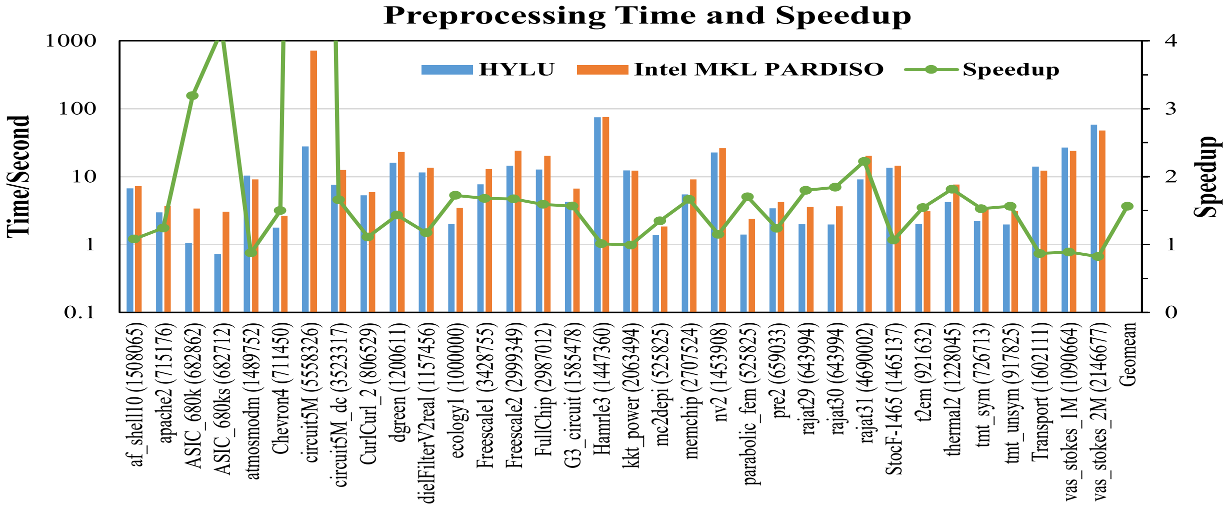
Test Results of HYLU and Performance Comparisons with Intel MKL PARDISO

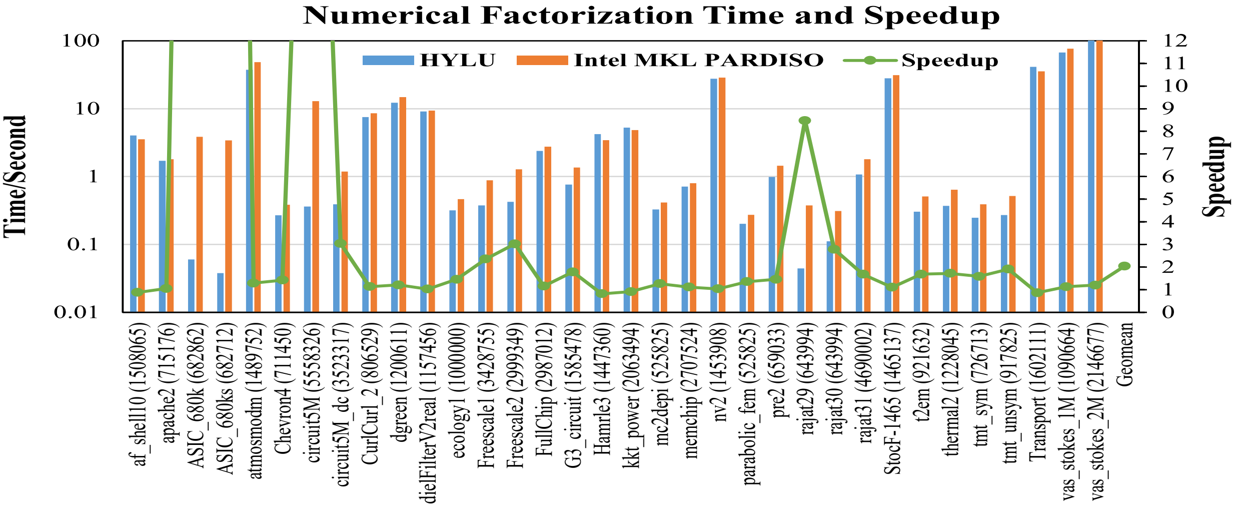
The experiments were carried out on a Linux server. The main hardware and software configurations are listed in the following table. All results presented in this document are wall-time measurements from **16-thread parallel execution**.

