SUBJECT INDEX

A*, 182
Accuracy, 222
Active wait, 243
Adjacency, 201
Adjacency list, 44
Adjacency matrix, 44
Agglomeration, 23, 120, 138
ALAP (as-late-as-possible), 98
Algorithm, 51
Allocated level, 99, 107
α field, 159, 214, 249
$\alpha|\beta|\gamma$ classification, 158
$\alpha|\beta|\gamma$ notation, 214, 249
AltiVec, 8, 154, 156
Ancestor, 43
Annotation, 183
Antidependent, 53
Application, 22, 24
  area, 27
  specification, 22
Architecture, *see* Parallel architecture
Array, 32
  dimension, 34
  element, 32
Array processors, 55, 61, 67, 127
  systolic, 62
  wavefront, 61
ASAP (as-soon-as-possible), 98
Assignment
  spatial, 23, 75
  temporal, 23, 75
Asymptotic notation, 44
Asynchronous, 60
Atomic instruction, 49
Attribution of start times, 118

Bandwidth, 12
β field, 159, 214, 249
BFS, *see* Breadth first search
Bin packing, 139
BINPACKING, 89, 165
Bisection width, 13
Blocking wait, 243
Bottom level, 95, 107
  computation, 96
Branching, 49
Breadth first search (BFS), 46
  algorithm, 46
  complexity, 46
Breeding, 170
Broadcast, 21
Bubble scheduling and allocation (BSA), 149, 158, 191, 200
Buffer size, 202
Bus, 11, 22, 193

C-SCHED, 211
Cache, 76
Cache effect, 76
CASCH, 183
CCR, *see* Communication to computation ratio
CFG, *see* Control flow graph
CG, *see* Computation graph
Chess, 38
Child, 43
Chromosome, 171–177
  direct representation, 173–174
  encoding, 172
  fittest, 181
  indirect representation, 172–173
  node list, 172, 176
  pool, 171
  processor allocation, 172, 176
  representation type, 174–175
Clan, 138
Classic model, 76
Cluster, 120
  assignment, 139–140
  linear, definition, 124
  merge, 121

Task Scheduling for Parallel Systems, by Oliver Sinnen

Cluster of workstations, 11, 37, 195
Clustering, 119–141
  algorithms, 121–124
  contention model, 220, 221
  definition, 120
  dynamic sequence, 137–138
  initial, 121
  linear, 124–128
    algorithm, 127–128
    bound, 125
    contention model, 221
    definition, 124
    example, 128
  list scheduling, 135–138
    contention model, 221
  node duplication, 138–139
  nonlinear, 124
  performance, 141
  principle, 121–122
  scheduling on processors, 141
  single edge, 128–135
    contention model, 221
    example, 131
  to scheduling, 139–141
Code generation, 267
Common buffer, 230
Communication, 32, 49
  concurrent, 76, 87, 188, 209
  conflict, 19
  contention, *see* Contention
  cost, 50, 68, 77, 163, 247
    definition, 50
    function, 77
    network model, 203
  cost free, 76, 81, 87
  delay, 75, 77, 247–248
  device, 231
  distance, 57
  high, 105
  interiteration, 33, 56, 64–65
  interprocessor, 29, 76
  intraiteration, 33, 56, 64–65
  intraprocessor, 76
  layer, 232, 245
  local, 76
  low, 105
  medium, 105
  network, *see* Network
  nonuniform, 58
  overhead, 28
  packet based, 235
  port, 189
  processor, 191
  processor involvement, 228, 268
  remote, 76
  resources, 76
  serialize, 228
  speed, 28
  subsystem, 76, 87, 222, 231, 239
  time, on link, 203
  uniform, 58
  without costs, 86
Communication sensitive data flow graph (CS-DFG), 61
Communication to computation ratio, (CCR), 104–105, 107, 180
  definition, 105
Comparison criteria, 50
Compile time, 74
Compiler, 53, 70
Completion time, 163
Complexity, 44
  analysis, 43
  notational convention, 44
Computation, 49
  cost, 50, 68, 77, 163
    definition, 50
    function, 77
    heterogeneous processors, 155
  cyclic, 32, 55–56, 63, 68
  iterative, 32, 51, 56, 58, 68
  regular, 32, 55
  regularity, 27
  type, 27, 51, 55, 58, 61, 63
  uniform, 61
Computation graph (CG), 61
Computing node, 10
Concurrency, 24
  degree of, 24
    average, 24
    maximum, 24
  inherent, 24
Conservative, 34, 58
Contention, 11, 76, 78, 187–188
  awareness, 188–192, 209, 211
    undirected topology graph, 190
  end-point, 189, 192, 195, 196
  free, 187, 196
  network, 189, 192, 193
Contention aware scheduling, 209–215
  basics, 209–211
  clustering, 220–221
  experimental results, 221, 223
  heuristic, 216, 223
  list scheduling, *see* List scheduling, contention aware

