[MPJ: Bug Report]

03/17/2006

Etsuko Sano

This is a bug report of MPJ, a part of the AgentTeamwork project. Some Java Grande Forum test programs were tested on Java sockets using mpjrun and resulted in some errors. All test programs can be found under the directory (~uwagent/MA/MPJ.new/JGF/).

For the test result, refer to the table below.

	2	4	8	16	32
PingPong	○	-	-	-	-
JGFPingPong	× (529010<)	-	-	-	-
JGFReduceBench	○	○	○	○	○
JGFAlltoallBench	○	× (288539<)	× (288539<)	× (154991<)	× (83255<)
JGFGatherBench	× (1626361<)	× (927577<)	× (529010<)	× (301708<)	× (172072<)
JGFBarrierBench	○	○	○	○	○

○ -> Passed more than 5 times

△ -> Failed a few times out of 5 times or more

× -> None succeeded(Array Size passed)

Table1: test result of MPJ-S using mpjrun

1. JGFPingPongBench.java *PingPong can be only run on 2 processors

   JGFPingPongBench class causes OutOfMemory errors when the array size becomes **529010.**

```
[etsuko71@medusa JGF]$ java -classpath MPJ.jar:. mpjrun JGFPingPongBench -port 20000 -np 2
-machinefile hostfile -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
Executing: java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFPingPongBench medusa.bothell.washington.edu 20000 -np 2
```

```
Executing: ssh mnode0 java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFPingPongBench medusa.bothell.washington.edu 20000 -amslave -yourname mnode0 -yourrank 1 -np 2
stdout 1> Slave adding master to comm
stdout 1> Slave completed comm entry
stdout 0> Master accepted connection from mnode0
stdout 1> Slave sending rank
stdout 0> Master trying to read slave rank
stdout 0> Master got slave 1
stdout 1> slave init complete
stdout 0> Java Grande Forum MPJ Benchmark Suite - Version 1.0 - Section 1
stdout 0> Executing on 2 processes
stdout 0>
remote process 1 forked.
stdout 0> Section1:PingPong:Double                  68169.03       (bytes/s)      Array Size = 4
stdout 0> Section1:PingPong:Double                  111024.2       (bytes/s)      Array Size = 7
stdout 0> Section1:PingPong:Double                  187223.42      (bytes/s)      Array Size = 12
stdout 0> Section1:PingPong:Double                  313551.53      (bytes/s)      Array Size = 21
stdout 0> Section1:PingPong:Double                  380216.7       (bytes/s)      Array Size = 37
stdout 0> Section1:PingPong:Double                  757575.25      (bytes/s)      Array Size = 66
stdout 0> Section1:PingPong:Double                   989480.1       (bytes/s)      Array Size =
116
stdout 0> Section1:PingPong:Double                   1175975.2      (bytes/s)      Array Size =
203
stdout 0> Section1:PingPong:Double                   1778649.2      (bytes/s)      Array Size =
357
stdout 0> Section1:PingPong:Double                   1628385.1      (bytes/s)      Array Size =
626
stdout 0> Section1:PingPong:Double                   845218.56      (bytes/s)      Array Size =
1098
stdout 0> Section1:PingPong:Double                   2008502.6      (bytes/s)      Array Size =
1926
stdout 0> Section1:PingPong:Double                   1339031.1      (bytes/s)      Array Size =
3377
stdout 0> Section1:PingPong:Double                   2055812.1      (bytes/s)      Array Size =
5921
stdout 0> Section1:PingPong:Double                   2874342.2      (bytes/s)      Array Size =
```

```
10383
stdout 0> Section1:PingPong:Double                    4105697.5       (bytes/s)      Array Size =