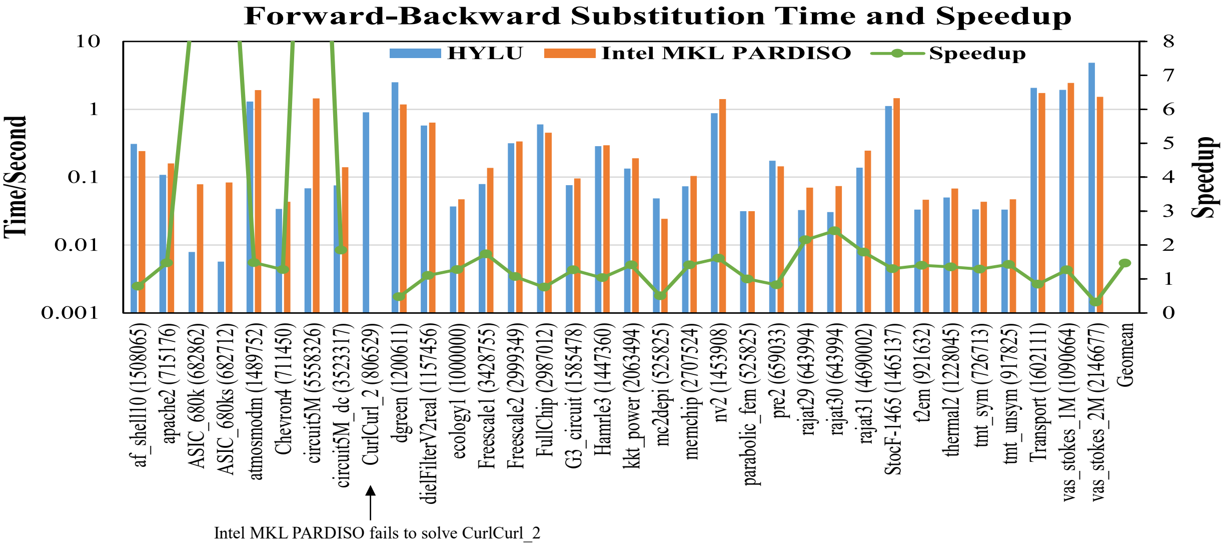
CPU	Intel Xeon Gold 6130 @ 2.1GHz
Memory	256GB
Operating system	Ubuntu 24.04 LTS for tests/CentOS 7.9 for compiling HYLU library
Compiler	gcc 13.3.0 for compiling test code/gcc 4.8.5 for compiling HYLU library
MKL version	2025.2.0.629
HYLU version	20251222
Benchmarks	34 matrices from SuiteSparse Matrix Collection, dimensions from 525,825 to 5,558,326

1. One-Time Solve

On geometric mean, HYLU achieves a **2.04X speedup in numerical factorization** compared with Intel MKL PARDISO, while the preprocessing and forward-backward substitution phases are also faster (**1.56X** and **1.46X** speedups, respectively). For the total one-time solve time (preprocessing + numerical factorization + forward-backward substitution), HYLU is **1.67X faster** than Intel MKL PARDISO on geometric mean.