NP-completeness, 214–215
priority schemes, 219
Contention model, 209–210
Control flow, 49, 70
Control flow graph (CFG), 49, 70
Control structure, 29, 50
Control unit, 8
Convex Exemplar, 20
Coordinate system, 53
Cost
communication, *see* Communication, cost
computation, *see* Computation, cost
uniform, 50
Cost matrix, 156
CP, *see* Critical path; Path, critical
CP/MISF (critical path/most immediate successors first), 117
Cray
J90, 9
T3D, 11, 198–199
T3E, 11, 17, 198–200, 222, 231
Cray-1, 8
Critical path, 93, 107
allocated, 107
computed, 99
node, 93
Crossbar, 11, 18
routing, 18
Crossover, 171, 177–179
node list, 178–179
permutation, 178
processor allocation, 177
single-point, 177
two-point, 177
CU, *see* Control unit
Cycle, 42, 58
simple, 42
subpath, 42

DAG, *see* Directed acyclic graph
Data
flow, 7
stream, 7, 51
Data flow graph (DFG), 61
Data flow language, 61
Data item, 60
initial, 60
Data parallel, 8, 188
Data rate, 196, 197, 207
Data ready time (DRT), 79
contention model, 217
cost free communication, 87
involvement-contention model, 258
node duplication, 151
Data-driven execution model, 60–61
Deadline, 105, 159
Deadlock handling, 202
Decision problem, 82
Decomposition
data, 25
exploratory, 25, 26
hybrid, 25
recursive, 25
speculative, 25, 27
Decomposition techniques, 25–27
Degree, 41
in-, 41
mean, 14
out-, 41
Delay, 12, 57, 58, 64, 66, 77, 210
Dependence, 29
analysis, 23, 24, 29–36, 51
anti-, 31, 52, 86
control, 29, 35–36, 49, 70
cycle, 58
data, 29–32, 52
direction vector, 35
distance, 33
distance vector, 34, 53, 66
elimination, 31
false, 31
flow, 30, 52, 64, 86
in scheduling, 32
interiteration, 33, 54
intraiteration, 33, 54
local, 55
loops, 32–35, 53
nonuniform, 53
output, 31, 52, 86
real, 31, 64
relation, 23, 51
structure, 27, 35, 50–51
test, 35
uniform, 33, 53
Dependence graph, 51–55
computer representation, 55
conservative, 55
definition, 51
iteration, *see* Iteration dependence graph
pictorial representation, 53
true, 55
Dependent
anti-, 30
control, 35
flow, 30, 53
output, 31, 53

Depth first search (DFS), 46
  algorithm, 47
  complexity, 46
Descendant, 43
DFG, *see* Data flow graph
DFS, *see* Depth first search
DG, *see* Dependence graph
Direct memory access (DMA), 231
Directed acyclic graph (DAG), 2, 40, 47, 62
  maximum number of edges, 45
Distance vector, *see* Dependence, distance vector
Distributed memory, *see* Memory, distributed
Divide-and-conquer, 25
DLS, *see* Dynamic level scheduling
DMA, *see* Direct memory access
Dominant sequence, 132–135
  definition, 132
DRT, *see* Data ready time
DSC, *see* Dynamic sequence clustering
DSH, *see* Duplication scheduling heuristic
Duplex
  full, 189, 193
  half, 192, 193
Duplication scheduling heuristic (DSH), 138, 153
Dynamic level scheduling (DLS), 115, 118, 158, 190
Dynamic sequence clustering (DSC), 137