18205
stdout 0> Section1:PingPong:Double                    4607062.0       (bytes/s)      Array Size =
31921
stdout 0> Section1:PingPong:Double                    4456709.0       (bytes/s)      Array Size =
55970
stdout 0> Section1:PingPong:Double                    4119211.8       (bytes/s)      Array Size =
98137
stdout 0> Section1:PingPong:Double                    4227488.5       (bytes/s)      Array Size =
172072
stdout 0> Section1:PingPong:Double                    5591230.0       (bytes/s)      Array Size =
301708
stdout 0> Section1:PingPong:Double                    5595214.0       (bytes/s)      Array Size =
529010
stdout 0> Section1:PingPong:Double                    5627953.0       (bytes/s)      Array Size =
927557
stdout 0> Section1:PingPong:Double                    5695912.5       (bytes/s)      Array Size =
1626361
stdout 0> Section1:PingPong:Double                    5668544.0       (bytes/s)      Array Size =
2851632
stdout 0> Section1:PingPong:Object                   5772.5713       (objects/s)    Array Size = 4
stdout 0> Section1:PingPong:Object                    8898.82  (objects/s)    Array Size = 7
stdout 0> Section1:PingPong:Object                   12007.329       (objects/s)    Array Size = 12
stdout 0> Section1:PingPong:Object                   20643.428       (objects/s)    Array Size = 21
stdout 0> Section1:PingPong:Object                   12017.445       (objects/s)    Array Size = 37
stdout 0> Section1:PingPong:Object                   26757.992       (objects/s)    Array Size = 66
stdout 0> Section1:PingPong:Object                    38641.508       (objects/s)    Array Size =
116
stdout 0> Section1:PingPong:Object                    44562.3  (objects/s)    Array Size = 203
stdout 0> Section1:PingPong:Object                    28174.797       (objects/s)    Array Size =
357
stdout 0> Section1:PingPong:Object                    49889.02        (objects/s)    Array Size =
626
stdout 0> Section1:PingPong:Object                    45524.01        (objects/s)    Array Size =
1098
```

```
stdout 0> Section1:PingPong:Object                    47128.273       (objects/s)     Array Size =
1926
stdout 0> Section1:PingPong:Object                    45431.29        (objects/s)     Array Size =
3377
stdout 0> Section1:PingPong:Object                    43737.766       (objects/s)     Array Size =
5921
stdout 0> Section1:PingPong:Object                    29486.688       (objects/s)     Array Size =
10383
stdout 0> Section1:PingPong:Object                    20429.232       (objects/s)     Array Size =
18205
stdout 0> Section1:PingPong:Object                    38049.316       (objects/s)     Array Size =
31921
stdout 0> Section1:PingPong:Object                    34624.188       (objects/s)     Array Size =
55970
stdout 0> Section1:PingPong:Object                    36577.34        (objects/s)     Array Size =
98137
stdout 0> Section1:PingPong:Object                    33889.117       (objects/s)     Array Size =
172072
stdout 0> Section1:PingPong:Object                   28893.7  (objects/s)     Array Size = 301708
stdout 0> Section1:PingPong:Object                    35823.797       (objects/s)     Array Size =
529010
stderr 0> Exception in thread "main" java.lang.OutOfMemoryError
stderr 1> Exception in thread "main" java.lang.OutOfMemoryError
```

