Typical System Scenarios

The scenarios described in this section of the help demonstrate how Citect SCADA can be used to support typical processes found in primary production, utilities delivery, and manufacturing.

In reality, a project will incorporate a combination of the scenarios described here, with a high degree of customization and scalability. However, these examples have been simplified to demonstrate how Citect SCADA can be configured and deployed to meet the specific requirements of a production system.

System Type	Description
Standalone	Every component of a system runs on a single computer. See Standalone System.
Distributed I/O	Citect SCADA is used to monitor and manage distributed devices that are each connected to remote I/O servers. See Distributed I/O System.
Redundant Server	One or more of the servers associated with a system are duplicated and defined as primary and standby units, allowing the system to keep running in the event one of the servers becomes inoperative. See Redundant Server System.
Client-server	The servers and clients associated with a system are independently distributed across a number of computers on a network, offering greater accessibility and performance benefits. See Client-Server System.
Redundant and Distributed Control	Remote or geographically separate sections of a production system have fully operational sub-systems in place that are monitored and controlled locally. If such a sub-system becomes partially or wholly inoperative in a manner preventing local control, this arrangement allows remote Control Clients to take control of the affected sub-system. See Redundant and Distributed Control System.
Cluster Controlled	A production system is organized into discrete areas being monitored by operators within each area. However, there is also a level of control that supervises every area of the system. See Clustered Control System.
Load Sharing System	The system splits the load of an otherwise stressed system across multiple machines, better utilizing the available infrastructure. See Load Sharing System.

Standalone System

A standalone installation of Citect SCADA runs every server and client component of a system on a single computer. These include:

- I/O server
- Alarm server
- Trends server
- Reports server
- Control client.

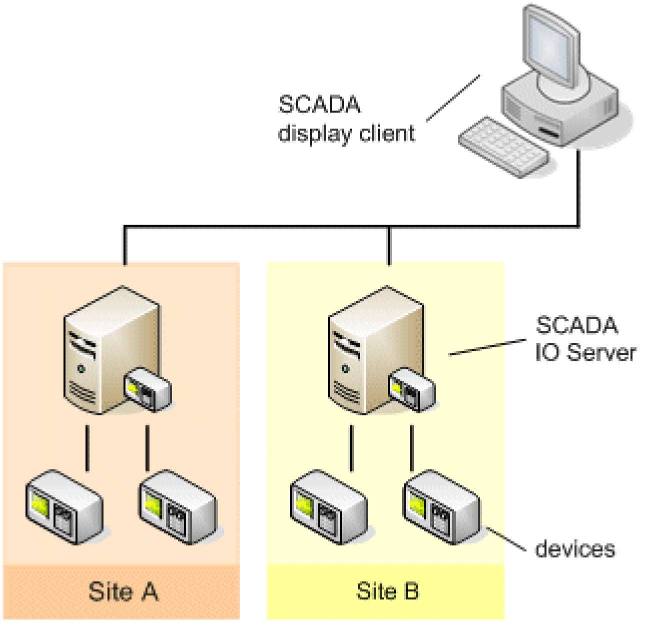
This allows Citect SCADA to be run as a small, self-contained system.

> **Note:** You can run the server and client components of a standalone system as a single-process or multi-process system. It is recommended that a single- process setup only be used as a short term solution for your control system, or to run demonstrations and test projects. Adding redundancy to your system will make it more reliable and more efficient.

Distributed I/O System

This scenario demonstrates a method of connecting Citect SCADA to a number of devices that are distributed across several sites over a wide geographical area.

