MPI tesztelése a nyelvi laboratóriumban (ELTE)

Hernyák Zoltán

2001.11.18

1. Kezdeti problémák

Az MPI futtató scriptje az /etc/mpich/machines.LINUX fileból olvassa ki azon gépek neveit, amelyeken a programokat futtatni akarom. Mivel minden gépen más és más az /etc alkönyvtár, így ezt minden gépen (amelyet a fejlesztés-futtatás szempontjából fő gépnek nevezhetünk) ezt a file-t fel kell tölteni. Vagyis ha az nyl01 gépen kezdtem el a tesztet, és ott ezt a file-t mar beállítottam, majd azt valaki kikapcsolta, akkor át kellett telepedni az nyl02-re, ahol ott újra be kellett állítani ezt a listát.

További kényelmetlenséget okozott, hogy a lista tartalmát is annak megfelelően kell feltölteni, hogy mely gépek vannak bekapcsolva. András tanácsait követve bejelentkezés után minden gépre meg kellett próbálni ssh-val bejelentkezni, és csak azon gépek neveit felvenni a listára, ahova aktuálisan sikerült bejelentkezni (működtek).

2. Teszt programok futtatásának eredményei

1. teszt: a rendszer működőképességének tesztje:
*** Hagyományos HELLO WORLD az összes processztől ...*
Működik.

2. teszt: stdio
*** Minden processz megpróbál bekérni egy érteket a billentyűzetről.*
A teszt szerint a 0-as rangú (fő) processznél működik csak a bekérés (valószínűleg azért, mert ő éppen ezen a gépen indul el, a többi távoli gépeken. A többi gép egyszerűen átlépi a scanf fv végrehajtását, és a változóba véletlen érték kerül. Így megállapítható, hogy a stdout-ra írás minden processznek működik, a kimenet a fő terminálon jelenik meg, de a stdin olvasása csak a 0-s processznél működik.

3. teszt: log file
*** A π értékének kiszámítása, közben „log” file készítése a* `MPE_Log_*` *eljárások segítségével.*
A program nem volt lefordítható, talán a telepített MPICH könyvtár nem teljes.

4. teszt: szeparált file io
*** Minden processz létrehoz egy egyedi file-t, ír bele, majd visszaolvassa, és az olvasás eredményét továbbküldi a többi taszk számára.*
A program nem fut le, a hibauzenet:

```
"File locking failed in ADIOI_Set_lock. If the file system is
NFS, you need to use NFS version 3 and mount the directory
with the 'noac' option (no attribute caching).
[0] MPI Abort by user Aborting program !"
```

A szöveg szerint ezen tesztkörnyezet nincs megfelelően konfigurálva a szeparált file i/o számára.

5. teszt: üzenetküldési sebesség
*** Adott byte-nyi üzenetet küld minden processz a másiknak, közben méri a sebességet.*
A tesztelés során maximum 64000 byte-nyi üzenetet küldtünk szét. Először 1 byte, majd kettő hatványaival növelt méretben. A teszt szerint minél több a küldendő byte-ok száma, annál jobb a sávkihasználtság. A sebességteszt persze ingadozó eredményt mutat, hiszen erősem függ a hálózat aktuális terhelésétől. A tesztelésbe a rendszer nagyobb méret esetén többször belefagyott (32k föloöt), de a Ctrl-C mindig működött :-))

6. teszt: Mekkora az MPI buffere.
*** A program sorban próbál egyre hosszabb üzeneteket küldeni, amíg le nem áll. Ezzel lehet meghatározni, mekkora az átvihető leghosszabb üzenet mérete, mekkora az MPI buffere.*
A teszt szerint a 65536 byte meg átvihető, a buffer mérete ezen implementációban 64k.

7. teszt: Blokkoló küldés-fogadás.
*** Minden processz elküld 100 db INT-t a root-nak, aki ezt összeszedi.*
A teszt rendben lefutott, az üzenetek megfelelő sorrendben érkeztek be a root-hoz (az MPI specifikációban ez elő van írva).

8. teszt: Nem blokkoló küldés-fogadás.
*** Minden processz elküld 100 db INT-t a root-nak, aki ezt összeszedi.*
A teszt lefutásakor kiderült, hogy a rangban kisebbek által küldött üzenetek "hamarabb" odaértek a root-hoz. Magyarázatot nem igazan találok rá, hiszen elvileg az `MPI_INIT()` pontban minden taszk szinkronizál (csak akkor lép tovább, ha minden taszk inicializálta az MPI környezetet), tehát az nem tekinthető magyarázatnak, hogy az 1. taszk "hamarabb" elindult, és hamarabb kezdet el küldözgetni az üzeneteit. De az üzenetkuldésben egyébként is megfigyelhető bizonyos szakaszosság, mert aztán egy kis "vegyes" szakasz után meg csupa 2. taszkbeli üzenet jött be.