2. JGFAlltoallBench

This program only succeeds on two processors, but fails after that. The more number of processors increases, the more array size that error occurs decreases. This program might hang while executing on 32 processors.

```
[etsuko71@medusa myMPJ]$ java -classpath MPJ.jar:. mpjrun JGFAlltoallBench -port 40000 -np 4
-machinefile hostfile -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar

Executing:    java    -cp    /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFAlltoallBench medusa.bothell.washington.edu 40000 -np 4
Executing: ssh mnode0 java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
```

```
JGFAlltoallBench medusa.bothell.washington.edu 40000 -amslave -yourname mnode0 -yourrank 1 -np 4
stdout 0> Master accepted connection from mnode0
stdout 1> Slave adding master to comm
stdout 1> Slave completed comm entry
stdout 1> Slave sending rank
stdout 0> Master trying to read slave rank
stdout 0> Master got slave 1
remote process 1 forked.
Executing: ssh mnode1 java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFAlltoallBench medusa.bothell.washington.edu 40000 -amslave -yourname mnode1 -yourrank 2 -np 4
stdout 0> Master accepted connection from mnode1
stdout 0> Master trying to read slave rank
stdout 0> Master got slave 2
stdout 2> Slave adding master to comm
stdout 2> Slave completed comm entry
stdout 2> Slave sending rank
remote process 2 forked.
Executing: ssh mnode2 java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFAlltoallBench medusa.bothell.washington.edu 40000 -amslave -yourname mnode2 -yourrank 3 -np 4
stdout 0> Master accepted connection from mnode2
stdout 0> Master trying to read slave rank
stdout 3> Slave adding master to comm
stdout 3> Slave completed comm entry
stdout 0> Master got slave 3
stdout 3> Slave sending rank
stdout 1> ServerSocket(40001)
stdout 2> ServerSocket(40002)
stdout 2> Connected to mnode0 at port 40001 should be mnode0
stdout 3> Connected to mnode0 at port 40001 should be mnode0
stdout 1> Received connection from mnode2 rank 3
stdout 2> Received connection from mnode2 rank 3
stdout 2> slave init complete
stdout 1> Received connection from mnode1 rank 2
stdout 1> slave init complete
stdout 3> Connected to mnode1 at port 40002 should be mnode1
stdout 3> slave init complete
```

```
stdout 0> Java Grande Forum MPJ Benchmark Suite - Version 1.0 - Section 1
stdout 0> Executing on 4 processes
stdout 0>
remote process 3 forked.
stdout 0> Section1:Alltoall:Double                  125367.766      (bytes/s)      Array Size = 4
stdout 0> Section1:Alltoall:Double                  193443.81       (bytes/s)      Array Size = 7
stdout 0> Section1:Alltoall:Double                  259667.17       (bytes/s)      Array Size = 13
stdout 0> Section1:Alltoall:Double                  420264.22       (bytes/s)      Array Size = 25
stdout 0> Section1:Alltoall:Double                  745698.25       (bytes/s)      Array Size = 48
stdout 0> Section1:Alltoall:Double                  1126330.8       (bytes/s)      Array Size = 89
stdout 0> Section1:Alltoall:Double                   1363280.2       (bytes/s)      Array Size =
166
stdout 0> Section1:Alltoall:Double                   1377312.4       (bytes/s)      Array Size =
309
stdout 0> Section1:Alltoall:Double                   1013570.06      (bytes/s)      Array Size =
577
stdout 0> Section1:Alltoall:Double                   1703758.4       (bytes/s)      Array Size =
1074
stdout 0> Section1:Alltoall:Double                   1732981.4       (bytes/s)      Array Size =
1999
stdout 0> Section1:Alltoall:Double                   1765284.2       (bytes/s)      Array Size =
3723
stdout 0> Section1:Alltoall:Double                   1694584.6       (bytes/s)      Array Size =
6931
stdout 0> Section1:Alltoall:Double                   1664588.6       (bytes/s)      Array Size =
12903