Earth Simulator, 9
Eccentricity, 13
(E)CPFD ((economical) critical path fast duplication), 149, 153
Edge, 40
  counter directed, 193
  directed, 41
  finish time, 78, 247
    contention model, 210
    cost free communication, 87
  finish time on link, 203
  finish time on processor, 253
    provisional, 253
  incident on, 41
  order
    dynamic, 132
    static, 130–131, 144
  parallel, 57
  selection, 134
  start time on link, 203
  start time on processor, 240
  undirected, 41
  weight, 50, 57, 62, 210, 247
Edge scheduling, 203–209, 240, 246
  aligned, 205
  concept, 190
  condition, 208
    causality, 204
    on a processor, 245
  definition, 208
  heterogeneous links, 208
  involvement-contention, 246
  nonaligned, 205
  on a processor, definition, 245
  on route, 204–208
  order, 211, 217, 254
  packet view, 206
  provisional, 252
Edge zeroing, 123–124
  implicit, 124, 129
  multiplicity, 124
    edge, 124, 128
    node, 124, 135
    path, 124, 127
Efficiency, 222
Elitism, 176, 181
Embedded systems, 154
Encoding
  permutation, 172
  value, 172
End technique, 109–111
  definition, 110
  optimal, 110
Estimated parallel time (EPT), 123
ETF (earliest time first), 115, 118
Evaluation, 171
Evolution, 170, 177
Execution
  order, 23, 52
  self-timed, 60
  time, 155
    average, 155

Fast Fourier transform (FFT), 38
Feasibility, 55
Feasible, 106. *See also* Schedule, feasible
FG, *see* Flow graph
FIFO, *see* First in first out
Finish time, *see* Edge; Node; Processor
Finish time minimization, 157
First in first out, 46, 60
Fitness function, 171, 176–177
  definition, 176
Flow control, 202

Flow graph (FG), 56–62
  definition, 57
  feasible, 58
  to iteration DG, 66
  to task graph, 64–65
Flynn's taxonomy, 7–9, 25
Forest, 160
  opposing, 160
Fully connected processors, 76, 188

GA, *see* Genetic algorithm
γ field, 163
Gantt chart, 81
Gap, 12
Gap elimination, 253
Gene, 172
Generalization, 164
Generation, 170
Genetic algorithm (GA), 170–182
  basics, 171–172
  complexity, 181
  involvement-contention model, 261
  node duplication, 186
Genetic involvement-contention scheduling (GICS), 261
Grain packing, 138
Grained, 55
  coarse, 25, 101
  fine, *see* Grained, small
  medium, 25
  small, 25
Granularity, 24, 49, 51, 55, 101–105, 107, 165
  coarse, 163, 165
  weak, 101–103, 107
Graph, 40–73
  acyclic, 42, 58, 62
  basic concepts, 40–49
  bipartie, 162
  chain(s), 91, 160
  computer representation, 43–45, 51
  conversion, 64
  dense, 45
  dependence, *see* Dependence graph
  directed, 41, 47
    maximum number of edges, 45
  directed acyclic, *see* Directed acyclic graph
  elementary algorithms, 46–49
  flow, *see* Flow graph
  fork, 83, 160
  fork-join, 106, 161
  harpoon, 161
  interval-ordered, 162
  join, 160
  model, 40, 49
  parallel, 162
  path, *see* Path
  pictorial representation, 41
  send, *see* Graph, fork
  series, 162
  series-parallel, 161
  size, 51, 55, 63
  sparse, 45
  subgraph, 70
  task, *see* Task graph
  topology, *see* Topology graph
  transformation, 64
  undirected, 41
    maximum number of edges, 45
  undirected topology, *see* Undirected topology graph
Gray code, 17
Greedy algorithm, 113

H-SCHED, 156
Heap, 115
Heterogeneity
  communication network, 188
  consistent, 156, 197
  functional, 154, 191
  inconsistent, 156, 197
  link, 197
  uniform, 156
  unrelated, 156
Heterogeneous processors, 154–159, 181
Hierarchical graph, 70
Hitachi S-3600, 8
HLF (highest level first), 117
HLFET (highest level first with estimated times), 117
Homogeneous system, 55, 76
Hop, 190, 199–200
HP SuperDome series, 11, 19, 20
HPF, 232
Hypercube, 17
  dimension, 17
Hyperedge, 194
  adjacency, 201
Hypergraph, 194
Hypertool, 183