9. teszt: Gyűrű-üzenetküldés
*** Minden processz küld egy INT-t a szomszédjának, aki azt továbbküldi A processzek "sorrendjét" az MPIRUN határozza meg (`process_rank`)*
A teszt a következővel párban állítva annak eldöntésében segítene, hogy az MPI által felállított sorrend nem optimális, amennyiben a taszkok nem a root-al, hanem egymással is kommunikálnak. Az MPI a sorrendet a konfigurációs fileban feltüntetett sorrendben osztja ki, amely alkalmasint nem jelent fizikailag, sávszelességileg is "szomszédot". Sajnos a teszt eredménye nem igazan kiértékelhető, hiszen a tesztlabor gépei fizikailag is szomszédok, ahol a 10. teszt eredménye nem ad mérhetően jobb vagy rosszabb eredményt.

10. teszt: Gyűrű üzenetküldés (CART)
*** Minden processz kuld egy INT-t a szomszédjának, aki azt továbbküldi. A "szomszéd"-t a `MPI_Cart_Create` hatarozza meg. (Cartesian Grid Create)*
Lásd előző értékelés.

11. teszt: Broadcast
*** A root mindenkinek küld egy INT-t*
A teszt rendben lefutott.

12. teszt: Broadcast + PACK
*** A root mindenkinek küld egy INT-t és egy DOUBLE-t, pack-olva.*
A teszt rendben lefutott.

13. teszt: Sebességteszt
*** A root egyre több darabb double-t küld a `#1` processznek, aki visszaküldi azt Max. 1 millio darab double-t küld át (minden esetben dupla annyit). Mindezt 10x ismétli meg.*
Valameddig eljutott, de valami gond van a nyelvi laborban (talán szünet, vagy óra vége), és a teszt közepén az egész kapcsolat kifagyott.

3. Összefoglalás

Az MPI lényegében működik. A küldő puffer mérete 64k ezen implementációban. A közel ekkora, vagy nagyobb méretű küldés eseten a működés bizonytalan. A képernyőre írás működik bármely processz esetén, de az olvasás csak a root-nak. A file-ba írás/olvasás nem működött, talán konfigurációs hibából fakadóan.

Nagyon kényelmetlenné teszi a használatot, hogy a konfigurációs file-t (ami a MPI számára hozzáférhető gépek listáját tartalmazza), mindig az alkalomhoz kellett igazítani. Az MPI nem kezeli azt, hogy ha egy gép nem hozzáférhető, akkor másikat választ a listából.

Ezen túlmenően a master/slave programoknál a futtatható kódban egyszerre kell szerepeljen a master es a slave kódja is, és a processz rangja határozhatja meg, hogy melyik kódrészlet kerüljön futtatásra. Ettől eltérni egy módon lehet: egy file-ban le kell írni a két eltérő futtatható program nevét, ebből az első lesz a master (rank=0), a második lesz a slave.

A processzek száma (és az őket futtató gépek) már indításkor eldőlnek, nincs lehetőség újabb processzek becsatlakoztatására a rendszerbe. Ezzel kapcsolatosan az MPI2 szabvány már ad uj fv-eket támogatásképpen, de konkrét implementációt még nem találtam vele kapcsolatban.

A programoknak MPI alatt be kell jelentkezni az MPI környezetbe (`MPI_INIT`). Ez egyben szinkronizációs pont is, a fv csak akkor tér vissza, ha minden processz elvégezte a bejelentkezést. Ha valamelyik ezt elmulasztja, akkor a rendszer itt leáll.

Hernyák Zoltán

Hivatkozások

[1] Peter S. Pacheco, *Parallel programming with MPI*, Morgan Kaufmann Publishers Inc., San Francisco, California, 1997

[2] Yukiya Aoyama, Jun Nakano, *RS/6000 SP: Practical MPI Programming*, International Technical Support Organization, http://www.redbooks.ibm.com

[3] William Gropp, Ewing Lusk, *User's Guide for mpich, a Portable Implementation of MPI, Version 1.2.1*, Argonne National Laboratory, University of Chichago, Mathematics and Computer Science Division,

