Index

$N_p$, 4, 235
$N_p^*$, 5
$P_{ID}$, 4, 235
$\text{np}$, 6
$\text{pid}$; 6
$\text{recvmsg}$, 7
$\text{sendmsg}$, 7
$\text{agg}$, 7
$\text{global_block_ranges}$, 191
$\text{global_block_range}$, 191
$\text{global_ind}$, 7
$\text{local}$, 7
$\text{map}$, 7, 85
$\text{pRUN}$, 9
$\text{put_local}$, 7
$\text{synch}$, 71
$\text{transpose_grid}$, 206
$1*N_p$, 5

Amdahl fraction, 118
Amdahl's Law, 117
Application characteristics, 112

Bandwidth, 112, 116
Bandwidth versus message size, 95
Barrier synchronization, 175
Beamforming, 39

Client/server, 61
Coarse grained block, 113
Code and Cost Modeling (CoCoMo), 135
Coding rate, 136

Communication
  all-to-all, 60, 97, 128
  bandwidth, 126
  complexity, 108
  latency, 126
  nearest-neighbor, 60
  network, 125
  overhead, 98
  pattern, 60
  pipeline, 60
  scatter/gather, 60
  time, 116, 126

Computation-to-communication, 63,
  65, 108, 155
Computational complexity, 108, 152
Computational science, 144
Compute time, 115
Concurrency, 87, 234
Convolution, 22, 57
Corner turn, 128
Critical path, 221

Data structures, 108
Debugging steps, 18, 71, 100
Degrees of parallelism, 108, 113, 154
Discrete Fourier Transform (DFT), 201
Distributed arrays
  block, 5, 81, 239
  block-cyclic, 82, 239
  constructor, 7
  cyclic, 81, 83, 239
  design goals, 80
  flat, 239
fragmented, 78
hierarchical, 239
hybrid, 184, 189, 191, 217, 220, 225
implementation level, 90
index, 185
indexing, 5, 238
local part, 5, 47, 77, 237
model, 62, 70, 130
output, 47
overlap, 62, 66, 81
pure, 78
redistribution, 5, 7, 93

Efficiency
communication, 156
compute, 133, 156
software, 134

Fast Fourier Transform (FFT)
1D, 201
2D, 203, 235
algorithm, 149
bandwidth, 205
Gigaflops, 204

File IO, 30
Filtering in two dimensions, 234
Fixed problem size, 121
Function overloading, 91

Hardware model, 123
High Performance Computing (HPC)
Challenge
characteristics, 154
Kuck diagram, 148
memory hierarchy, 146
spatial/temporal, 147
High Performance Embedded Computing (HPEC)
Challenge, 235
High Performance Linpack (HPL)
algorithm, 151, 217
Gigaflops, 218
Inverse bandwidth, 126
Kuck diagram, 128, 148, 235, 238, 240
Latency, 131
Latency versus message size, 98
Load balancing, 83
Load imbalance, 84
Locality, 87, 234
LU
algorithm, 217
parallel, 219
performance, 221
Manager/worker, 61, 62, 67, 130
Mandelbrot set, 13
Map, 5, 8, 50, 62, 80, 236, 240
Map-based programming, 8, 80
MATLAB
usage, xix
MatlabMPI, 95
Memory bandwidth, 177
Memory hierarchy, 123, 124, 129, 146, 156
Message passing, 7, 62, 65, 69, 95, 130
Message Passing Interface (MPI)
rank, 62
Message Passing Interface (MPI)
size, 62
Multicore, 123
Multinode, 123
Network topology, 125
Owner computes, 62, 78, 131
P notation, 5, 235
Parallel
coarse-grained, 21
data, 113
easy applications, 145
embarrassingly, 12, 103, 172
execution, 9, 92
fine-grained, 12
hard applications, 145
multistage, 41
pipeline, 49, 83, 113, 242
pleasingly, 12
task, 61, 113, 243
Performance, 98
estimating, 157
memory, 55
message size, 98, 126, 187
metrics, 107
multicore, 177, 197, 209
multinode, 177, 198, 211
profiling, 56
time, 55
versus effort, xix, 80, 89, 102, 136, 163
pMatlab
development, xxiv
Polymorphism, 90
Processor Index Tagged FAmiLy of Line Segments (PITFALLS), 93
Processor sizing, 113
Productivity, 137, 166
Programming patterns, 59
RandomAccess
algorithm, 151, 183
Giga Updates Per Second (GUPS), 183
Scaled problem size, 121
Scaling, 102
Scaling out, 178
Scaling up, 178
Simulation science, 144
Single-Instruction Multiple-Data (SIMD), 144
Single-Program Multiple-Data (SPMD), 4, 234
Size, weight, and power, 132
Software cost, 135
Software Lines of Code (SLOC), 100, 135
Spatial locality, 122
Spatial score, 122
Speedup, 20, 119
linear, 100, 119
saturation, 120
sublinear, 119
superlinear, 120
StarP, xxiii
Stream
algorithm, 149, 171
all-to-all, 150
bandwidth, 171
embarrassingly parallel, 149
Strong scaling, 121
Supercomputer, 143
Temporal locality, 122
Temporal score, 122
Throughput, 132
Vectorize, 144
Von Neumann model, 121
Weak scaling, 121