FairCom
High-performance, engineering-level database technology

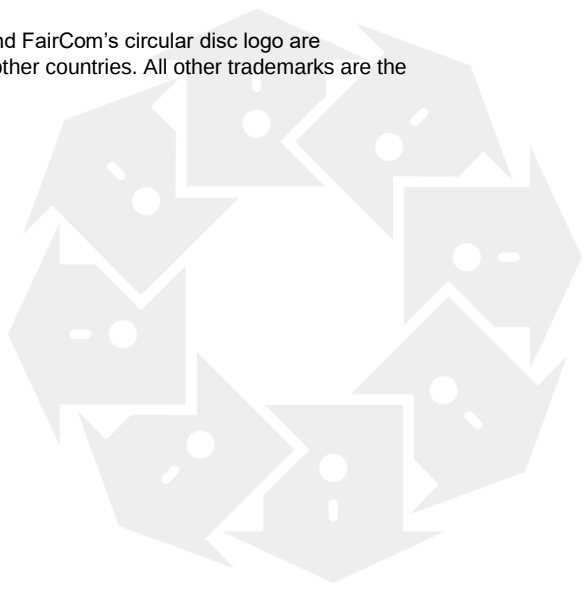
FairCom White Paper

FairCom's CPU Accounting - Definitions and Procedures

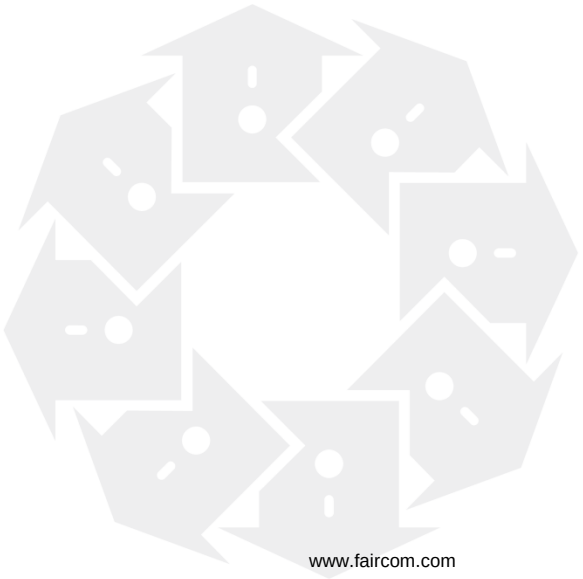
FairCom's c-treeACE Database Technology

FairCom Corporation
6300 West Sugar Creek Dr.
Columbia, MO
65203-9052 USA
(t) 573.445.6833
(f) 573.445.9698

North America • South America • Europe • Asia



Contents



1. CPU Counting

This white paper discusses the way c-treeACE handles actual physical processors, multi-core processors, multithreaded processors, and virtual machines.

Modern processor design allows multiple processor cores to be packaged in one integrated circuit chip. Each of those cores is essentially a complete CPU. Simultaneous multithreading (SMT) is a technique that allows a single processor core to execute multiple threads (Intel's hyper-threading is an example).

FairCom builds multi-threading into all c-treeACE Servers so they can take advantage of any available processors or threads. The architecture and performance (page 3) benefits are discussed later in this paper.

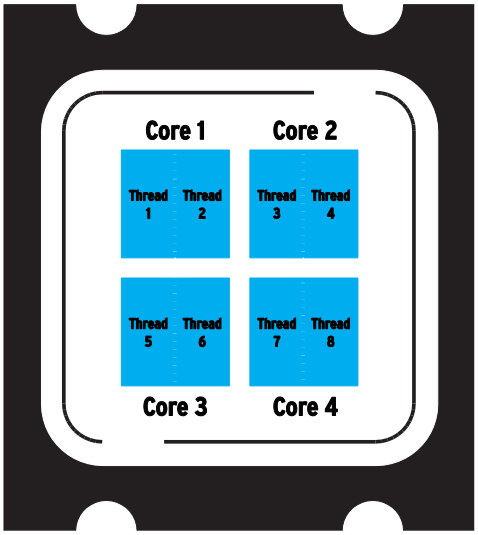
Licensing

The section titled Counting CPUs & Threads (page 4) explains of how processors are counted for CPU-based pricing. Virtualization (page 2), running more than one instance of an operating system on a physical server, is discussed from a licensing perspective.

1.1 CPUs and Threads

With years of experience developing multi-threaded applications, FairCom has built multi-threading into all c-treeACE Servers since the late 1980s. This allows c-treeACE Server to take advantage of any processors or threads available to it.

