Index

abstract partition, 78
abstraction, 13, 52
abstraction type (distribution), 13
access transparency, 10
active object, 38
adaptive lazy flushing, 46
All-Pairs Shortest Paths Problem, 34
allocation
    annotation, 83
    initial, 65
array
    global, 82, 83
ASP, 34
asynchronous object migration, 57, 91
asynchronous watching, 92
automatic distribution, 55, 113
automatic graph assignment, 80
automatic strategy-finding, 79
base object, 91
batching, 109
binding (to existing instance), 16
broadcast
    totally ordered, 41
C++, 13
C++/++, 38
C#, 13, 29
caching, 13, 16, 34, 40, 47
call graph construction, 59
causal consistency, 41
centralized program, 8
dependency closure, 62
diffing, 40
direct object (Emerald), 18
dispersal object, 21
distribute (verb), 8
distributed computing (A Note on...), 50
Distributed Hello World, 84
distributed object, 21
distributed object system, 10
distributed program, 8
Distributed Shared Memory, 14, 40
Distributed Smalltalk, 18
distribution
    automatic, 55, 113
    naı̈ve, 111
distribution abstraction layer, 13
distribution assembler, 82
communication mechanism, 13
compilent, 21
complexity
    of object graph analysis, 67
component, 21
compostance, 21
compound operator, 83
computation migration (Jackal), 46
context-sensitive, 58
copyable (Doorastha), 36
CORBA, 26
core generation, 84
CPU, 8
de-facto local call, 99
deployment, 62
dependency closure, 62
diffing, 40
direct object (Emerald), 18
dispersal object, 21
distribute (verb), 8
distributed computing (A Note on...), 50
Distributed Hello World, 84
distributed object, 21
distributed object system, 10
distributed program, 8
Distributed Shared Memory, 14, 40
distributed Smalltalk, 18
distribution
    automatic, 55, 113
    naı̈ve, 111
distribution abstraction layer, 13
distribution assembler, 82

119
distribution platform, 10
distribution strategy, 55, 77
distribution technology, 10
distribution transparency, 10, 50, 113
Do, 37
Doorastha, 35
code generation, 88
platform performance, 99
DOWL, 20
DSM, 14, 40
dynamic object, 61
Eiffel, 21, 27
Eiffel//, 38
Emerald, 17
Equanimity, 22
evolution, 50
of distributed object systems, 13
exceptions, 64, 72
explicit distribution, 14, 15
export edge, 63
factory method, 48, 103
false sharing, 40
fan-in, 23
fan-out, 23
flow-insensitive, 58
folding, 63
function shipping, 41
future (transparent), 38
generation dialog, 76
global array, 82, 83
global object (Emerald), 18
global type (Pangaea), 81
globalizable (Doorastha), 36
globalization (Pangaea), 83
graph assignment algorithm, 80
graph editor, 73
graph partitioning, 80
GUI, 73
Hamming’s Problem, 69, 78
Hello World (Distributed), 84
HERON, 21
host, 8
Hyperion, 43
IDL, 27, 85
immutability, 79, 116
and equality, 90
performance impact, 108
immutability analysis, 56
immutable, 16, 47
implicit distribution, 14, 40
import edge, 63
increment operator, 83
indefinite object, 61
industry standard, 24
inherently distributed, 9
initial allocation, 65
interactive response time, 111
interface type (CORBA), 27
interoperability (CORBA), 27
invalidation, 35, 40
irrelevant type, 73
Jackal, 46
JacORB, 26
Java, 13
Java//, 38
Java/DSM, 43
Java/RMI, 24
JavaNaise, 34
JavaParty, 30
code generation, 87
platform performance, 99
JHotDraw, 73
Juggle, 44
KaRMI, 31
language mapping (CORBA), 28
latency, 50
Launcher, 57, 96
local object (Emerald), 18
local object (Pangaea), 77
local type (Pangaea), 81
location transparency, 10
machine, 8
Manta, 32
marshalling, 15
memory access, 50
message passing, 13
MICO, 26
middleware, 10, 14
migratable (Doorastha), 36
migratable (Pangaea), 77
migratable type (Pangaea), 81
migration, 13, 16, 31, 44
asynchronous, 57, 91
single chance, 48
strong, 16
weak, 16
migration instrumentation, 82
migration strategy, 91
migration transparency, 10
mobile agents, 14
module-based, 17, 22
MPI, 13
name mapping, 82
name server, 16, 85
network reference, 15
node, 8
object graph analysis, 57, 58, 114
object population, 64
object-based, 17
object-graph aggregation (Jackal), 46
object-orientation, 10
Objects-by-Value (CORBA), 85
ODP, 10
OMG, 26
ontology (object graph analysis), 61
operator (design pattern), 37
Orbix, 26
Orca, 41

Panda, 32
Pangaea
  architecture, 55
  meaning of word, 7
paradigm
  object-oriented, 13
parallelization, 7
partial failure, 51, 56
partition, 8
  abstract, 78
  root, 93
partition object, 93
pass-by-copy, 36
pass-by-move, 18, 25, 36, 91
pass-by-refvalue, 36
pass-by-value, 25
pass-by-visit, 18
performance
  global arrays, 104
  migration subsystem, 104
  object graph analysis, 70
remote vs. local calls, 99
persistent, 34
platform, 10
PM2, 43
POA (CORBA), 28
points-to analysis, 60
ProActive, 38
processor, 8
producer/consumer, 67
project summary, 113
proof, 90
proxy, 15, 47
proxy-in, 35
proxy-out, 35
PVM, 13
ray tracer, 105
ray tracer (Juggle), 44
RC5, 101
reactiveness (Smalltalk), 18
read-locally, 47
read-only in practice, 47
read/write ratio, 42
reference propagation, 66
reflection, 38, 62
region (Jackal), 46
registry (RMI), 24
release consistency, 41
remote inheritance, 21
remote invocation, 13
remote object creation, 16, 85
Remote Object Invocation, 15
Remote Procedure Call, 15
RemoteException, 24, 52
remotenew (Doorastha), 88
remoting (C#), 29
replication, 13, 16, 20, 33, 44
replication transparency, 10
RepMI, 33
response time, 111
RM-ODP, 10
RMI, 24
ROI, 15
root partition, 93
RPC, 15, 22
run-time system (Pangaea), 91
scope analysis, 55, 116
sequential consistency, 41
shared object, 41
shortcut operator, 83
single chance migration, 48
skeleton, 15
Smalltalk
Distributed, 18
sockets, 13
speedup, 8, 105, 108
standard, 24
static analysis, 53
static object, 61
static variable, 94, 116
strategy
distribution, 55, 77
migration, 91
stub, 15
synchronized methods, 90
synchronous watching, 92
system type, 73
threads & objects (programming model), 14
totally ordered broadcast, 41
transparency (distribution), 10
transparent future, 38
Travelling Salesman Problem, 34
TreadMarks, 43
Trellis, 20
TSP, 34
twining, 40
type filter, 73
type graph, 63
type inference, 59
unique (Doorastha), 89, 94
update protocol, 40
usage edge, 63
user interface, 73
value type (CORBA), 27
Visibroker, 26
visitor pattern, 90
Waldo, 50
watcher object, 91
weak consistency, 41
write-invalidate, 40
write-update, 40, 41
X Window System, 23