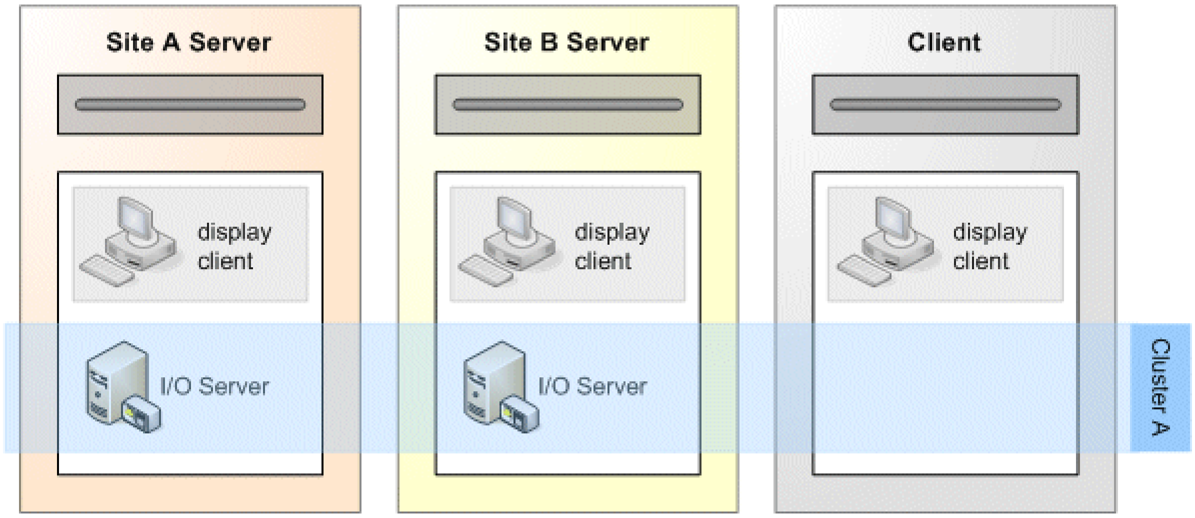
Instead of attempting to connect devices directly via a remote connection, an I/O server is placed at each site, enabling communication to be managed within the system.



This model is also useful in plants that contain devices with a serial port or limited communications capabilities. By placing I/O servers on the factory floor to interface with these devices, you can optimize communications on slow or low-bandwidth networks and improve overall performance.

Despite the geographical distribution of I/O servers across many sites, this type of system can be configured as a single cluster system, as a cluster is able to support many I/O servers.

The diagram below demonstrates how to approach the deployment of this type of system across the server machines using a single cluster.

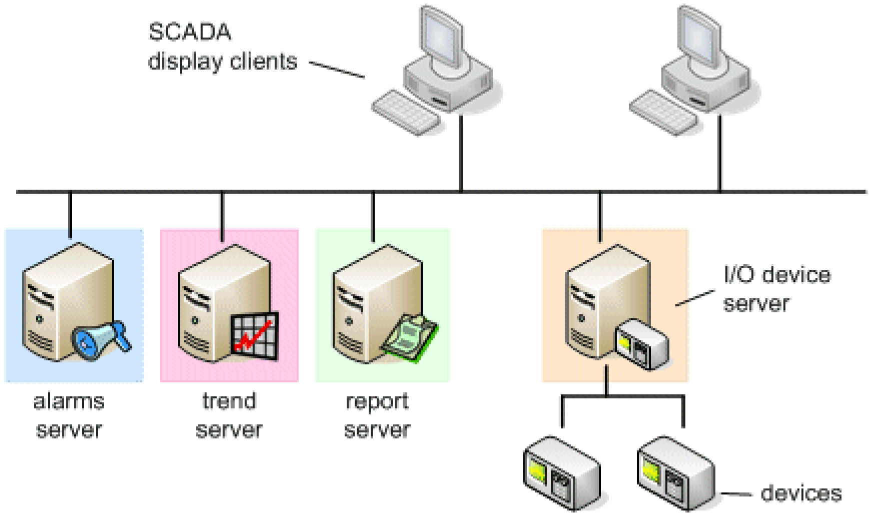


A second cluster will only become necessary if your project requirements call for more than one redundant pair of alarms, trends or reports servers.

