Subject: Re: SCGI Class
Posted by mirek on Sun, 27 Mar 2011 08:02:18 GMT

Mindtraveller wrote on Sat, 26 March 2011 18:00zsolt wrote on Sat, 26 March 2011 00:55BTW it will not solve the problem of long running SQL queries or doing some slow communication in the scgi process.

In such situations you will have to start a lot of scgi processes to be able to handle the traffic.

Increasing that number, just allows the scgi process to have a large backlog.
What if in the main cycle the the newly created client socket is processed in another thread while server socket is free to accept more connections?

```
while (run)
{ if  (serverSocket.Accept(&clientSocket)
  {
    GetThreadFromPoolAndProcess(clientSocket);
  }
}
```

Actually, this can be quite nicely with something like

```
while(run)
{ if  (serverSocket.Accept(&clientSocket) {
    DoWork();
  }
}
```

and then simply starting several threads to run this loop (with single clientSocket). As accept is reentrant and MT safe, it would return only for single thread running, thus managing the thread pool.