

# Performance requirements

The Ipssoft D2000 real-time application server can be used for a whole range of applications - from small SCADA systems (built on the Raspberry PI platform or on industrial computers with OS Windows/Linux) to large MES/EMS type systems with dozens of users, large application databases (in dimensions of TB units), multi-terabyte archives and depository databases. In this chapter, we will try to summarize some best practices for designing and managing D2000 systems.

## Virtualization

Today, the D2000 application server is often also operated in a virtualized environment (VMware, Hyper-V, Proxmox), especially for systems such as MES, EMS, SELT, and balance systems, more rarely for SCADA systems.

In this environment, resources are shared, resulting in two basic sharing problems:

- allocation of sufficient resources
- monitoring and diagnostics

The "hyper-converged architecture", in which we use powerful servers with local disks (in a RAID10 or RAID50 disk array), has proven to be successful. It is ideal when the virtualized D2000 applications have their own servers (which are not shared with other applications) and when the administrators of the D2000 applications not only have administrative rights to the virtual machines on which the D2000 is running but also have access to the virtualization environment so that they can perform operational performance diagnostics in case of problems.

In the case of several administrators (e.g. for network infrastructure and firewalls, virtualization, Active Directory, application servers), we recommend the introduction of an operation log in which all operations that could affect the functionality of the servers will be recorded, in particular:

- configuration changes in the network infrastructure
- configuration changes in virtualization, adding virtual servers, changes in allocation and limitation of resources (CPU, RAM)
- moving virtual servers, moving their disks
- configuration changes in disk arrays, adding additional guests
- configuration changes in the AD policy
- changes in antivirus and antimalware software settings
- software installations and upgrades
- firmware upgrades (servers, switches, firewalls)
- changes in server settings in BIOS (performance, security, other)

The subsequent analysis (usually if there is a slowdown and reduction in performance) is greatly facilitated by the existence of the operation log. To share the log, you can use e.g. SVN or GIT repository, SharePoint repository, and the like.

## Allocation of sufficient resources in a virtualized environment

The resources that the D2000 primarily needs are three: memory (RAM), CPU, and disk space (we haven't experienced any problems with bandwidth limits on network interfaces yet). We recommend:

- RAM - allocation of a sufficient amount of memory. It is ideal to reserve memory in a virtual environment, so that memory is not shared between virtual machines (so-called ballooning).  
Small D2000 applications need roughly 1 GB of RAM (the minimum for an application server on Windows/Linux is 4-8 GB), large ones several GB to tens of GB, depending on the number of configured objects, processes, and users. If more memory is available, we recommend allocating it to the SQL database for the archive (we recommend PostgreSQL) and the archive cache (we recommend several GB for the so-called [isochronous cache](#)). Applications using [EDA](#) technology (Energy Databank) benefit from several GB of memory allocated to the EDA server.
- CPU - the usage of CPU strongly depends on the nature of the application (constant CPU consumption for SCADA-type applications, significant peaks for balance systems or systems where user-triggered events take place - e.g. preparation of documents for monthly invoicing). In the case of physical servers, today's processors have enough necessary performance. In a virtualized environment, we encountered a case where VMware administrators artificially limited the maximum usable frequency for the balance system, because they thought it "consumed too much CPU". First, they caused significant user dissatisfaction (the preparation of invoicing documents took instead of 30 minutes several hours) and, on the one hand, the analysis showed that a significant part of the performance was consumed by the antivirus (ESET NOD), since it did not have configured exceptions.  
For large applications, it is advisable to allocate more vCPUs (4-8-16) - the D2000 architecture allows good parallelization (parallel tasks within the D2000 Kernel, D2000 Event, and D2000 Archive processes [configurable]).
- Disk space:
  - for the partition with the OS, we recommend approx. 20 GB (Linux) or 100 GB (Windows)
  - for the partition with D2000, we recommend at least 40 GB for the start, while the archive database usually has the largest consumption - several GB to several TB (for SELT systems, the monitoring database - up to tens of GB)
  - if [depository databases](#) are enabled in the D2000 Archive (storage of historical data with unlimited depth), we recommend a separate partition for the depository databases (the size of the depository databases will gradually grow on it). Currently, there are customers with more than 20 TB depository databases, but newer versions of the D2000 allow you to turn on [depository data compression](#), which usually has a compression ratio of 1:10 and better, which significantly saves disk space.

For the partition with the OS and D2000, we recommend fast disks (SSD), for the partition with safes, slower HDDs or NAS are also sufficient.