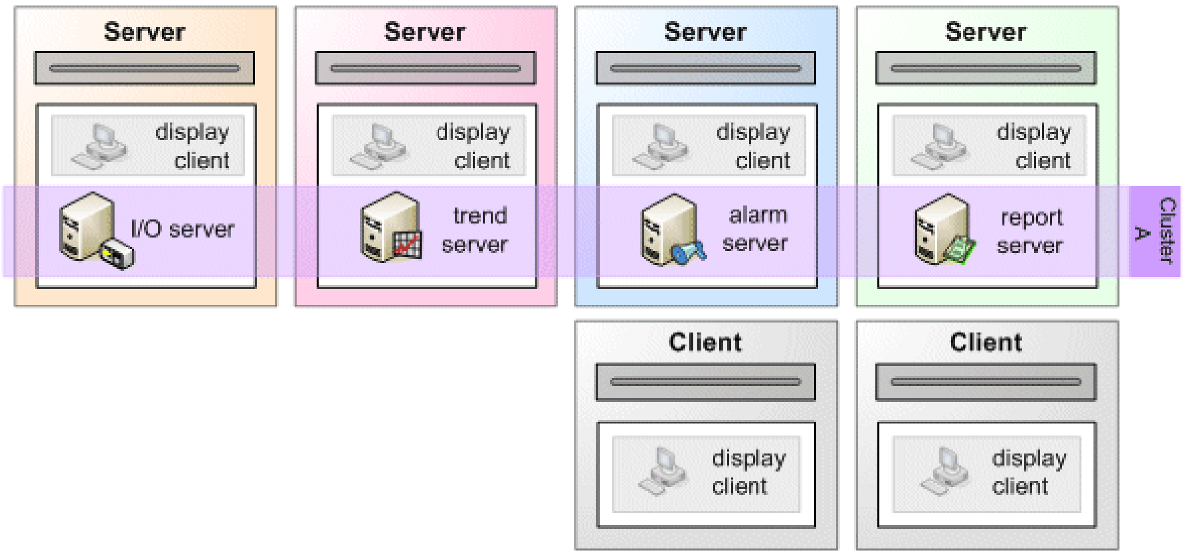
Client-Server System

Citect SCADA's client-server architecture allows the components of a system to be distributed across a number of computers on a LAN, creating a system that offers geographical flexibility and performance benefits.

Each component is simply identified within the project by an address, allowing the location and hardware requirements for each to be considered independently.



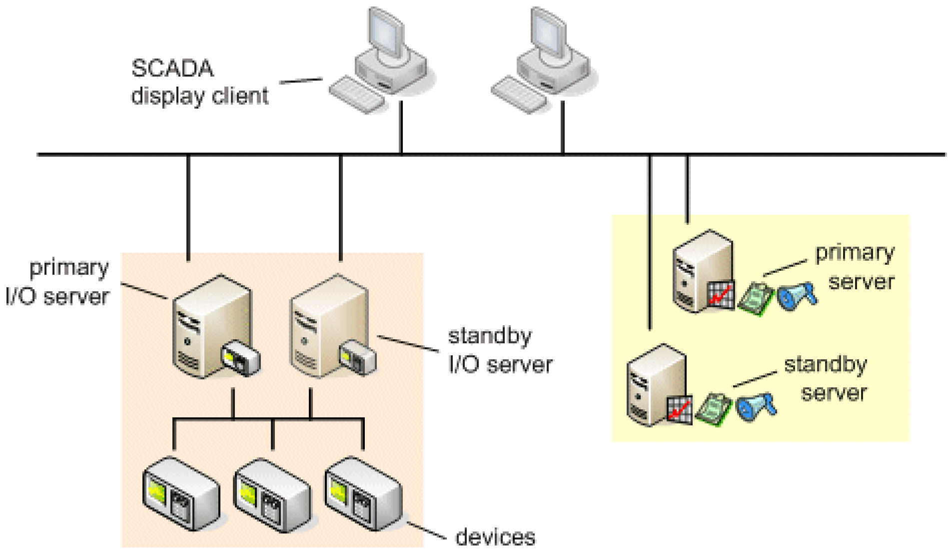
The diagram below demonstrates how this example can still be configured within a single cluster.



Each server also acts as a display client across the system architecture.

Redundant Server System

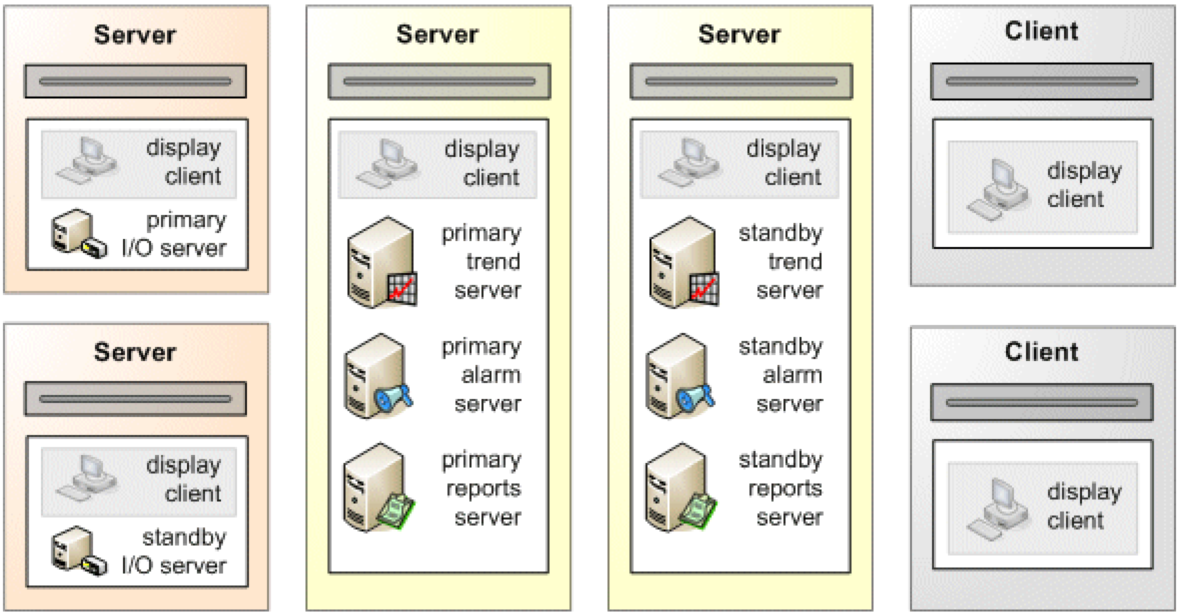
The ability to define primary and standby servers within a project allows hardware redundancy to be built into your system infrastructure. This helps prevent situations where an error on one server results in the overall system becoming inoperative. Systems of this type are especially beneficial when service continuity and/or secure data collection are important.