[4] *MPI-2: Extensions to the Message-Passing Interface*, Message Passing Interface Forum, July 18, 1997, University of Tenesse, Knoxville, Tenessee

```
------------------------------------------------------------------- Melleklet: 5. teszt
Process 0 is alive on nyl02.nylab.inf.elte.hu
Process 1 is alive on nyl03.nylab.inf.elte.hu
Process 2 is alive on nyl04.nylab.inf.elte.hu
Process 3 is alive on nyl06.nylab.inf.elte.hu
Process 4 is alive on nyl08.nylab.inf.elte.hu

Ring test...time network performance
---------

Input maximum message size: 1200000
len=        1 bytes, used= 0.000325 sec., rate=0.006154 Mbytes/sec
len=        2 bytes, used= 0.000232 sec., rate=0.017241 Mbytes/sec
len=        4 bytes, used= 0.000227 sec., rate=0.035242 Mbytes/sec
len=        8 bytes, used= 0.281651 sec., rate=0.000057 Mbytes/sec
len=       16 bytes, used= 0.074103 sec., rate=0.000432 Mbytes/sec
len=       32 bytes, used= 0.000237 sec., rate=0.270042 Mbytes/sec
len=       64 bytes, used= 0.000249 sec., rate=0.514056 Mbytes/sec
len=      128 bytes, used= 0.000251 sec., rate=1.019920 Mbytes/sec
len=      256 bytes, used= 0.000266 sec., rate=1.924812 Mbytes/sec
len=      512 bytes, used= 0.000318 sec., rate=3.220126 Mbytes/sec
len=     1024 bytes, used= 0.000428 sec., rate=4.785047 Mbytes/sec
len=     2048 bytes, used= 0.000620 sec., rate=6.606452 Mbytes/sec
len=     4096 bytes, used= 0.001117 sec., rate=7.333930 Mbytes/sec
len=     8192 bytes, used= 0.001893 sec., rate=8.655045 Mbytes/sec
len=    16384 bytes, used= 0.003676 sec., rate=8.914037 Mbytes/sec
len=    32768 bytes, used= 0.006654 sec., rate=9.849113 Mbytes/sec
len=    65536 bytes, used= 0.014177 sec., rate=9.245397 Mbytes/sec
len=   131072 bytes, used= 0.034859 sec., rate=7.520124 Mbytes/sec
len=   262144 bytes, used= 0.070097 sec., rate=7.479464 Mbytes/sec
len=   524288 bytes, used= 0.133978 sec., rate=7.826479 Mbytes/sec
len=  1048576 bytes, used= 0.274663 sec., rate=7.635364 Mbytes/sec
clock resolution in seconds: 0.00000200
```

```
----------------------------------------------------------------- Melleklet: 8. teszt
Msg from 1 with tag 0
Msg from 1 with tag 1
Msg from 1 with tag 2
Msg from 1 with tag 3
Msg from 1 with tag 4
Msg from 1 with tag 5
Msg from 1 with tag 6
Msg from 1 with tag 7
Msg from 1 with tag 8
Msg from 1 with tag 9
Msg from 1 with tag 10
Msg from 1 with tag 11
Msg from 1 with tag 12
Msg from 1 with tag 13
Msg from 1 with tag 14
Msg from 1 with tag 15
Msg from 1 with tag 16
Msg from 1 with tag 17
Msg from 1 with tag 18
Msg from 1 with tag 19
Msg from 1 with tag 20
Msg from 1 with tag 21
Msg from 2 with tag 0
Msg from 1 with tag 22
Msg from 2 with tag 1
Msg from 1 with tag 23
Msg from 2 with tag 2
Msg from 1 with tag 24
Msg from 2 with tag 3
Msg from 1 with tag 25
Msg from 2 with tag 4
Msg from 1 with tag 26
Msg from 2 with tag 5
Msg from 1 with tag 27
Msg from 2 with tag 6
Msg from 1 with tag 28
Msg from 2 with tag 7
Msg from 1 with tag 29
Msg from 2 with tag 8
Msg from 1 with tag 30
Msg from 2 with tag 9
Msg from 1 with tag 31
Msg from 2 with tag 10
Msg from 1 with tag 32
Msg from 2 with tag 11
```

```
Msg from 1 with tag 33
Msg from 2 with tag 12
Msg from 1 with tag 34
Msg from 2 with tag 13
Msg from 1 with tag 35
Msg from 2 with tag 14
Msg from 1 with tag 36
Msg from 2 with tag 15
Msg from 1 with tag 37
Msg from 2 with tag 16