IBM
  Blue Gene/L, 11, 17, 21, 231
  RS/6000 SP-SMP, 19–20
  SP-2, 199, 231, 234
IC-SCHED, 249

Idle, 146, 149
Idle time interval, 208, 244
`if`-statement, 35, 49
ILP, *see* Integer linear program
Index, 32
  bound, 58
  variable, 30
  vector, 34
Insertion into gaps, 254
Insertion scheduling heuristic (ISH), 138, 149
Insertion technique, 145–149
  definition, 148
  end technique, 148
Instance, 34, 53, 58, 59
Instruction
  flow, 7
  stream, 7, 51
Integer linear program (ILP), 138, 182
Integer programming, 35
Intel Paragon, 11, 17, 195
Intel Pentium, 8
Interiteration, *see* Communication; Dependence
Intermediate node, 20
Intermediate representation, 55
Intraiteration, *see* Communication; Dependence
Intree, 153, 160
Involvement, 187, 229, 238, 249
  characteristic, 232–236
  direct, 233
  function, 245
  length, 233
  type, 229, 232
Involvement scheduling, 238–250
  algorithmic approaches, 257
  causality condition, 242
  edge scheduling, *see* Edge scheduling
  experimental results, 263
  genetic algorithm, 261
  heuristic, 229, 232, 257, 264
  list scheduling, *see* List scheduling, involvement-contention
  node-edge order, 242
  nonaligned, 243
  NP-completeness, 248, 250
  order overhead-involvement, 242
  priority schemes, 260
  two-phase, 261
Involvement-contention model, 238
  homogeneous system, 240
  task graph, 247
ISH, *see* Insertion scheduling heuristic
Iteration, 32, 57, 64
  number of, 58, 68
  space, 55
Iteration dependence graph, 53–55
  computer representation, 55
  dimension, 53, 66
  granularity, 55
  irregular, 53
  projection, 66
  regular, 54
  to flow graph, 66
Iteration-driven execution model, 57
Iterative
  block, 27
  kernel, 64
Iterative data-flow program, 61
Iterative task graph (ITG), 61

KNAPSACK, 89

LAN (local area network), 37, 195
Latency, 12, 199
Leaf node, 20
Level bound, 97
Linear algebra, 27
Link, 13, 190, 196
  directed, 193, 196
  heterogeneous, 207
  multidirectional, 196
Link set, 196
List scheduling, 108–119
  clustering, *see* Clustering, list scheduling
  contention aware, 216, 218–220
    complexity, 219
    example, 218
  dynamic priorities, 114–115
    complexity, 114–115
  example, 111
  involvement-contention, 257, 261
    complexity, 261
    example, 259
    with processor allocation, 262
  node insertion, 148–149
    complexity, 149
  simple, 108
    complexity, 112
  with processor allocation, 119
    involvement-contention, 262
Load balance, 180
Load balancing, 139
LogGP, 237

LogP, 12, 189, 228, 235–237
  generalized, 237
  scheduling, 238
Lookahead technique, 137
Loop, 32–35, 49, 53, 56
  body, 32, 54, 64–65
  double, 34, 54
  instance, 32
  kernel, 32, 54, 65
    double, 34
  nest, 34, 54
  stride, 35
  transformation, 55

