



ITM 6.x – Scenario-based troubleshooting

Scenario #3 : High CPU utilization

Yew Hoong Ng – Global Response Team Asia Pacific



Scenario #7

- High CPU utilization
- Main factors contributing to high CPU usage:
 - ▶ Situations
 - ▶ Historical Data Collection
 - ▶ Policy



General hints

- Reduce the total number of distributed situations if you have a large number of connected agents
- Make your situation efficient
 - ▶ Use duperization if possible
 - ▶ Avoid using wildcard characters in situations whenever possible.
 - ▶ Properly specify your situation collection intervals
 - ▶ Disable the autostart flag for unneeded situations
- Configure historical data to keep short-term history files at the agent, if possible, rather than at the TEMS
- Set longer collection intervals for tables that collect an huge amount of data
- Enable warehouse collection only for attribute groups are interested
- Reduce the frequency of historical data collection



High CPU on Windows

- Use the tools and data provided by Task Manager to identify the process causing high CPU usage. In the Processes tab you can reorder the processes by CPU usage.
- In the Perfmon GUI, activate the collection specifying the counters **%processor time** and **ID thread** and select as threads the ones associated to the processes highlighted on step above.
- Check in the Perfmon "view report" window which is/are the thread id which is consuming that much CPU. Take a screen shot of the view , ensuring that both the %processor and the ID thread are displayed.
- Additionally you may use the Process Explorer tool to get details for thread id.



Windows task manager

- **kdsmain.exe** process has 50 threads.
- Its process id is 2700.

