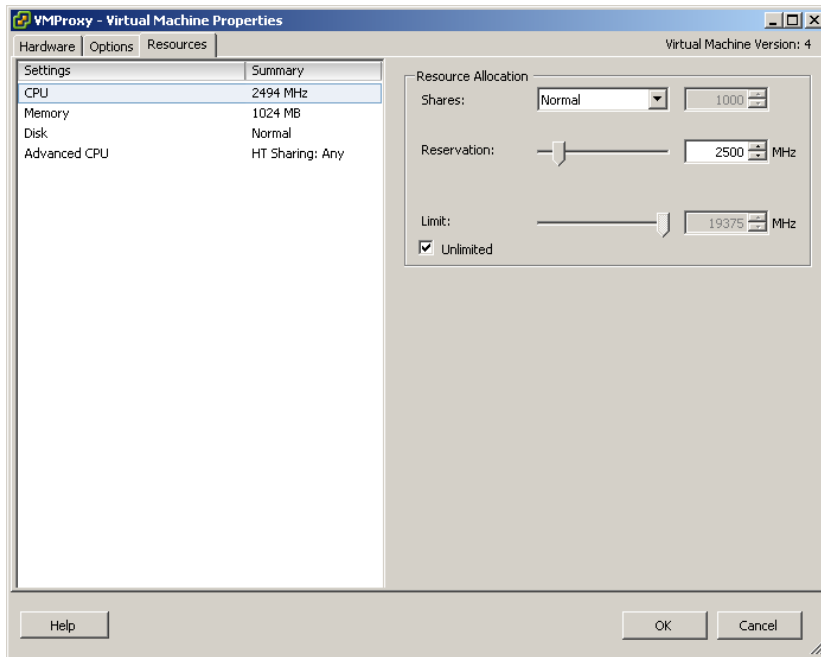
In the Hardware Section, select CPUs. Only 1 Virtual Processor should be required for performance at up to 4800 users.



In the Hardware Section, select Memory. The amount of memory can be set to as low as 512 but 1024 would be preferable.



Now select the Resources tab. Set the Reservation to the CPU of one of the cores. Example: If the ESX Host Server has 8 cores @ 2.5Ghz make this setting 2500.



## 1.4 Test Results

### Testing Results with 3.0 IC WebServices on VMWare ESX 3.5

Testing was performed with multiple tools, including a Microsoft-issued IIS web stress tool, a Java Bulk Chatter, and the Multisession Sim and Controller. The Microsoft WAS web stress tool is designed to realistically simulate multiple browsers requesting pages from a web site. This tool is used to gather performance and stability information about our WebServices web application.

During a ten minute load test, specific monitoring of the W3WP.exe (IIS App pool) CPU and threads under serious load (both direct with the web stress tool, and indirect with server resource contention from other Virtual Machines) will be applied.

**Max CAPS** is defined as maximum Connection Attempts Per Second (generally html plus images) and **Current Connection Average** pertains to the number of simultaneous connections maintained by the service during the sample interval. Both are Web Services Object Counters in Perf Mon.

The testing range of browser-connected sessions starts at 1200 users and scales to 4800 in 1200 user increments.

Users	Reservation?	CPU Contention?	Java Bulk Chat	Ave. CPU	Max. CAPS	Current Connection Ave
1200	No	No	No	78.7	4197	1029
1200	Yes (Single Core 2.5 Ghz)	Yes	No	35.03	2206	889
1200	Yes	No	Yes	88.03	4157	1045

Users	Reservation?	CPU Contention?	Java Bulk Chat	Ave. CPU	Max. CAPS	Current Connection Ave
2400	No	No	No	97.44	4530	1641
2400	Yes (Single Core 2.5 Ghz)	Yes	No	62.07	4991	1631
2400	Yes	No	Yes	95.07	4855	1693

Users	Reservation?	CPU Contention?	Java Bulk Chat	Ave. CPU	Max. CAPS	Current Connection Ave
3600	No	No	No	98.169	6689	2316
3600	Yes (Single Core 2.5 Ghz)	Yes	No	72.3	4225	2146
3600	Yes	No	Yes	97.01	5045	2497

Users	Reservation?	CPU Contention?	Java Bulk Chat	Ave. CPU	Max. CAPS	Current Connection Ave
4800	No	No	No	99.65	45754	4465
4800	Yes (Single Core 2.5 Ghz)	Yes	No	79.92	5227	4966
4800	Yes	No	Yes	99.05	14109	3609

## 1.5 Additional Information

The results above are a concise summary of many test runs and other analysis of performance.