Lab 2: MPI Collective Operations

EEL 6763

========== Initial Matrix ==========

215 100 200 204 233 50 85 196 71 141 122 160 93 131 243

234 162 183 36 155   4 62 35 205 40 102 33 27 255 55

131 214 156 75 163 134 126 249 74 197 134 197 102 228 72

 90 206 235 17 243 134 22 49 169 227 89 16   5 117 16

 60 248 230 217 68 138 96 194 131 170 136 10 112 238 238

184 72 189 163 90 176 42 112 225 212 84 58 228 89 175

244 150 168 219 112 236 101 208 175 233 123 55 243 235 37

225 164 110 158 71 201 78 114 57 48 70 142 106 43 232

 26 32 126 194 252 239 175 98 191 94 75 59 149 62 39

187 32 203 42 190 19 243 13 133 45 61 204 187 168 247

163 194 23 34 133 20 17 52 118 209 146 193 13 40 255

 52 227 32 255 13 222 18   1 236 152 46 41 100 233 209

 91 141 148 115 175 25 135 193 77 254 147 224 191 161   9

191 213 236 223 212 250 190 231 251 170 127 41 212 227 19

166 63 161 58 179 81 84 59 18 162 57 166 130 248 71

==========

sendcounts (# rows): [5 6 6 4]

displacements: [0 45 105 165]

Rank 0 submatrix:

========== Matrix ==========

215 100 200 204 233 50 85 196 71 141 122 160 93 131 243

234 162 183 36 155   4 62 35 205 40 102 33 27 255 55

131 214 156 75 163 134 126 249 74 197 134 197 102 228 72

90 206 235 17 243 134 22 49 169 227 89 16   5 117 16

60 248 230 217 68 138 96 194 131 170 136 10 112 238 238

==========

Rank 1 submatrix:

========== Matrix ==========

90 206 235 17 243 134 22 49 169 227 89 16   5 117 16

60 248 230 217 68 138 96 194 131 170 136 10 112 238 238

184 72 189 163 90 176 42 112 225 212 84 58 228 89 175

244 150 168 219 112 236 101 208 175 233 123 55 243 235 37

225 164 110 158 71 201 78 114 57 48 70 142 106 43 232

26 32 126 194 252 239 175 98 191 94 75 59 149 62 39

==========

Rank 2 submatrix:

========== Matrix ==========

225 164 110 158 71 201 78 114 57 48 70 142 106 43 232

26 32 126 194 252 239 175 98 191 94 75 59 149 62 39

187 32 203 42 190 19 243 13 133 45 61 204 187 168 247

163 194 23 34 133 20 17 52 118 209 146 193 13 40 255

52 227 32 255 13 222 18   1 236 152 46 41 100 233 209

91 141 148 115 175 25 135 193 77 254 147 224 191 161   9

==========

Rank 3 submatrix:

========== Matrix ==========

52 227 32 255 13 222 18   1 236 152 46 41 100 233 209

91 141 148 115 175 25 135 193 77 254 147 224 191 161   9

191 213 236 223 212 250 190 231 251 170 127 41 212 227 19

166 63 161 58 179 81 84 59 18 162 57 166 130 248 71

==========

gather sendcounts (# rows): [5 6 6 4]

gather displacements: [0 45 105 165]

=============== Final Matrix ===============

```
  0 145 164  32 139   3  55  31 184  36  91  29  24 229   0
  0 192 140  67 146 120 113 224  66 177 120 177  91 205   0
  0 185 211  15 218 120  19  44 152 204  80  14   4 105   0
  0 223 207 195  61 124  86 174 117 153 122   9 100 214   0
  0  64 170 146  81 158  37 100 202 190  75  52 205  80   0
  0 135 151 197 100 212  90 187 157 209 110  49 218 211   0
  0 147  99 142  63 180  70 102  51  43  63 127  95  38   0
  0  28 113 174 226 215 157  88 171  84  67  53 134  55   0
  0  28 182  37 171  17 218  11 119  40  54 183 168 151   0
  0 174  20  30 119  18  15  46 106 188 131 173  11  36   0
  0 204  28 229  11 199  16   0 212 136  41  36  90 209   0
  0 126 133 103 157  22 121 173  69 228 132 201 171 144   0
  0 191 212 200 190 225 171 207 225 153 114  36 190 204   0
```

==========

Total time: 0.000090 sec

R = 4 vs various N:

N	Time (with R=4) (sec)
15	0.000090
100	0.002050
1000	0.05490

N = 1k for various R:

R	Time (with N = 1000) (sec)

4	0.05490
8	crash
16	crash
32	crash

The code runs well under all circumstances except that of 8+ ranks. I would expect the values for 8, 16, 32 to be faster and faster, but drop off towards 32, since at that point diminishing returns are likely reached.