In the case of I/O server redundancy, a standby server is maintained in parallel to the primary server. If a hardware error is detected, the standby server can assume control of device communication with minimal interruption to the system. You can also use redundant I/O servers to split the processing load.

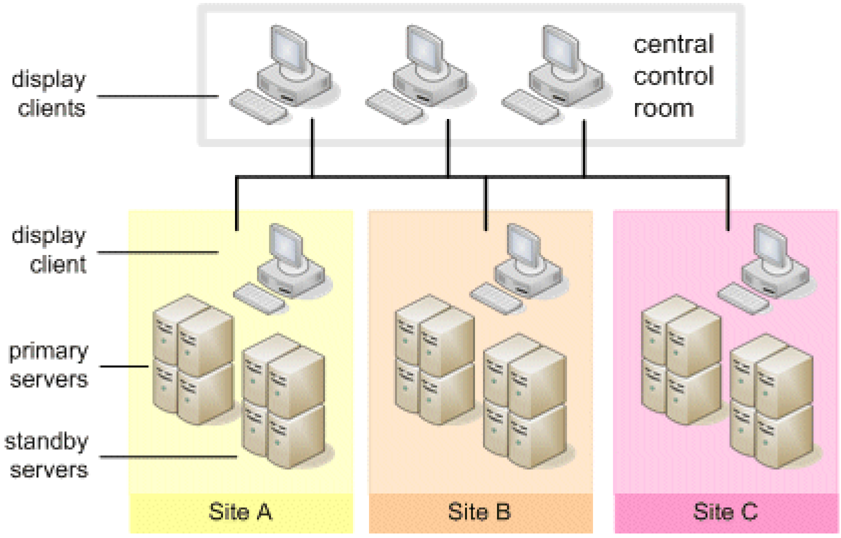
Alarm, report and trends servers can also be implemented as redundant servers. This improves the likelihood that clients will continue to have access to data from a standby server in the case a primary server becomes inoperative. Citect SCADA maintains identical data on both servers.

In the diagram below, the primary and standby I/O servers are deployed independently, while the alarms, trends and reports servers are run as separate processes on common primary and standby computers. In this case, the entire system can be configured as a single cluster.