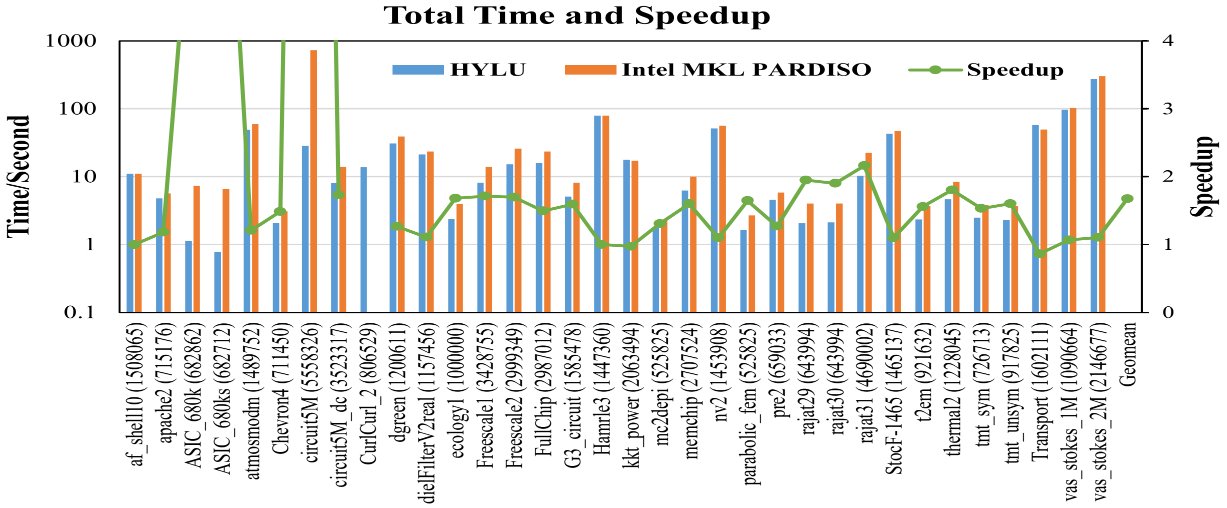






Forward-Backward Substitution Time and Speedup

Intel MKL PARDISO fails to solve CurlCurl_2

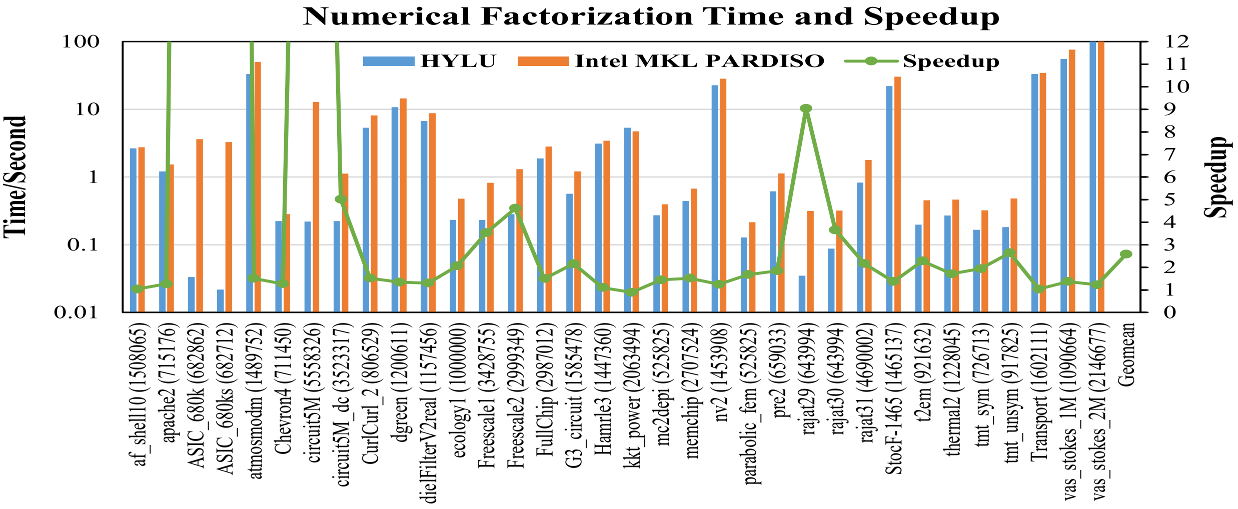


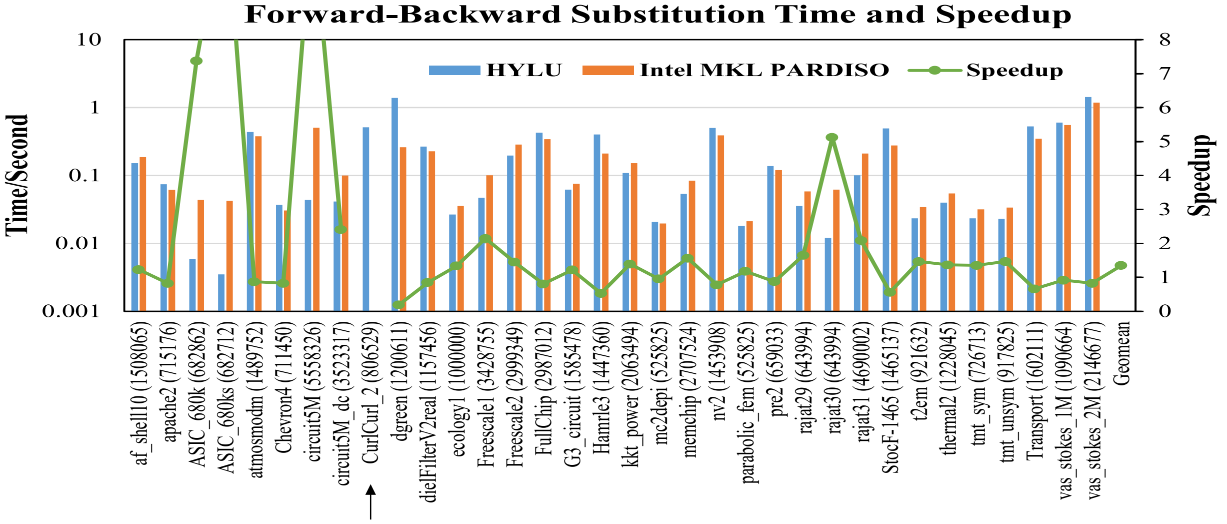
Total Time and Speedup

Another advantage of HYLU is its stable performance for various sparsities. Intel MKL PARDISO generates a huge quantity of fill-ins for some benchmarks (ASIC_680k, ASIC_680ks, and circuit5M), and the performance of these matrices is poor.