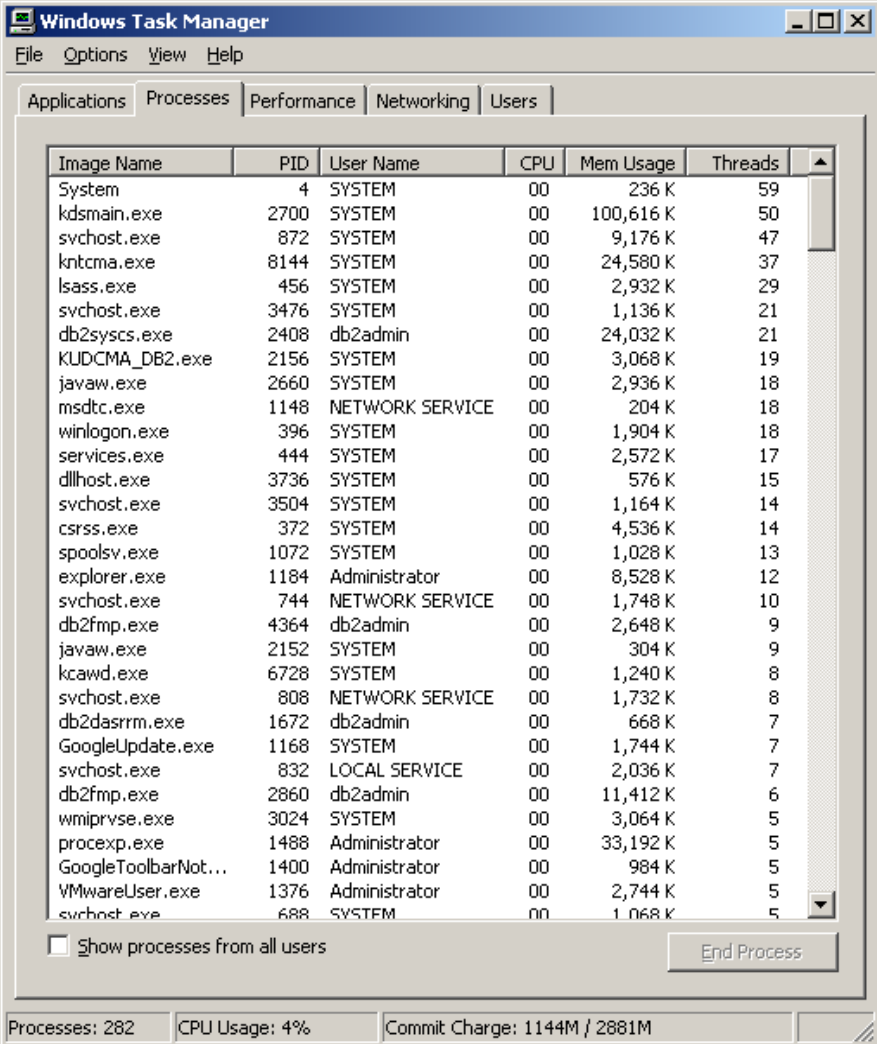


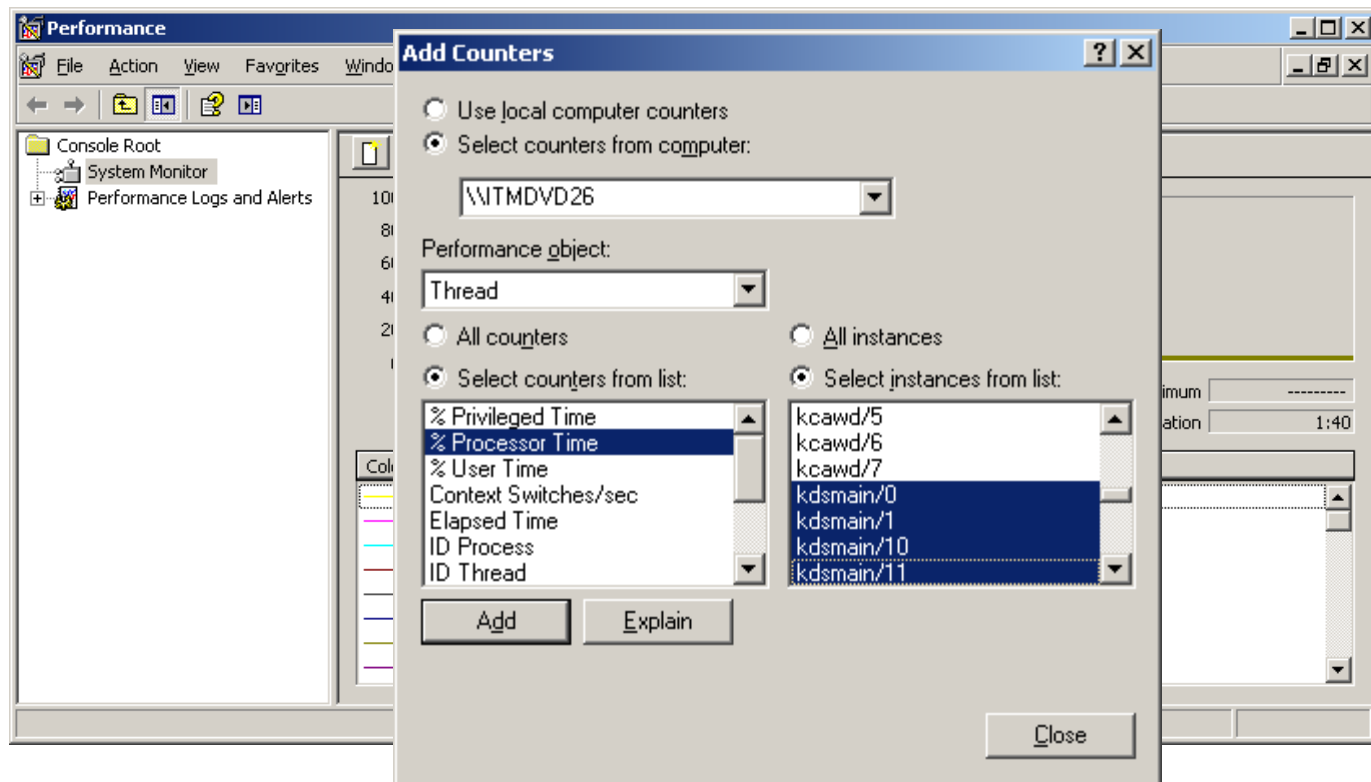
Image Name	PID	User Name	CPU	Mem Usage	Threads
System	4	SYSTEM	00	236 K	59
kdsmain.exe	2700	SYSTEM	00	100,616 K	50
svchost.exe	872	SYSTEM	00	9,176 K	47
ntcma.exe	8144	SYSTEM	00	24,580 K	37
lsass.exe	456	SYSTEM	00	2,932 K	29
svchost.exe	3476	SYSTEM	00	1,136 K	21
db2syscs.exe	2408	db2admin	00	24,032 K	21
KUDCMA_DB2.exe	2156	SYSTEM	00	3,068 K	19
javaw.exe	2660	SYSTEM	00	2,936 K	18
msdtc.exe	1148	NETWORK SERVICE	00	204 K	18
winlogon.exe	396	SYSTEM	00	1,904 K	18
services.exe	444	SYSTEM	00	2,572 K	17
dllhost.exe	3736	SYSTEM	00	576 K	15
svchost.exe	3504	SYSTEM	00	1,164 K	14
csrss.exe	372	SYSTEM	00	4,536 K	14
spoolsv.exe	1072	SYSTEM	00	1,028 K	13
explorer.exe	1184	Administrator	00	8,528 K	12
svchost.exe	744	NETWORK SERVICE	00	1,748 K	10
db2fmp.exe	4364	db2admin	00	2,648 K	9
javaw.exe	2152	SYSTEM	00	304 K	9
kcawd.exe	6728	SYSTEM	00	1,240 K	8
svchost.exe	808	NETWORK SERVICE	00	1,732 K	8
db2dasrm.exe	1672	db2admin	00	668 K	7
GoogleUpdate.exe	1168	SYSTEM	00	1,744 K	7
svchost.exe	832	LOCAL SERVICE	00	2,036 K	7
db2fmp.exe	2860	db2admin	00	11,412 K	6
wmiprvse.exe	3024	SYSTEM	00	3,064 K	5
proccp.exe	1488	Administrator	00	33,192 K	5
GoogleToolbarNot...	1400	Administrator	00	984 K	5
VMwareUser.exe	1376	Administrator	00	2,744 K	5
svchost.exe	688	SYSTEM	00	1,068 K	5

