

Vaja 3

Postavitev in upravljanje računalniških oblakov

David Rubin

(david.rubin@student.um.si)

8. november 2018

1 Opis naloge

Napisati je bilo potrebno OpenMPI program, ki nad gručo 9 računalnikov izvaja podajanje sporočil. Vozlišče z rangom 0 pošlje vsem ostalim naključno realno število med 0 in 180, ti pa potem z isto številko odgovorijo. Izhodno vozlišče seštevava odgovore (števila) po modulu 360. Postopek se izvaja dokler ni vsota med 270.505 in 270.515. Na koncu vozlišče 0 ipiše število parov podaj v konzolo in v datoteko *RESULT.txt*.

2 Opis rešitve

Rešitev sem implementiral s pomočjo Vagrant datoteke in dvema skriptama. Vagrant datoteka ustvari podobno okolje kot pri 2. vaji (privatno omrežje, lahko se povezujejo med sabo, enoten ključ, 9 instanc, itd.), kar je sedaj novega je, da se namestita še paketa *openmpi-bin* in *libopenmpi-dev*, ter v */etc/hosts* datoteko se dodajo zapisi o IP naslovih instanc. Slednje olajša delo pri povezovanju med njimi.

Bistvo te rešitve je program *ping-pong*. Če na hitro opišemo program: preko *MPI_Bcast* vozlišče 0 pošlje svojo vrednost, ostali pa jo sprejmejo. V naslednjem koraku se na vozlišču 0 preko *MPI_Gather* sprejme vrednosti od ostalih vozlišč in se jih prišteje k vsoti (po modulu seveda), na ostalih vozliščih pa se preko *MPI_Gather* pošlje vrednosti vozlišču 0. Da se program na koncu ustavi vodimo evidenco o vsoti tudi na vozliščih ranga višje od 0. Vse to se izvaja dokler ni izpolnjen pogoj o vsoti, na koncu pa se izpiše še rezultat v konzolo in v datoteko. Program je prikazan v kodnem bloku 1.

Prevajanje programa: `mpicc -std=c99 -o ping-pong ping-pong.c -lm`

Zagon programa: `mpirun -mca btl_tcp_if_include eth1 -n 9 -hostfile mpi_hosts ping-pong`, pri zagonu se uporablja mca stikalo, da povemo kateri vmesnik se naj uporablja, prav tako imamo datoteko *mpi_hosts*, v kateri so pa zgolj naštet domenska imena vseh virtualk.

Listing 1: Program ping-pong

```
// @author David Rubin
#include <mpi.h>
#include <stdio.h>
```

```

#include <stdlib.h>
#include <math.h>
#include <time.h>

int main(int argc, char** argv) {
    // Osnovno vozlisce
    const int root = 0;
    // Za generiranje random števil
    srand(time(NULL));
    MPI_Init(NULL, NULL);
    // Število procesov
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);
    // Rank vozlišca (0 - število_procesov)
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    // Vsota števil, ki se podajajo (po modulu 360)
    float sum = 0.f;
    // Število podajanj (v enem ciklu se jih izvede World_size-1)
    int ping_count = 0;
    // Vrsta, ki jo posiljamo okoli
    float send_buf;

    // Delaj dokler ni vsota med podanima vrednostnima
    while (sum < 270.505 || sum > 270.515) {
        // Pridobi random float med 0 in 180
        if (world_rank == root) {
            send_buf = ((float)rand()/(float)(RAND_MAX) * 180);
        }
        // Poslji float na vsa ostala vozlišca (naredi root)
        // Sprejmi float iz vozlišca root (naredi rank > 0)
        MPI_Bcast(&send_buf, 1, MPI_FLOAT, root, MPI_COMM_WORLD);

        if (world_rank == root) {
            // Izvedlo se je world-size-1 ping pong
            ping_count += (world_size-1);
            float * recv_buf = malloc((world_size-1) * sizeof(float
                ↪ ));
            // Sprejmi vse vrednosti od ostalih (root)
            MPI_Gather(&send_buf, 1, MPI_FLOAT,
                recv_buf, 1, MPI_FLOAT,
                root, MPI_COMM_WORLD);

            // Sestoj vrednosti v novo vsoto
            for (int i = 0; i < world_size-1; ++i) {
                sum += recv_buf[i];
                sum = fmod(sum, 360);
            }
            //printf("Trenutna vsota=%f\n", sum);
        } else {
            // Poslji vrednost prejeto v broadcast nazaj na root
            MPI_Gather(&send_buf, 1, MPI_FLOAT,
                NULL, 0, MPI_FLOAT,
                root, MPI_COMM_WORLD);

            // Na slave vozlišcih tudi sestevaj (zaradi broadcast
                ↪ je tam *)
            // da lahko zaključijo in se program ustavi
            for (int i = 0; i < world_size-1; ++i) {
                sum += send_buf;
                sum = fmod(sum, 360);
            }
        }
    }
}