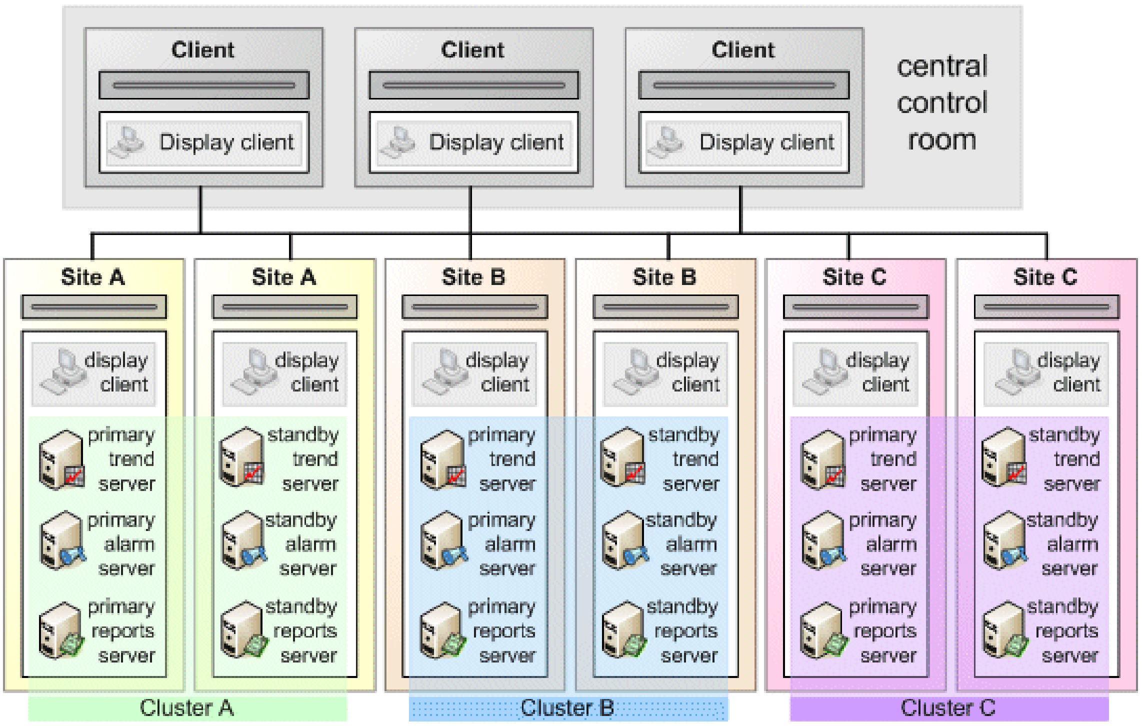
Clustered Control System

In this scenario, the system is organized into discrete sites being controlled by local operators, and supported by local redundant servers. At the same time, there is a level of management that requires sites across the system to be monitored simultaneously from a central control room.



Each site is represented in the project with a separate cluster, grouping its primary and standby servers. Clients at each site are only interested in the local cluster, whereas clients at the central control room are able to view every cluster.
The deployment of a control room scenario is fairly straightforward, as each site can be addressed independently within its own cluster. The control room itself only needs control clients.

The deployment of servers could be mapped out as follows:



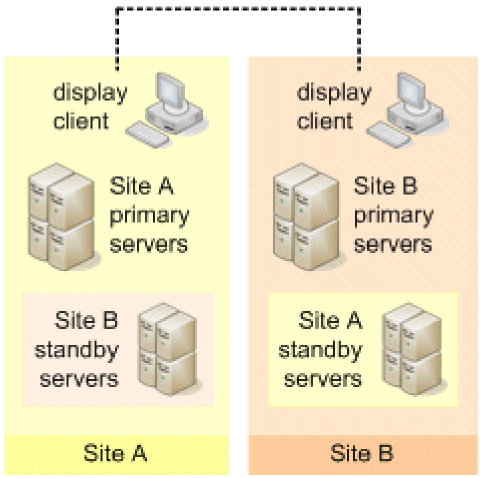
Citect SCADA's support for dynamic clustering means each site can be monitored and controlled from the central control room if necessary. For example, if an operator at a particular site only works during regular business hours, then the monitoring can be switched to the central control room after hours.

