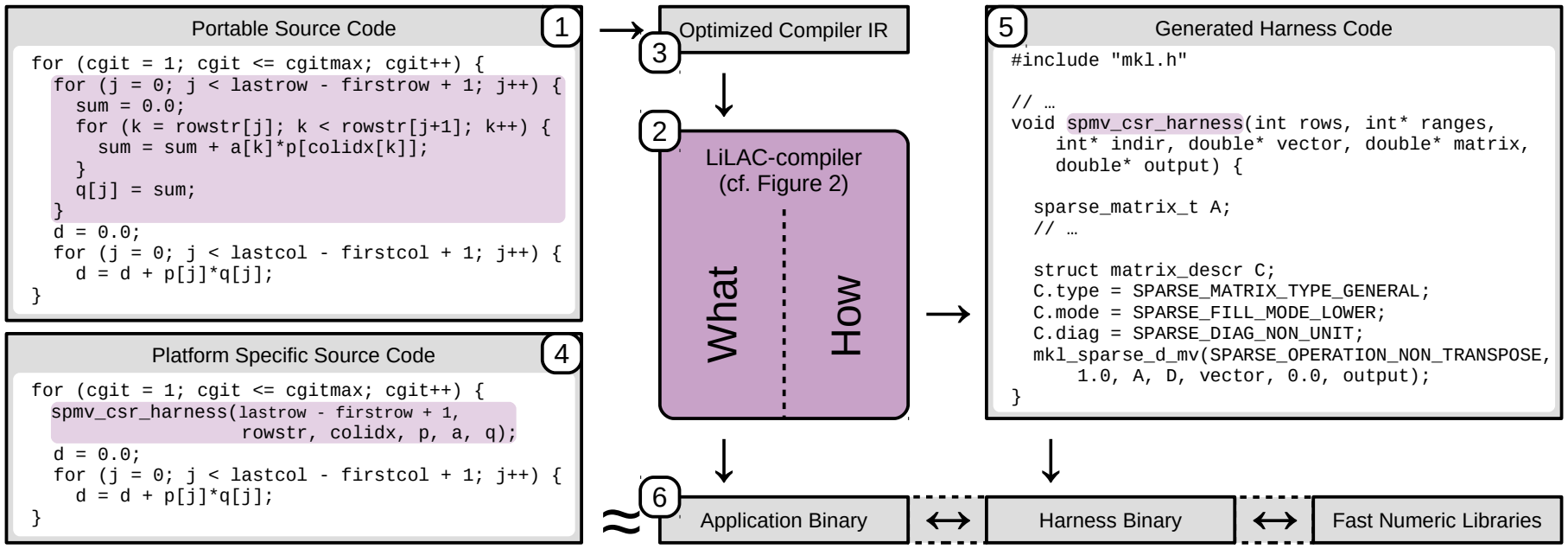
Portable Source Code
1
for (cgit = 1; cgit <= cgitmax; cgit++) {
for (j = 0; j < lastrow - firstrow + 1; j++) {
sum = 0.0;
for (k = rowstr[j]; k < rowstr[j+1]; k++) {
sum = sum + a[k]*p[colidx[k]];
}
q[j] = sum;
}
d = 0.0;
for (j = 0; j < lastcol - firstcol + 1; j++) {
d = d + p[j]*q[j];
}
Platform Specific Source Code
4
for (cgit = 1; cgit <= cgitmax; cgit++) {
spmv_csr_harness(lastrow - firstrow + 1,
rowstr, colidx, p, a, q);
d = 0.0;
for (j = 0; j < lastcol - firstcol + 1; j++) {
d = d + p[j]*q[j];
}
3
Optimized Compiler IR
2
LiLAC-compiler
(cf. Figure 2)
What
How
5
Generated Harness Code
#include "mkl.h"
// …
void spmv_csr_harness(int rows, int* ranges,
int* indir, double* vector, double* matrix,
double* output) {
sparse_matrix_t A;
// …
struct matrix_descr C;
C.type = SPARSE_MATRIX_TYPE_GENERAL;
C.mode = SPARSE_FILL_MODE_LOWER;
C.diag = SPARSE_DIAG_NON_UNIT;
mkl_sparse_d_mv(SPARSE_OPERATION_NON_TRANSPOSE,
1.0, A, D, vector, 0.0, output);
}
6
Application Binary
Harness Binary
Fast Numeric Libraries