```

```

    }
}
// Na vozlišcu root se obvesti uporabnika o rezultatih
if (world_rank == root) {
    FILE *fp;
    fp = fopen("RESULT.txt", "w");
    fprintf(fp, "%d\n", ping_count);
    fclose(fp);
    // Vsota je med postavljenima mejama, izpisi na stdout in v
    ↪ datoteko
    printf("Vseh podajanj je bilo %d\n", ping_count);
    printf("Na koncu je sum enaka %f\n", sum);
}
// Zaključni program
MPI_Finalize();
return 0;
}

```

Pri delu sem uporabil tudi skripto imenovano *progam-copy.sh*, ki podani program prekopira na vse instance v gruči in ga na vsaki prevede.

Listing 2: Skripta za kopiranje programa

```

# @author David Rubin

file=$1
for ip in `cut -f1 ../cluster/trusty_hosts`; do
    scp -i ~/.ssh/ubuntu_id $file vagrant@$ip:~
    ssh -i ~/.ssh/ubuntu_id vagrant@$ip mpicc -std=c99 -o $(basename $file
    ↪ .c) $file -lm
done

```

Prikaz delovanja programa je prikazan na sliki 1.

3 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.

```
david@Ava: ~/Projects/PURO/3-mpi-ping-pong
Projects/PURO/3-mpi-ping-pong
→ ./program-copy.sh ping-pong.c && cd ../cluster && vagrant ssh trusty0
ping-pong.c 100% 2771 6.7MB/s 00:00
ping-pong.c 100% 2771 6.9MB/s 00:00
ping-pong.c 100% 2771 6.5MB/s 00:00
ping-pong.c 100% 2771 6.5MB/s 00:00
ping-pong.c 100% 2771 6.0MB/s 00:00
ping-pong.c 100% 2771 6.6MB/s 00:00
ping-pong.c 100% 2771 6.8MB/s 00:00
ping-pong.c 100% 2771 5.6MB/s 00:00
ping-pong.c 100% 2771 6.1MB/s 00:00
ping-pong.c 100% 2771 4.9MB/s 00:00
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-161-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Nov  8 21:00:38 UTC 2018

System load:  0.61           Processes:      75
Usage of /:   3.7% of 39.34GB Users logged in: 0
Memory usage: 37%           IP address for eth0: 10.0.2.15
Swap usage:   0%             IP address for eth1: 192.168.33.10

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Nov  8 20:57:41 2018 from 10.0.2.2
vagrant@trusty0:~$ mpirun -mca btl_tcp_if_include eth1 -n 9 -hostfile mpi_hosts ping-pong
Vseh podajanj je bilo 91448
Na koncu je sum enaka 270.505676
vagrant@trusty0:~$ cat RESULT.txt
91448
vagrant@trusty0:~$ exit 0
logout
Connection to 127.0.0.1 closed.
```

Slika 1: Delovanje programa pri 9 instancah