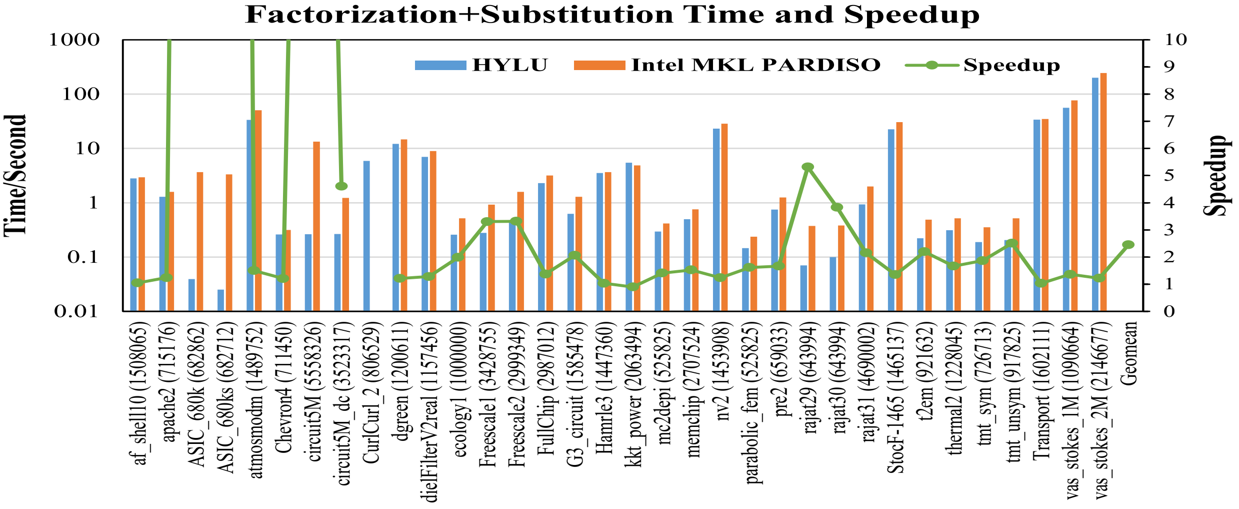
2. Repeated Solve

HYLU offers an optimization option for repeated solve of linear systems with an identical sparse pattern in the coefficient matrix. In this case, HYLU achieves a **2.58X geometric mean speedup in numerical factorization** over Intel MKL PARDISO, while the forward-backward substitution phase is slightly faster (**1.35X** speedup). When comparing the total time of numerical factorization and forward-backward substitution, HYLU is **2.46X** faster than Intel MKL PARDISO on geometric mean. In this scenario, the preprocessing phase is on geometric mean 1.76X slower than that of the one-time solve scenario. However, in the repeated solve scenario, the pre-processing time is less important, as it is executed only once.





Intel MKL PARDISO fails to solve CurlCurl_2


Factorization+Substitution Time and Speedup
Time/Second
Speedup
HYLU
Intel MKL PARDISO
Speedup
af_shell10 (1508065)
apache2 (715176)
ASIC_680k (682862)
ASIC_680ks (682712)
atmosmodm (1489752)
Chevron4 (711450)
circuit5M (5558326)
circuit5M_dc (3523317)
CurlCurl_2 (806529)
dgreen (1200611)
dielFilterV2real (1157456)
ecology1 (1000000)
Freescale1 (3428755)
Freescale2 (2999349)
FullChip (2987012)
G3_circuit (1585478)
Hamrle3 (1447360)
kkt_power (2063494)
mc2depi (525825)
memchip (2707524)
nv2 (1453908)
parabolic_fem (525825)
pre2 (659033)
rajat29 (643994)
rajat30 (643994)
rajat31 (4690002)
StocF-1465 (1465137)
t2em (921632)
thermal2 (1228045)
tmt_sym (726713)
tmt_unsym (917825)
Transport (1602111)
vas_stokes_1M (1090664)
vas_stokes_2M (2146677)
Geomean