Makespan, 80
Mapping, 23, 51, 61, 75, 118, 120, 254
  load minimization, 140
  wrap, 140
Mapping heuristic (MH), 190
Massively parallel processors (MPP), 11
Matrix–vector multiplication, 25, 27
MCP, *see* Modified critical path
MD, *see* Mobility directed
Meander, 183
Meiko CS-2, 231
Memory
  access policy, 9, 11
  access time, 9
  architecture, 9–11
  centralized, 9
  concurrent access, 11–12
  distributed, 9
  exclusive access, 12
  hierarchy, 10
  local, 10, 13
  location, 9
  organization, 9, 11
  remote, 10
  shared, 9
Memory architecture, 51
Memory bank, 20
Mergesort, 79, 115, 140
Mesh, 15
  acyclic, 15
  bisection width, 17
  cyclic, 15
  degree, 15
  diameter, 15
  dimension, 15
  grid, 15
  linear, 15
  ring, 15
  torus, 15
  wrapped, 15
Message, 193
  long, 237
  size, 13
Message passing, 9
  architecture, 10, 76
Message passing interface (MPI), 222, 231
MH, *see* Mapping heuristic
MIMD, *see* Multiple instruction multiple data
MISD, *see* Multiple instruction single data
MMX, 8
Mobility directed (MD), 139, 149
Model
  asynchronous, 13
  synchronous, 12
Modified critical path (MCP), 117, 137
MPI, *see* Message passing interface
MPP, *see* Massively parallel processors
Multigraph, 57
Multiple instruction multiple data (MIMD), 8, 69
Multiple instruction single data (MISD), 7
Multiprocessor
  centralized shared-memory, 9
  distributed memory, 10
  distributed shared-memory, 10
Mutation, 170–171, 179–180
  node list, 180
  processor allocation, 179–180
  swap, 179–180

NEC SX-6, 9
Network, 13–22
  binary tree, 20, 195
  blocking, 15, 19
  butterfly, 19
  capacity, 13, 237, 238
  cost free, 86
  degree, 13
  diameter, 13
  dimension, 15, 17
  direct, 13
  dynamic, 18–22, 194
    properties, 21
  fat tree, 21
  fully connected, 14, 209, 220
  grid, 15
  hardware costs, 14
  hypercube, 17
  indirect, 13, 18
  interface, 189, 195
  linear, 15
  mesh, 15
  model, 192–202

Network (*continued*)
modular topology, 18
multistage, 19
nonblocking, 15, 18
omega, 19
ring, 15, 193
star, 214, 220
static, 13–18
properties, 16
subtopology, 18
topology, 13, 187
torus, 15
tree, 20
type, 13
vertex, 196
Node, 49. *See also* Vertex
branch, 49
fictive, 147
finish time, 77
heterogeneous processors, 155
free, 109
definition, 109
partially, 137
grain, definition, 101
input, 49
level, 95
computation, 96
computing, 99
examples, 99
order, 119
optimal, 85, 111
ordering, 130
out-degree, 152
output, 49
priority, 115–118
dynamic, 114, 117–118
start times, 117
static, 114–117
tie(s), 116–117, 142
ready, 117, 119
redundant, 154
root, 160
sink, 93
source, 93
start time, 75
duplication, 151
strictness, 49, 52, 58, 60, 78
type, 49
weight, 50, 62
Node duplication, 150–154
clustering, 138–139
contention model, 226, 227
heuristic, 153–154
primary candidate, 152
redundant node removal, 154
Node order anomaly, 151–152
Node-processor pair, 114
Nonbacktracking, 124
Nonblocking, 243, 246, 264
Nonuniform, 33
Nonuniform memory access (NUMA), 10, 76
NP-complete, 82
strong sense, 83
weak sense, 84
NP-completeness, 82
strong sense, 84
weak sense, 84, 106
NUMA, *see* Nonuniform memory access

Offspring, 171
One-port model, 189, 196, 214
One-sided, 229–230
OpenMP, 232
Operator, 171
Optimality criterion, 163
Orchestration, 23, 120, 138
OREGAMI, 183
Outtree, 153, 160
Overhead, 12, 232, 239, 249
function, 245
length, 233
Overlap, 78, 231

Packet, 199
dropping, 202
Parallax, 183
Parallel
computation, 49
time, 222
Parallel architecture, 7–13, 51
Parallel computer, 7
Parallel programming, 22
Parallel random access machine (PRAM), 12
Parallel system, 7, 11. *See also* Target parallel system
consistent, 156
dedicated, 76, 86
inconsistent, 156
uniform, 156
unrelated, 156
Parallelism, 24, 49
inherent, 24
Parallelization, 22–24
hardware-oriented, 61
loop, 53