Simultaneous multithreading (SMT) is a technique some processor manufacturers use so that a single processor can execute multiple threads. Intel's hyper-threading technology is an implementation of SMT that supports two threads on each processor core.



A single CPU chip (1 "socket") with 4 cores and 8 threads

In the limiting case, SMT would be as effective as multiple processors or CPUs; in actual practice the performance gains depend on the application. In particular, applications that are I/O bound, such as databases that must wait for disks, can benefit from SMT because one thread can execute while another is waiting for I/O.

Modern processor design allows multiple processor cores to be packaged in one integrated circuit chip. Each of those cores is essentially a complete CPU. For example, a single chip may contain four processor cores, so it is equivalent to four separate CPUs. In addition, each core may use SMT, causing it to appear to be two or more processors. The result is a single chip could contain four cores each of which appears as two processors, for a total of eight processors reported by the operating system. A large server could use more than one of these chips, resulting in a large number of reported processors.

The amount of performance improvement provided by these technologies depends on the specifics of your application.

1.1.1 Virtualization

From a licensing perspective, virtualization is an issue related to CPU counting.

Virtualization is broadly defined as running more than one instance of an operating system on a physical server machine. This gives one physical server the ability to run multiple applications each inside of its own virtualized environment.

Each virtual environment that requires a c-treeACE Server must have its own server license. The c-treeACE Servers in each virtual environment can have unique user, connection, and CPU counts if desired.

Reporting

On a machine running a VMware server or IBM's VIOS manager, the operating system of the virtual machine reports the number of processors assigned to the virtual machine, not the total number of processors on the physical machine. In turn, the c-treeACE server reports only the number of processors indicated by the operating system.

1.2 Performance

c-treeACE Server takes advantage of any processors available to it. The performance depends on the saturation point with your specific application. Specifically, performance depends on whether your application is bound by CPU, memory, or disk I/O.

We have seen situations where simultaneous multithreading (SMT) with two threads on a single CPU core provides close to the same performance as two CPU cores, and we have seen situations where it does not. The only way to definitively understand your performance is to run your application in both configurations and monitor the system throughput.

The FairCom c-treeACE statistics monitoring program, **ctstat**, provides a good way to monitor your application's throughput. You will need to run your application for a period of time in both manners to see which yields the best throughput. Documentation for **ctstat** can be found in the *c-treeACE Server Administrator's Guide http://docs.faircom.com/doc/ctserver/52856.htm*.

To use this utility, use the *-text* option as it dumps everything c-treeACE monitors except for the function timings. Note you can enable function timings using the **ctstat** *-wrktime on* support, but this will impact performance as the function timings are a costly exercise. The rest of the monitoring does not impact performance because the Server is already keeping these metrics.

The output from **ctstat** *-text* will go to a text file called *snapshot.fcs*. With SMT enabled, run this utility 5 or 6 times every 10 minutes over the course of an hour (or longer as you feel appropriate). Move the *snapshot.fcs* file from your first run to a new location. Then repeat the same exercise with the application running the same tests with SMT disabled.

Now that you have two *snapshot.fcs* files, you can view them with an editor and compare selected metrics. The **ctsnpr.exe** utility allows you to read a *snapshot.fcs* and dump it to a comma-delimited file you can open in Excel. The usage for **ctsnpr** is:

```
ctsnpr.exe snapshot.fcs > snapshot1.out
```

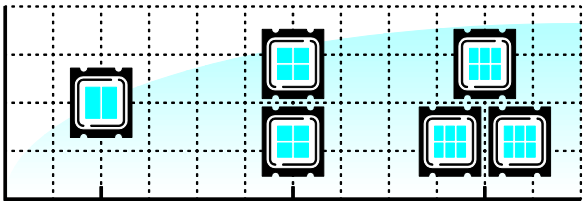
Repeat this for both *snapshot.fcs* files and open them in Excel to compare the results.