☐ Show processes from all users

End Process

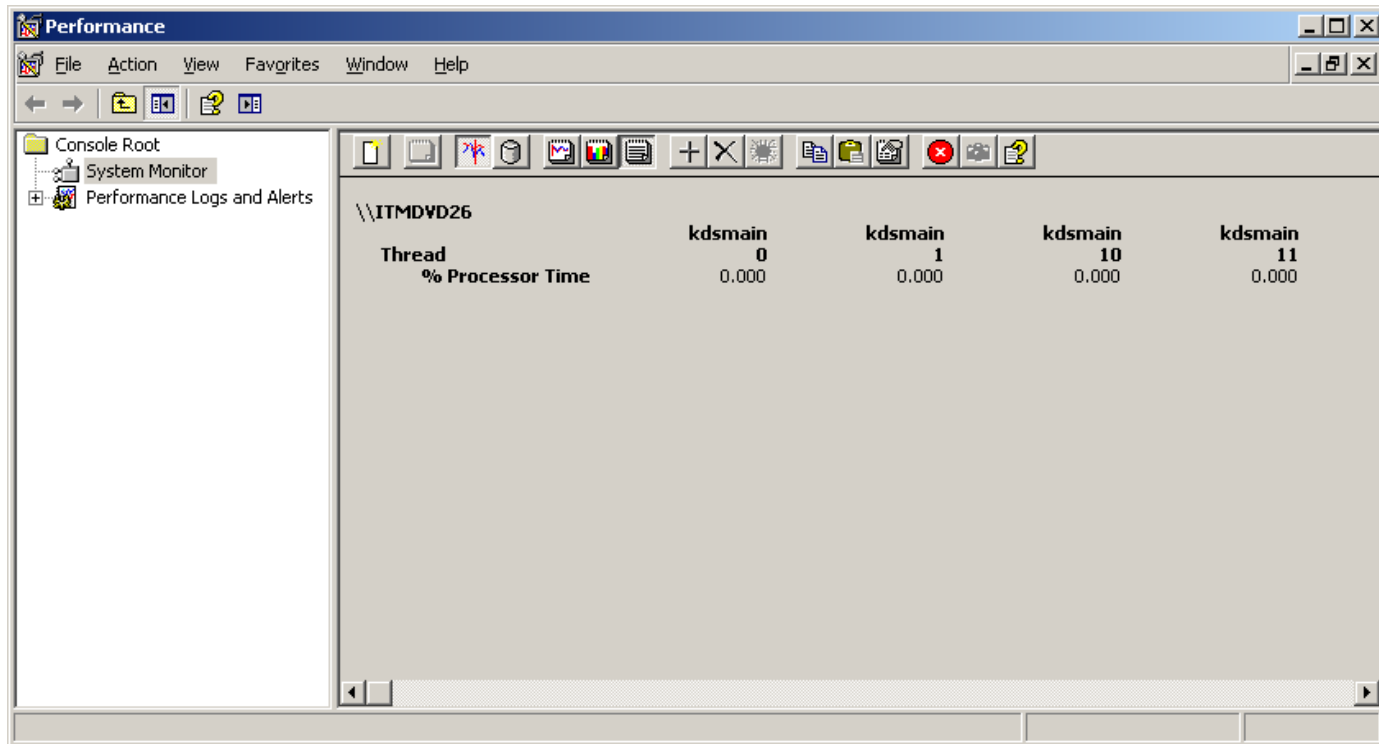
Processes: 282 CPU Usage: 4% Commit Charge: 1144M / 2881M