manual, 28
process, 23–24
technique, 28
tool, 182
Parallelizing compiler, 28, 31, 53, 55
Parent, 43
Partial order, 23, 52
Partial representation, 55
Partially connected processors, 181
PARTITION, 84, 106
3-PARTITION, 83
Partitioning
input data, 25
intermediate data, 25
output data, 25
Passive wait, 246
Path, 41
bottom, 95, 100
contains, 41
critical, *see* Critical path
cycle, *see* Cycle
length, 42, 248
allocated, 93, 117
computation, 92
in task graph, 92
with overhead, 248
level, 97
shortest, 46
simple, 42
subpath, 42
top, 96, 100
PE, *see* Processing element
Perfect shuffle, 19
Performance bottleneck, 22
Point-to-point, 193
Polynomial time, 82
Population, 170, 171
initial, 176
size, 176
PowerPC, 8
PRAM, *see* Parallel random access machine
Precedence constraint, 22, 52, 160
Precedence order, 31, 52, 78
Precedence relation, 160
Predecessor, 42
Preemption, 105, 159
Preemptive, 76, 87
Problem
instance, 159
type, 159
Processing element (PE), 8
Processing node, 20
Processing speed, 155
Processor, 9, 13, 189, 196
active, 230
allocation
definition, 75
optimal, 85
available, 85, 90
dedicated, 76, 87
dual, 197
environment, 159
functional capability, 154
heterogeneous, *see* Heterogeneous processors
idle, 146
involvement, *see* Involvement
number, 28, 159
fixed, 159
unlimited, 85, 89, 159
passive, 231
physical, 120
speed, 154
used, 86, 90
definition, 80
virtual, 120, 191
Processor allocation, 75, 118, 261
node duplication, 151
Processor finish time, 80
Processor heterogeneity, 154–155
Processor network, 13, 21
Program, 22
execution, 50
feasible, 58, 59, 62
definition, 52
lemma, 52
formulation, 28
input, 63
linear, 32
model, 49
transformation, 51, 55
Programming
message passing, 11, 77
model, 11–13, 51
paradigm, 11–13, 29
shared memory, 11
technique, 24
Projection, 59, 66
linear, 67, 127
PVM, 232
PYRROS, 183

Queue, 46, 60
priority, 115

r-SCT, 163
Retiming, 61

Reachable, 43
Real time, 154
Real time system, 191
Reassignment, 31
Receive, 11
Receiving, 12
Recursive algorithm, 46
Reduction, 164
  graph, 164
Regularity, 51
Reordering, 53
Reproduction, 177–180
Rescheduling, 149, 254
Root node, 20
Round robin, 166
Route, 18, 200, 204
Routing, 198, 202
  adaptive, 198
  algorithm, 198, 200
    complexity, 201
  cut-through, 199
  delay, 199, 200
  dimension ordered, 201
  minimal, 201
  nonadaptive, 198
  static, 198
  store-and-forward, 199, 209
Runtime, 74

Safeguard, 179
Scalability, 11
Scalable coherent interface (SCI), 17, 193
Scalar, 32
SCHED, 83
SCHED-C0, 89
Schedule
  accuracy, 222
  compaction, 253
  condition
    data ready time constraint, 80
    exclusive processor allocation, 77
    link constraint, 202
    precedence constraint, 78, 151
  definition, 75
  example, 81–82, 87–88
  execution time, 222
  feasible, 79
  implicit, 122–123
  length, 80
  optimal, 82
  partial, 92, 109
Scheduling, 23, 24, 51, 61
  communication contention in, 187–226
  complexity, 158–170
    heterogeneous processors, 170
    with communication costs, 82–86, 165–168
    with node duplication, 168–170
    without communication costs, 88–92, 165
  condition, 147
  contention aware, *see* Contention aware scheduling
  direct, 251
  dynamic, 74
  entering edges, 251
  framework, 74
  fundamentals, 74–76
  heterogeneous processors, 157–158
    heuristic, 157–158
  involvement, *see* Involvement scheduling
  list, *see* List scheduling
  node, involvement-contention model, 247
  NP-completeness, 83, 85
    contention model, 211
    heterogeneous processors, 156
    involvement-contention model, 248
    unlimited number of processors, 85
    without communication costs, 89
  optimal, unlimited processors, 90
  phase two, 119
  problem, 82
    general, 75
  process, 82
  static, 74
  tool, 182
  trade-off, 82
  two-phase, 118
  with communication costs, 76–86
  with processor allocation, 118–119, 254
  without communication costs, 86–92
SCI, *see* Scalable coherent interface
Scientific computing, 25
SCT, *see* Small communication time
Seed, 176
Selection, 171, 180–181
  roulette wheel, 181
  tournament, 180
Self-loop, 41
Send, 11
Sending, 12
Sequent NUMA-Q, 11
Sequential computer, 7
Sequential implementation, 22
Sequential order, 49, 91
Sequential program, 28
Sequential programming, 22