Redundant and Distributed Control System

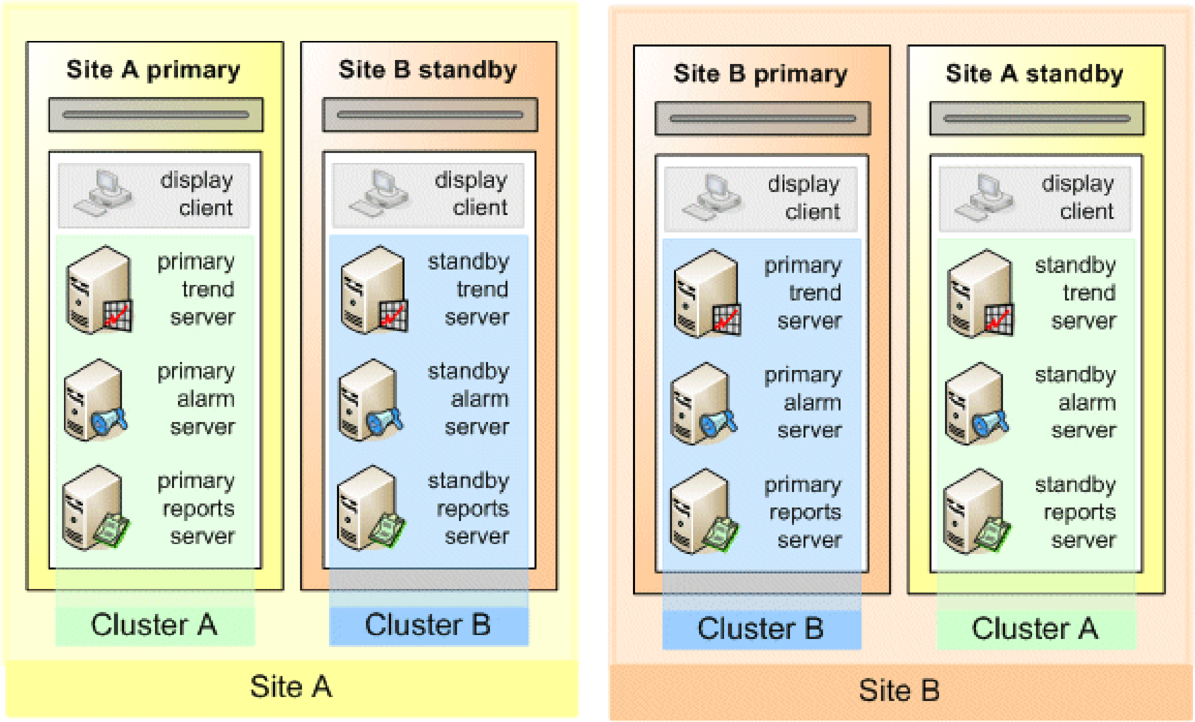
In this scenario, a project represents a number of locally operated sites each containing its own set of servers and clients. For example, a number of pumping stations across a water distribution system, or multiple production lines in a manufacturing facility. However, there is a requirement for monitoring to continue in the event the system at one of the sites becomes inoperative.

This is achieved by distributing the primary and standby servers across the different sites, or by placing the standby servers in a central location.

Clustering is used to define the role of the different servers at each site, which can be viewed in a common project running on every client. This means Site A can be monitored from Site B, and vice versa, if a system becomes inoperative at one of the sites.



The example above would require the creation of two clusters, so that the project can include two sets of primary and standby servers. The clusters represent the redundant pairs of servers, and would be deployed across the two sites as follows:



The clusters offer the benefit of keeping a logical structure to the project during configuration, despite the unusual distribution of redundant server pairs.

Load Sharing System

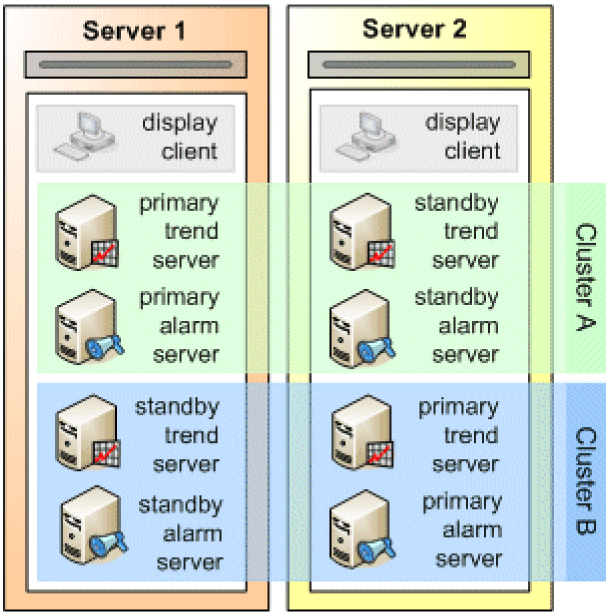
Load sharing of system components across different computers and CPUs means the work load of a potentially stressed system can be split across multiple machines, better utilizing the available infrastructure.

For example, managing alarms can draw heavily on a CPU's performance, while trending data can use a lot of disk space. By assigning your trends and alarm servers to different processes on a shared computer, an alarm server can be used as a standby trends server, making practical use of idle disk space.

This approach can be used to improve network performance, data access times, and general system stability.

If you introduce clustering, you have the flexibility to run multiple servers of the same type on a single computer. As long as a client has access to every cluster configured in a project, it doesn't matter if a set of servers is distributed across a number of clusters.

In the diagram below, two servers have been configured to act as standby units for each other, supporting two sets of redundant trends and alarm servers.



Both machines have an even balance of trends and alarm servers, making effective use of the CPU and disk space. By distributing the servers across two clusters, the servers are also able to act as redundant units to each other. This has reduced the necessary number of computers from a maximum of eight down to just two.