Configure Performance Monitor



- Run `perfmon.exe` from command prompt.
- Add counters and choose the following options:
 - ▶ Performance object : **Thread**
 - ▶ Counters : **% Processor Time**
 - ▶ Instances : **kdsmain/0** until **kdsmain/55**

Find the Thread



- Find the thread that uses the highest Processor Time.

Process Explorer

The screenshot shows the Process Explorer application window with the 'Process' tab selected. The process list is sorted by CPU usage, with 'kdsmain.exe' (PID 2700) highlighted in blue, indicating it is the selected process. The status bar at the bottom shows 'CPU Usage: 1.54%', 'Commit Charge: 39.62%', 'Processes: 282', and 'Physical Usage: 53%'.

The 'kdsmain.exe:2700 Properties' dialog box is open, showing the 'Threads' tab. The thread list shows 50 threads, with the first thread (TID 6216) selected. The thread details section shows the following information:

TID	CPU	CSwitch Delta	Start Address
6216			kdsmain.exe+0xd43ec
648			MSVCR71.dll!endthread+0x3e
3716			MSVCR71.dll!endthread+0x3e
1700			MSVCR71.dll!endthread+0x3e
7996			MSVCR71.dll!endthread+0x3e
7428			MSVCR71.dll!endthread+0x3e
6672			MSVCR71.dll!endthread+0x3e
3080			MSVCR71.dll!endthread+0x3e
7464			MSVCR71.dll!endthread+0x3e
7232	2		MSVCR71.dll!endthread+0x3e
3988			ntdll.dll!RtlConsoleMultiByteToUnicode...

The thread details section also shows the following information:

