Table of Contents

Hyper Data Transfer for BioStar 2
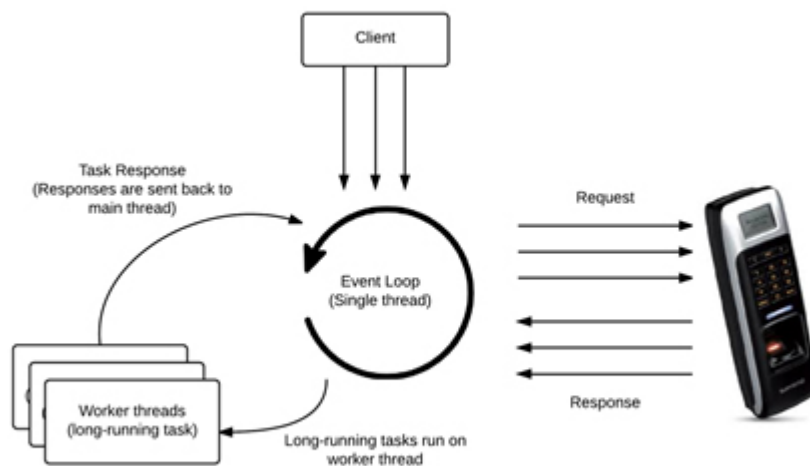
The data transfer of Input and Output (I/O) devices on a computer is relatively much slower compared to the data processing within CPU and memory due to a number of different factors. The available memory bandwidth and increasing CPU performance outpace that of I/O devices. As a result, handling the I/O operations perfectly is the key point for designing a system architecture.

For this reason, the BioStar 2 server adapted an asynchronous system architecture for handling network I/O operations. Before taking a deep look into the asynchronous system architecture, let's understand the synchronous system architecture first.

Synchronous (Blocking I/O) System Architecture

In a synchronous system architecture, the system conducts I/O operations in a sequential order.
If the server requests an I/O operation which is to be handled by a device, the server has to wait for the response from the device. If one of the devices does not respond to the I/O operation because of network problems, the other I/O operations will not continue.
This can be solved by spawning a new process or making a worker thread for each I/O operation. Spawning new processes consumes system resources of the system such as memory. Making a worker thread is more lightweight than spawning the process, but the more I/O operations are necessary the more threads need to be created. Because system resources are limited, the number of threads which a system can create is limited.[1]

Asynchronous (Non-blocking I/O) System Architecture

The asynchronous system architecture is extremely different from the synchronous system architecture.

In the asynchronous system architecture, I/O operations are handled by an event-driven system. If the server requests an I/O operation with callback, the system just returns and lets the processor continue to work. When the operation is completed, it pushes the complete event to the event-loop with the result of the operation and calls the callback function with its result. Thus, in the asynchronous system architecture only a single thread is required for I/O operations.

By adapting the asynchronous system architecture, BioStar 2 server does not need to wait for the response of the device. The server can send another request to a different device without collisions. In case that an I/O operation with a device has a problem such as a network disconnection, the I/O operations with the other devices will not be interrupted and can be handled normally.

This system architecture doesn't have large benefits for small-sized systems, however it exposes its advantages when a large number of devices or traffic is required.

[1)]

See C10K problem.

From:
http://kb.supremainc.com/knowledge/ -

Permanent link:
http://kb.supremainc.com/knowledge/doku.php?id=en:tc_technology_hyper_data_transfer

Last update: **2017/04/12 17:13**