Sequential time, 81
SFG, *see* Signal flow graph
SGI Origin, 11, 17, 19, 230
Shared memory, *see* Memory, shared
  get, 230
  put, 230
Signal flow graph (SFG), 61
Signal processing, 27, 55
SIMD, *see* Single instruction multiple data
Simulated annealing, 170
Single instruction multiple data (SIMD), 8, 25, 55
Single instruction single data (SISD), 7
Single processor, 22
Single program multiple data (SPMD), 9, 55, 69
SISD, *see* Single instruction single data
Slot, 147
Small communication time (SCT), 163
SMP, *see* Symmetric multiprocessors
Software layer, 231
Software pipelining, 54, 61
sp-graph, *see* Graph, series-parallel
Speed, 155
  relative, 156
Speedup, 222
SPMD, *see* Single program multiple data
SSE, 8, 154, 156
Start time, *see* Edge; Node; Processor
Start time minimization, 111–114
Start time order, 78
Statement, 30, 32, 35, 49, 51, 53
  conditional, 29, 50
Steps of refinement, 121
Stochastic search, 170
Subscript, 34
  function, 55
Subtask, 22
Subtask decomposition, 23–29, 138
Successor, 42
Sun Enterprise, 11
  3500, 221
  10000, 11, 19–20
Sun Fire E25K, 19–20
Survival of the fittest, 171, 180
Switch, 13, 18, 194–195, 197
  implicit, 14, 195
  internal, 18
  vertex, 194, 197
Switching
  circuit, 199
  packet, 199
  strategy, 199
Symmetric multiprocessors (SMP), 9
Synchronization, 11, 60
Synchronous, 61

Target parallel system, 28, 76
  classic model, 76
  contention model, 209
  cost free communication, 86
  heterogeneous processors, 155
  involvement-contention model, 238
Target system, 50
Task, 22, 30, 49
  characteristic, 159
  independent, 160
Task duplication, 163
Task graph, 2, 40, 62–69
  bottom of, 96
  contention model, 210
  costs, 158
  definition, 62
  granularity, 101
  involvement-contention model, 245, 247
  kernel, 65, 66
  limitations, 68
  motivations, 68
  properties, 92–105
  sample, 62
  scheduled, definition, 132
  simplified, 62
  structure, 158
  to dependence graph, 64
  top of, 96
  unrolled, 65
  weak granularity, definition, 102
Task Grapher, 183
Task interaction graph (TIG), 70
Task scheduling, *see* Scheduling
TCG, *see* Temporal communication graph
TCP/IP, 229, 231
Technique, 51
Temporal communication graph (TCG), 70
Thinking Machines CM-5, 11
Third party, 229
Thread, 49
TIG, *see* Task interation graph
Token, 60
Top level, 95, 107
  computation, 96
Topological order, 47, 52, 78, 89
  definition, 47
  lemma, 47
Topological sort algorithm, 48
Topology, star, 223

Topology graph, 192–197
  breath first search, 201
  definition, 196
  shortest path, 201
  symmetric, 220
Tree, 160
Triplet, 83
Two-sided, 229

Unit communication time (UCT), 163
Unit execution time (UET), 153, 163
Uniform memory access (UMA), 9, 76
Undirected topology graph, 189
Unfolding, 59, 61
Unrolling, 59, 65
  partial, 66

Variable
  array, 32
  boolean, 36
  input, 55
  renaming, 31
  reuse, 31
Vector
  architecture, 8
  machine, 8
  processor, 8
Vertex, 40
  adjacent, 41, 201
  degree, *see* Degree
  destination, 41
  distance, 13, 46
  entry, 43
  exit, 43
  origin, 41
  relationships, 42
    in cycle, 43
  sink, 43
  source, 43
Virtual edge, definition, 132
VLSI, 55, 61, 67, 127
von Neumann model, 7, 12

Wrap around, 20

Zero-copy, 234