2. Counting CPUs & Threads

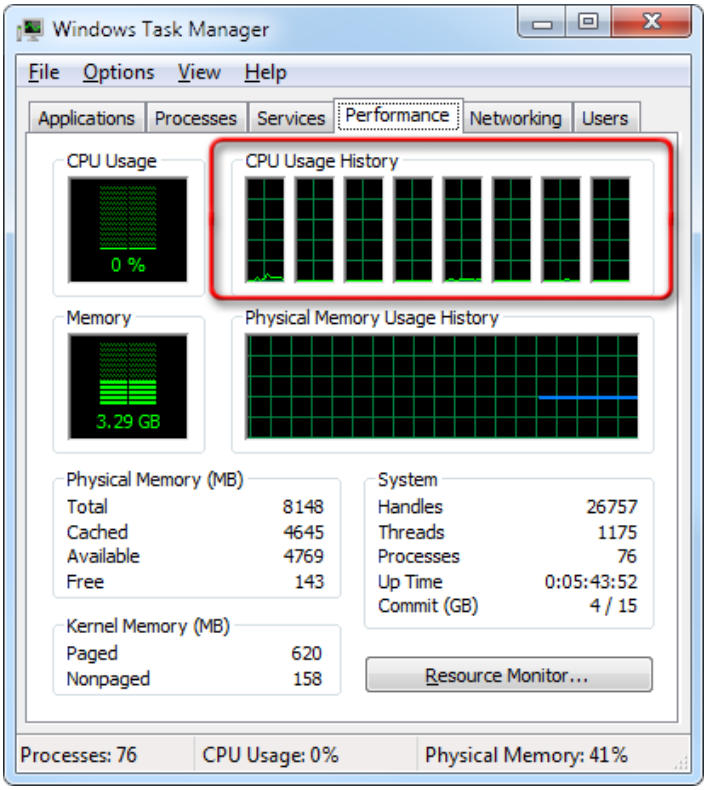
A question frequently arises about the definition of a CPU for purposes of counting.

When determining the number of CPUs in a host machine, **FairCom counts each CPU reported by the operating system** (be that a *physical CPU*, a *CPU core*, an *SMT thread* (page 2), or a *virtual machine* (page 2)).

Unlike many database vendors, we allow the end-user to bind the c-treeACE Server to a given number of CPUs in the machine. For example, a c-treeACE Server supporting 4 CPUs can be run on a machine with 8 CPUs by binding the c-treeACE Server to 4 of those CPUs. Further, the pricing for each additional CPU is an incremental price rather than a 100% increase as with most other database providers. This helps FairCom pricing to better reflect the benefits of multiple cores and threads.



In some cases, the system calls provided by the operating system do not distinguish between actual processors and SMT threads. This causes the operating system to report the number of CPUs as the number of processor cores times the number of SMT threads running on each core. For example, a system may have four processor cores each of which has two SMT threads causing a total of eight CPUs to be reported. For example, the Windows Task Manager shows eight CPUs on systems that have a single, quad-core, hyper-threaded chip:



2.1 AIX

An AIX system with two CPUs using the Workload Partitioning feature (WPAR), which supports two SMT threads per processor, will report four CPUs. The AIX system call that detects the number of CPUs counts the virtual processors created by the threads, which causes it to report four CPUs in this example. If the c-treeACE Server was licensed for two CPUs, the following message would be displayed:

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  LICENSE NOTICE:
  c-treeACE is licensed for 2 CPU's, but 4 CPU's have
  been detected in the host machine. Either upgrade
  the c-treeACE license to support a greater number
  of CPU's or bind c-treeACE to specific CPU's.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

One of the following options can be used to correct this situation:

1. Change the binding to a single CPU.
2. Disable the SMT support.
3. Purchase a larger c-treeACE Server license that supports 4 cores.

AIX does not support the CPU_AFFINITY keyword.

Binding to Specific CPUs

execrset

On AIX, the c-treeACE Server can be launched via the **execrset** command to operate with a given resource set.

Usage:

```
execrset [-P][-F] rsetname [-e] command [args]

execrset [-P][-F] -c cpuNum [cpuNum] [-m memNum [memNum]] -e command [args]
```

For example, **execrset** *-c 0 -e ctreesql* will launch the server and bind it to processor 0.

To use the **execrset** command, the user must have root permissions or have the CAP_NUMA_ATTACH capability assigned. To assign this capability to a user, the **chuser** command can be used.

bindprocessor

If the c-treeACE Server is already running, the **bindprocessor** command can be used. This command may also be used to query the list of available processors.

taskset

A process can be bound to a given set of CPUs by using the **taskset** command (see the *Linux section* (page 7)). This utility is installed as part of the *util-linux* package.

An AIX Program

The simple program listed below compiles on AIX 5.3 and later showing one way to programmatically determine the CPU count, whether or not the partition supports SMT, and whether or not it is on. This program could easily be incorporated into the server startup logic.

```
#include <stdio.h>
#include <libperfstat.h>

int main(int argc, char* argv[])
{
    perfstat_cpu_total_t cpustats;
    perfstat_partition_total_t partstats;

    if (!perfstat_cpu_total(NULL, &cpustats, sizeof(perfstat_cpu_total_t), 1)) {
        perror("perfstat_cpu_total");
        exit(-1);
    }

    if (!perfstat_partition_total(NULL, &partstats, sizeof(perfstat_partition_total_t), 1)) {
        perror("perfstat_partition_total");
        exit(-1);
    }

    printf("\nCurrent number of active CPUs: %d", cpustats.ncpus);

    if (partstats.type.b.smt_capable)
```

```
        printf("\nOS supports SMT mode");
    else
        printf("\nOS DOES NOT support SMT mode");

    if (partstats.type.b.smt_enabled)
        printf("\nSMT mode is ON\n\n");
    else
        printf("\nSMT mode is OFF\n\n");

    return(0);
}
```