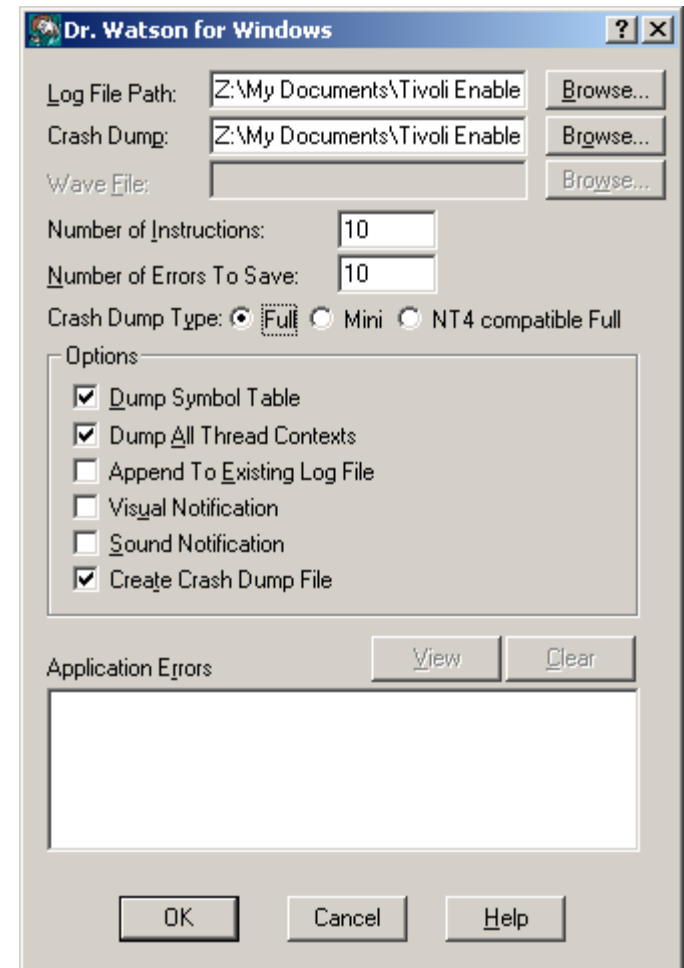
Thread ID	Start Time	State	Kernel Time	User Time	Context Switches
6216	4:23:15 AM 9/10/2010	Wait:UserRequest	0:00:07.703	0:00:05.125	1,099

The 'Permissions' tab is also visible, showing the thread's permissions. The 'Kill' button is highlighted in red.

- Download **Process Explorer** from <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>
- Double click on the process name → choose **Threads** tab
- Find the thread id that causes high CPU utilization

Dr. Watson

- Run `drwtsn32.exe` from command prompt.
- Change the location of **Log File** and **Crash Dump**.
- Choose **Full** for **Crash Dump Type**.
- Choose **Dump Symbol Table**, **Dump All Thread Contexts** and **Create Crash Dump File**.
- Click on **OK** button
- Run `drwtsn32.exe -p <pid>`, where `<pid>` is the process id.
- 2 files will be produced:
 - ▶ `drwtsn32.log`
 - ▶ `user.dmp`



drwtsn.log

- Thread id 0x1848 == 6216

```
*----> State Dump for Thread Id 0x1848 <----*
eax=632b3330 ebx=7ffd8000 ecx=0012f240 edx=00000000 esi=00000098
edi=00000000
eip=7c82860c esp=0012f49c ebp=0012f50c iopl=0          nv up ei pl
    zr na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000
    efl=00000246
0012f50c 77e61c8d 00000098 ffffffff 00000000
    ntdll!KiFastSystemCallRet
0012f520 64542a8a 00000098 ffffffff 77e61d48
    kernel32!WaitForSingleObject+0x12
0012f580 00404afe 0012fd64 0012fd68 00000000
    KLX!pthread_cond_wait+0x33a
0012f598 0040450e 0012fd64 0012f69c 004d78dc kdsmain+0x4afe
0012f8e0 00403df6 0012fcb4 0012fda4 0012fe14 kdsmain+0x450e
0012fe1c 00401239 00000001 003b2a10 00001601 kdsmain+0x3df6
0012ff60 004d452f 00000001 003b2a10 003b2cb8 kdsmain+0x1239
0012ffc0 77e6f23b 00000000 00000000 7ffd8000 kdsmain+0xd452f
0012fff0 00000000 004d43ec 00000000 78746341
    kernel32!ProcessIdToSessionId+0x209
```



High CPU on UNIX

- Identify the process id involved in the high CPU consumption

```
ps auxw | sort -r +2 | head -10 -
```

- Identify the thread id involved in the high CPU consumption

- ▶ **AIX**: `ps -mo THREAD -p <pid>`

- ▶ **Linux**: `top -p <pid> -H`

- ▶ **Solaris**: `prstat -p <pid> -L`

- Identify the thread stack associate to process id

- ▶ **AIX**: `procstack -F <pid>`

- ▶ **LINUX**: `gdb -F <pid>`

- ▶ **Solaris**: `pstack -F <pid>`