stdout 0> Section1:Alltoall:Double                   1703971.1       (bytes/s)      Array Size =
24022
stdout 0> Section1:Alltoall:Double                   1584468.4       (bytes/s)      Array Size =
44721
stdout 0> Section1:Alltoall:Double                   1603828.8       (bytes/s)      Array Size =
83255
stdout 0> Section1:Alltoall:Double                   1555134.2       (bytes/s)      Array Size =
154991
stdout 0> Section1:Alltoall:Double                   1539644.5       (bytes/s)      Array Size =
288539
```

```
stdout 0> Section1:Alltoall:Double                    1519410.2       (bytes/s)      Array Size =
537159
stdout 0> time 0.241 size 1
stdout 0> time 0.071 size 2
stdout 0> time 0.044 size 4
stdout 0> time 0.021 size 8
stdout 0> time 0.046 size 16
stdout 0> time 0.164 size 32
stdout 0> time 0.261 size 64
stdout 0> time 0.297 size 128
stdout 0> time 0.509 size 256
stdout 0> time 0.568 size 512
stdout 0> time 1.206 size 1024
stdout 0> time 2.18 size 2048
stdout 0> time 4.366 size 4096
stdout 0> time 8.971 size 8192
stdout 0> time 17.567 size 16384
stdout 0> Section1:Alltoall:Object                    7142.888        (objects/s)    Array Size = 4
stdout 0> time 0.08 size 1
stdout 0> time 0.042 size 2
stdout 0> time 0.04 size 4
stdout 0> time 0.012 size 8
stdout 0> time 0.024 size 16
stdout 0> time 0.05 size 32
stdout 0> time 0.098 size 64
stdout 0> time 0.182 size 128
stdout 0> time 0.34 size 256
stdout 0> time 0.672 size 512
stdout 0> time 1.445 size 1024
stdout 0> time 2.658 size 2048
stdout 0> time 5.098 size 4096
stdout 0> time 10.182 size 8192
stdout 0> Section1:Alltoall:Object                    9550.966        (objects/s)    Array Size = 7
stdout 0> time 0.075 size 1
stdout 0> time 0.041 size 2
stdout 0> time 0.043 size 4
```

```
stdout 0> time 0.014 size 8
stdout 0> time 0.027 size 16
stdout 0> time 0.054 size 32
stdout 0> time 0.11 size 64
stdout 0> time 0.204 size 128
stdout 0> time 0.441 size 256
stdout 0> time 0.803 size 512
stdout 0> time 1.56 size 1024
stdout 0> time 3.088 size 2048
stdout 0> time 6.173 size 4096
stdout 0> time 12.419 size 8192
stdout 0> Section1:Alltoall:Object                    12880.503       (objects/s)     Array Size = 13
stdout 0> time 0.077 size 1
stdout 0> time 0.043 size 2
stdout 0> time 0.048 size 4
stdout 0> time 0.016 size 8
stdout 0> time 0.032 size 16
stdout 0> time 0.065 size 32
stdout 0> time 0.13 size 64
stdout 0> time 0.252 size 128
stdout 0> time 0.516 size 256
stdout 0> time 1.024 size 512
stdout 0> time 2.084 size 1024
stdout 0> time 4.078 size 2048
stdout 0> time 7.829 size 4096
stdout 0> time 15.984 size 8192
stdout 0> Section1:Alltoall:Object                    17930.31        (objects/s)     Array Size = 25
stdout 0> time 0.073 size 1
stdout 0> time 0.044 size 2
stdout 0> time 0.051 size 4
stdout 0> time 0.024 size 8
stdout 0> time 0.05 size 16
stdout 0> time 0.085 size 32
stdout 0> time 0.178 size 64
stdout 0> time 0.336 size 128
stdout 0> time 0.709 size 256
```

```
stdout 0> time 1.384 size 512
stdout 0> time 2.744 size 1024
stdout 0> time 5.495 size 2048
stdout 0> time 10.942 size 4096
stdout 0> Section1:Alltoall:Object                    22925.373       (objects/s)     Array Size = 48
stdout 0> time 0.075 size 1
stdout 0> time 0.047 size 2
stdout 0> time 0.053 size 4
stdout 0> time 0.035 size 8
stdout 0> time 0.072 size 16
stdout 0> time 0.142 size 32
stdout 0> time 0.273 size 64
stdout 0> time 0.586 size 128
stdout 0> time 1.134 size 256
stdout 0> time 2.223 size 512
stdout 0> time 4.38 size 1024
stdout 0> time 8.728 size 2048
stdout 0> time 17.202 size 4096
stdout 0> Section1:Alltoall:Object                    24075.023       (objects/s)     Array Size = 89
stdout 0> time 0.042 size 1
stdout 0> time 0.048 size 2
stdout 0> time 0.028 size 4
stdout 0> time 0.056 size 8
stdout 0> time 0.113 size 16
stdout 0> time 0.223 size 32