To compile the program, use the command line:

```
- cc cpustats.c -o cpustats -lperfsta
```

2.2 Linux

On Linux, a process can be bound to a given set of CPUs by using the **taskset** command shown below. The CPU affinity is represented as a bitmask with the lowest order bit corresponding to the first logical CPU and the highest order bit corresponding to the last logical CPU. For example, 0x00000001 is processor #0 (first processor), 0x00000003 is processors #0 and #1, and 0x00000004 is processor #2 (third processor). You can use the *-p* option to set the affinity for an already running process, or you can start a process with affinity by specifying the mask and the command to follow as shown in the first example below.

Usage:

```
taskset [options] [mask | cpu-list] [pid | cmd [args...]]
```

Set or get the affinity of a process:

- *-p, --pid* - Operate on existing given pid
- *-c, --cpu-list* - Display and specify CPUs in list format
- *-h, --help* - Display this help
- *-v, --version* - Output version information

The default behavior is to run a new command:

```
taskset 03 sshd -b 1024
```

You can retrieve the mask of an existing task:

```
taskset -p 700
```

Or set it:

```
taskset -p 03 700
```

List format uses a comma-separated list of CPUs instead of a mask:

```
taskset -pc 0,3,7-11 700
```

Ranges in list format can take a stride argument, for example:

```
0-31:2
```

is equivalent to mask:

```
0x55555555
```

2.3 Solaris

On Solaris systems, `CPU_AFFINITY` accepts a single numeric value, which is interpreted as a processor set number. For example, `CPU_AFFINITY 2` configures c-treeACE to run on processor set 2.

> **Note**: To use `CPU_AFFINITY` under Solaris you need to run the c-treeACE Server with root permission. This because Solaris requires any process using a processor set to have such permission.

To create a processor set on Solaris, use the Solaris command `psrset`. For example, to create a set comprising processors 4 through 7, use:

```
psrset -c 4-7
```

where:

- *-c* - Create processor set.
- *4-7* - The processor numbers included in the set.

The ID of the newly created processor set is returned:

```
created processor set ps_id
```

To bind a process to this processor set, use:

```
psrset -b ps_id pid
```

where:

- *-b* - Bind.
- *ps_id* - The ID returned by the command when the processor set was created.
- *pid* - The ID of the process to be bound to the processor set.

If the process does not have permission to assign itself to the specified processor set (or if an invalid processor set is specified), c-treeACE logs the following message to *CTSTATUS.FCS*, where *<configuration_file_name>* is the name of the c-treeACE configuration file, *<line_number>* is the line number on which the `CPU_AFFINITY` option was specified, and *<error_code>* is the system error code returned by the OS.

```
Configuration error: <configuration_file_name>, line <line_number>: Failed to set CPU affinity:
system error code <error_code>.
```

2.4 Mac OS X

On Mac OS X, there is no check by c-treeACE for the processor count. There is no system command to set the CPU affinity for this platform.

2.5 Windows

On Windows systems, `CPU_AFFINITY` server keyword can be used to set the processor affinity mask for the c-treeACE process. The option accepts a comma-delimited list of processor numbers. For example: `CPU_AFFINITY 0,1,2,3,8,9,10,11` indicates that c-treeACE is to be run on the eight specified CPUs.

If c-treeACE successfully sets the CPU affinity to the specified CPUs, the following message is logged to *CTSTATUS.FCS*, where *<cpulist>* is the list of CPUs specified for the `CPU_AFFINITY` option:

```
Successfully set CPU affinity to: <cpulist>
```

The following error situations can occur when using the `CPU_AFFINITY` option:

If the list of CPUs specifies a CPU number that is out of range on the system, a message is logged to *CTSTATUS.FCS*:

```
Configuration error: <config_file_name>, line <line_number>: The CPU_AFFINITY option specifies
an invalid CPU number for this system.
```

3. Document History

02/12/2015	Added AIX program.
02/11/2015	Removed an unnecessary reference in the Linux section.
11/11/2014	Created separate sections for *Solaris* (page 8) and *Windows* (page 9) and added information specific to each.
06/20/2014	Document published.

Last published Wednesday, August 2, 2017.

4. Index