## Monitoring and diagnostics in a virtualized environment

In a virtualized environment, it is essential to have access to monitor operational parameters to ensure that the D2000 application does not suffer from resource sharing. We recommend that environment administrators monitor and be able to provide the following data (according to the graphs available in vCentre):

- graphs of the CPU load of D2000 servers, other virtual servers within the guest, and the total CPU load of the guests (to diagnose if there is a lack of CPU power)
- graphs of RAM consumption (proof that there is no ballooning - memory sharing between servers when there is a lack of RAM and subsequent swapping)
- graphs of I/O subsystem load (metrics: reads/writes per second, read/write data amounts [kB/s], read/write latencies) - again, for D2000 servers, other virtual servers within the guest, and if the storage in question is shared then a load of all hosts using disk storage (to diagnose whether there is a lack of I/O performance). Some shared storages provide their own load diagnostics from individual guests that can be used (LeftHand, 3PAR).

We recommend having all these graphs and data for them available for at least 3 months, for long-term performance monitoring.

In a virtualized environment, not only the speed but also the latency of the disks is important for the D2000 Archive. It should be noted that when archiving, hundreds and thousands of database tables for individual archive objects are written in parallel.

## Antiviruses

In the case of using antivirus and antimalware programs (Microsoft Defender, ESET Nod, Symantec, and others, on the Linux platform e.g. McAfee 'OAS Manager'), it is necessary to correctly reset exceptions so that antiviruses do not overload the CPU and slow down the functionality of D2000 systems.

Directory exceptions: by default, we recommend adding directories with D2000 and databases for D2000, e.g. on the Windows platform:

- C:\Program Files\PostgreSQL - installation of the PostgreSQL database
- D:\D2000 - D2000 installation
- D:\\_FTP - directory for FTP update
- D:\\_Backup - directory for creating backups

On the Linux platform:

- /opt/d2000 - D2000 installation + application directory (contains tablespace for Syscfg, Logfile, Archive)
- /var/lib/pgsql - installation of PostgreSQL databases

Exceptions for programs in memory - so that antiviruses do not try to analyze communication (external - D2000 KOM, between processes - D2000 Kernel, with databases - D2000 DbManager). We recommend adding exceptions to the D2000 processes that consume the most CPU, by default they are:

- *postgres.exe* - SQL database (oracle.exe process for Oracle DB, dbsrv12.exe and dbeng12.exe processes for Sybase SQL Anywhere 12)
- *kernel.exe*
- *kom.exe*
- *calc.exe*
- *dbmanager.exe, dbmanager\_ora.exe*
- *event.exe, event\_edathin.exe*
- *archiv.exe, archiv\_ora.exe*
- *gtwcli.exe, gtwsrv.exe*
- *tcts.exe*
- *alarm.exe*

On the Linux platform:

- *postgres, postmaster*
- *kernel*
- *kom*
- *calc*
- *dbmanager, dbmanager\_ora*
- *event, event\_edathin*
- *archiv, archiv\_ora*
- *gtwcli, gtwsrv*
- *tcts*
- *alarm*

In the case of some antiviruses (Microsoft Defender), it is advisable to monitor the total CPU consumption of the antivirus (*msmpeng.exe*) in the Task Manager. If it is high, exceptions are insufficient (and exceptions should be set for other processes, usually those that also have high CPU consumption). Other antiviruses (ESET NOD) work "undercover" and consume the CPU in the context of running processes - Task Manager thus shows e.g. high CPU consumption for *postgres.exe*.

There are also negative experiences with the program *xagt.exe* (FireEye Endpoint Security), which (probably due to missing exceptions) consumed quite a lot of CPU power (4 out of 16 available CPUs) and disabled several real-time communications ([IEC 870-5-101](#) and [IEC 870-5-104](#) protocols).

## Useful diagnostic tools

On the Windows platform:

- *Resource monitor* (available from *Task Manager*) - display statistics about CPU, memory consumption, disk operations and network usage

On the Linux platform:

- the *iostat* utility is used to display statistics of disk operations  
*iostat* (interactive display of current values for individual processes)

*iostat -ao* (display cumulative I/O statistics of individual processes)

- *ps* utility (display information about individual processes)  
*ps -eo pcpu,pid,user,args | sort -nk 1 -r | head -20* (display all processes - CPU usage, process PID, username and arguments - and sort by CPU, showing the 20 processes with the highest CPU usage)