stdout 0> time 0.479 size 64
stdout 0> time 0.921 size 128
stdout 0> time 1.81 size 256
stdout 0> time 3.613 size 512
stdout 0> time 7.269 size 1024
stdout 0> time 14.483 size 2048
stdout 0> Section1:Alltoall:Object                     25267.04        (objects/s)     Array Size =
166
stdout 0> time 0.048 size 1
stdout 0> time 0.061 size 2
stdout 0> time 0.074 size 4
```

```
stdout 0> time 0.096 size 8
stdout 0> time 0.195 size 16
stdout 0> time 0.393 size 32
stdout 0> time 0.812 size 64
stdout 0> time 1.625 size 128
stdout 0> time 3.205 size 256
stdout 0> time 6.316 size 512
stdout 0> time 12.602 size 1024
stdout 0> Section1:Alltoall:Object                  26176.043       (objects/s)     Array Size =
309
stdout 0> time 0.024 size 1
stdout 0> time 0.063 size 2
stdout 0> time 0.087 size 4
stdout 0> time 0.175 size 8
stdout 0> time 0.348 size 16
stdout 0> time 0.726 size 32
stdout 0> time 1.447 size 64
stdout 0> time 2.85 size 128
stdout 0> time 5.58 size 256
stdout 0> time 11.126 size 512
stdout 0> Section1:Alltoall:Object                  27177.92        (objects/s)     Array Size =
577
stdout 0> time 0.044 size 1
stdout 0> time 0.092 size 2
stdout 0> time 0.16 size 4
stdout 0> time 0.325 size 8
stdout 0> time 0.652 size 16
stdout 0> time 1.323 size 32
stdout 0> time 2.614 size 64
stdout 0> time 5.261 size 128
stdout 0> time 10.486 size 256
stdout 0> Section1:Alltoall:Object                  26544.121       (objects/s)     Array Size =
1074
stdout 0> time 0.082 size 1
stdout 0> time 0.151 size 2
stdout 0> time 0.289 size 4
```

```
stdout 0> time 0.576 size 8
stdout 0> time 1.148 size 16
stdout 0> time 2.305 size 32
stdout 0> time 4.553 size 64
stdout 0> time 9.136 size 128
stdout 0> time 18.398 size 256
stdout 0> Section1:Alltoall:Object                  28011.604       (objects/s)    Array Size =
1999
stdout 0> time 0.143 size 1
stdout 0> time 0.28 size 2
stdout 0> time 0.548 size 4
stdout 0> time 1.051 size 8
stdout 0> time 2.097 size 16
stdout 0> time 4.241 size 32
stdout 0> time 8.467 size 64
stdout 0> time 16.968 size 128
stdout 0> Section1:Alltoall:Object                  28192.865       (objects/s)    Array Size =
3723
stdout 0> time 0.316 size 1
stdout 0> time 0.587 size 2
stdout 0> time 1.135 size 4
stdout 0> time 2.31 size 8
stdout 0> time 4.61 size 16
stdout 0> time 9.268 size 32
stdout 0> time 18.508 size 64
stdout 0> Section1:Alltoall:Object                  24008.66        (objects/s)    Array Size =
6931
stdout 0> time 0.646 size 1
stdout 0> time 1.266 size 2
stdout 0> time 2.386 size 4
stdout 0> time 4.714 size 8
stdout 0> time 9.487 size 16
stdout 0> time 18.824 size 32
stdout 0> Section1:Alltoall:Object                  21953.21        (objects/s)    Array Size =
12903
stdout 0> time 1.258 size 1
```

```
stdout 0> time 2.436 size 2
stdout 0> time 4.666 size 4
stdout 0> time 9.46 size 8
stdout 0> time 19.652 size 16
stdout 0> Section1:Alltoall:Object                    19565.873       (objects/s)     Array Size =
24022
stdout 0> time 2.751 size 1
stdout 0> time 5.267 size 2
stdout 0> time 10.44 size 4
stdout 0> Section1:Alltoall:Object                    17137.766       (objects/s)     Array Size =
44721
stdout 0> time 5.846 size 1
stdout 0> time 10.536 size 2
stdout 0> Section1:Alltoall:Object                    15803.91        (objects/s)     Array Size =
83255
stdout 0> time 11.181 size 1
stdout 0> Section1:Alltoall:Object                    13861.998       (objects/s)     Array Size =
154991
stdout 0> time 21.267 size 1
stdout 0> Section1:Alltoall:Object                    13568.09        (objects/s)     Array Size =
288539
stderr 0> Exception in thread "main" java.lang.OutOfMemoryError
stderr 2> Exception in thread "main" java.lang.OutOfMemoryError
stderr 1> Exception in thread "main" java.lang.OutOfMemoryError
stderr 3> Exception in thread "main" java.lang.OutOfMemoryError
```

