```cpp
#include "ServerSocket.h"

// Structure comportant l'action du serveur à exécuter dans le nouveau thread
// et le socket associé.
typedef struct _thread_args {
    void (*action)(ClientSocket sock);
    int client_fd;
} thread_args;

void _thread_routine(void *v_args);

// Lance un serveur écoutant sur un port et gérant les connexions sur
// plusieurs threads en utilisant une pool de thread.
void with_server_socket(const int port, const int n_threads,
                        void (*action)(ClientSocket sock))
{
    int _socket = socket_utils::init_socket(INADDR_ANY, port, socket_utils::SERVER_SOCKET);
    ThreadPool pool(n_threads);

    socket_utils::listen(_socket, 0);

    for (;;) {
        // Lance la gestion du client dans un nouveau thread.
        int client_fd = socket_utils::accept(_socket, NULL, NULL);

        thread_args *args = new thread_args;
        args->action = action;
        args->client_fd = client_fd;

        pool.inject(_thread_routine, (void *) args);
    }

    shutdown(_socket, SHUT_RDWR);
    close(_socket);
}

// Appelle l'action de gestion du client depuis le nouveau thread
// (extrait l'object socket des arguments de lancement du thread).
void _thread_routine(void *v_args)
{
    thread_args *args = (thread_args*) v_args;
    ClientSocket sock(args->client_fd);

    args->action(sock);

    sock.close();
    delete args;
}
```