Msg from 1 with tag 38
Msg from 2 with tag 17
Msg from 1 with tag 39
Msg from 2 with tag 18
Msg from 1 with tag 40
Msg from 2 with tag 19
Msg from 2 with tag 20
Msg from 2 with tag 21
Msg from 2 with tag 22
Msg from 2 with tag 23
Msg from 2 with tag 24
Msg from 2 with tag 25
Msg from 2 with tag 26
Msg from 2 with tag 27
Msg from 2 with tag 28
Msg from 2 with tag 29
Msg from 2 with tag 30
Msg from 2 with tag 31
Msg from 2 with tag 32
Msg from 2 with tag 33
Msg from 2 with tag 34
Msg from 2 with tag 35
Msg from 2 with tag 36
Msg from 2 with tag 37
Msg from 2 with tag 38
Msg from 2 with tag 39
Msg from 2 with tag 40
Msg from 1 with tag 41
Msg from 1 with tag 42
Msg from 1 with tag 43
Msg from 1 with tag 44
Msg from 1 with tag 45
Msg from 1 with tag 46
Msg from 1 with tag 47
Msg from 1 with tag 48
Msg from 1 with tag 49
```

```
Msg from 1 with tag 50
Msg from 1 with tag 51
Msg from 1 with tag 52
Msg from 1 with tag 53
Msg from 1 with tag 54
Msg from 1 with tag 55
Msg from 1 with tag 56
Msg from 1 with tag 57
Msg from 1 with tag 58
Msg from 1 with tag 59
Msg from 1 with tag 60
Msg from 1 with tag 61
Msg from 1 with tag 62
Msg from 1 with tag 63
Msg from 1 with tag 64
Msg from 1 with tag 65
Msg from 1 with tag 66
Msg from 1 with tag 67
Msg from 2 with tag 41
Msg from 1 with tag 68
Msg from 2 with tag 42
Msg from 1 with tag 69
Msg from 2 with tag 43
Msg from 1 with tag 70
Msg from 2 with tag 44
Msg from 1 with tag 71
Msg from 2 with tag 45
Msg from 1 with tag 72
Msg from 2 with tag 46
Msg from 1 with tag 73
Msg from 2 with tag 47
Msg from 1 with tag 74
Msg from 2 with tag 48
Msg from 1 with tag 75
Msg from 2 with tag 49
Msg from 1 with tag 76
Msg from 2 with tag 50
Msg from 1 with tag 77
Msg from 2 with tag 51
Msg from 1 with tag 78
Msg from 2 with tag 52
Msg from 1 with tag 79
Msg from 2 with tag 53
Msg from 1 with tag 80
Msg from 2 with tag 54
Msg from 1 with tag 81
```

```
Msg from 2 with tag 55
Msg from 1 with tag 82
Msg from 2 with tag 56
Msg from 1 with tag 83
Msg from 2 with tag 57
Msg from 1 with tag 84
Msg from 2 with tag 58
Msg from 1 with tag 85
Msg from 2 with tag 59
Msg from 1 with tag 86
Msg from 2 with tag 60
Msg from 1 with tag 87
Msg from 2 with tag 61
Msg from 1 with tag 88
Msg from 2 with tag 62
Msg from 1 with tag 89
Msg from 2 with tag 63
Msg from 1 with tag 90
Msg from 2 with tag 64
Msg from 1 with tag 91
Msg from 2 with tag 65
Msg from 1 with tag 92
Msg from 2 with tag 66
Msg from 1 with tag 93
Msg from 1 with tag 94
Msg from 1 with tag 95
Msg from 1 with tag 96
Msg from 1 with tag 97
Msg from 2 with tag 67
Msg from 1 with tag 98
Msg from 2 with tag 68
Msg from 1 with tag 99
Msg from 2 with tag 69
Msg from 2 with tag 70
Msg from 2 with tag 71
Msg from 2 with tag 72
Msg from 2 with tag 73
Msg from 2 with tag 74
Msg from 2 with tag 75
Msg from 2 with tag 76
Msg from 2 with tag 77
Msg from 2 with tag 78
Msg from 2 with tag 79
Msg from 2 with tag 80
Msg from 2 with tag 81
Msg from 2 with tag 82
```

```
Msg from 2 with tag 83
Msg from 2 with tag 84
Msg from 2 with tag 85
Msg from 2 with tag 86
Msg from 2 with tag 87
Msg from 2 with tag 88
Msg from 2 with tag 89
Msg from 2 with tag 90
Msg from 2 with tag 91
Msg from 2 with tag 92
Msg from 2 with tag 93
Msg from 2 with tag 94
Msg from 2 with tag 95
Msg from 2 with tag 96
Msg from 2 with tag 97
Msg from 2 with tag 98
Msg from 2 with tag 99
```