3. JGFGatherBench

This program fails regardless of the number of processors to execute. The more number of processors increases, the more array size that error occurs decreases as well as the result of JGFAlltoallBench.

```
[etsuko71@medusa JGF]$ java -classpath MPJ.jar:. mpjrun JGFGatherBench -port 20005 -np 2 -machinefile
hostfile -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar

Executing:    java    -cp    /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
```

```
JGFGatherBench medusa.bothell.washington.edu 20005 -np 2
Executing: ssh mnode0 java -cp /home/etsuko71/myMPJ/MPJ.jar:/home/uwagent/MA/GridTcp/GridTcp.jar
JGFGatherBench medusa.bothell.washington.edu 20005 -amslave -yourname mnode0 -yourrank 1 -np 2
stdout 1> Slave adding master to comm
stdout 1> Slave completed comm entry
stdout 1> Slave sending rank
stdout 0> Master accepted connection from mnode0
stdout 0> Master trying to read slave rank
stdout 0> Master got slave 1
stdout 0> Java Grande Forum MPJ Benchmark Suite - Version 1.0 - Section 1
stdout 0> Executing on 2 processes
stdout 0>
stdout 1> slave init complete
remote process 1 forked.
stdout 0> Section1:Gather:Double                    -472437.94      (bytes/s)      Array Size = 4
stdout 0> Section1:Gather:Double                     761730.2       (bytes/s)      Array Size = 7
stdout 0> Section1:Gather:Double                    2650150.0       (bytes/s)      Array Size = 12
stdout 0> Section1:Gather:Double                    4411077.0       (bytes/s)      Array Size = 21
stdout 0> Section1:Gather:Double                    1.1728329E7     (bytes/s)      Array Size = 37
stdout 0> Section1:Gather:Double                    4213712.5       (bytes/s)      Array Size = 66
stdout 0> Section1:Gather:Double                    3815874.5       (bytes/s)      Array Size = 116
stdout 0> Section1:Gather:Double                    3505614.8       (bytes/s)      Array Size = 203
stdout 0> Section1:Gather:Double                    6919950.5       (bytes/s)      Array Size = 357
stdout 0> Section1:Gather:Double                    6393258.0       (bytes/s)      Array Size = 626
stdout 0> Section1:Gather:Double                     6677666.0       (bytes/s)      Array Size =
1098
stdout 0> Section1:Gather:Double                     6177074.5       (bytes/s)      Array Size =
1926
stdout 0> Section1:Gather:Double                     6262452.5       (bytes/s)      Array Size =
3377
stdout 0> Section1:Gather:Double                     5988251.0       (bytes/s)      Array Size =
5921
stdout 0> Section1:Gather:Double                     5479099.0       (bytes/s)      Array Size =
10383
stdout 0> Section1:Gather:Double                     4830764.5       (bytes/s)      Array Size =
18205
```

```
stdout 0> Section1:Gather:Double                    6572253.5       (bytes/s)       Array Size =
31921
stdout 0> Section1:Gather:Double                    4982030.5       (bytes/s)       Array Size =
55970
stdout 0> Section1:Gather:Double                    4489469.5       (bytes/s)       Array Size =
98137
stdout 0> Section1:Gather:Double                    3557905.8       (bytes/s)       Array Size =
172072
stdout 0> Section1:Gather:Double                    3622590.2       (bytes/s)       Array Size =
301708
stdout 0> Section1:Gather:Double                    3371168.0       (bytes/s)       Array Size =
529010
stdout 0> Section1:Gather:Double                    3236487.2       (bytes/s)       Array Size =
927557
stdout 0> Section1:Gather:Double                    3094883.0       (bytes/s)       Array Size =
1626361
stderr 0> Exception in thread "main" java.lang.OutOfMemoryError
stderr 1> Exception in thread "main" java.lang.OutOfMemoryError
```