BLAS and LAPACK comprise all the low-level linear algebra subroutines that handle your matrix operations in R and other software. Fedora ships the reference implementation from Netlib, which is accurate and stable, but slow, as well as several optimized backends, such as ATLAS, BLIS (serial, OpenMP and threaded versions) and OpenBLAS (serial, OpenMP and threaded flavours as well). However, up to version 32, Fedora lacked a proper mechanism to switch between them.

We are excited to announce that this situation changes with the upcoming release, which is already in beta status. Starting with Fedora 33, R (as well as Numpy, Octave and all the other BLAS/LAPACK consumers) is linked against the outstanding FlexiBLAS library, a BLAS/LAPACK wrapper that enables runtime switching of the optimized backend, and the OpenMP version of OpenBLAS is set as the default system-wide backend.

Moreover, the accompanying `flexiblas` R package enables changing the BLAS/LAPACK provider, as well as setting the number of threads for parallel backends, without leaving the R session. Let's give this a quick test using docker:

```
$ docker run --rm -it fedora:33
$ dnf install R-flexiblas # install R and the FlexiBLAS API interface
for R
$ dnf install flexiblas-* # install all available optimized backends
```

Then, in an R session we see:

```
library(flexiblas)

# check whether FlexiBLAS is available
flexiblas_avail()
#> [1] TRUE

# get the current backend
flexiblas_current_backend()
#> [1] "OPENBLAS-OPENMP"

# list all available backends
flexiblas_list()
#> [1] "NETLIB"           "__FALLBACK__"     "BLIS-THREADS"
"OPENBLAS-OPENMP"
#> [5] "BLIS-SERIAL"      "ATLAS"            "OPENBLAS-SERIAL"
"OPENBLAS-THREADS"
#> [9] "BLIS-OPENMP"

# get/set the number of threads
flexiblas_set_num_threads(12)
flexiblas_get_num_threads()
#> [1] 12
```

This is an example of GEMM benchmark for all the backends available:

```
library(flexiblas)

n <- 2000
```

```
runs <- 10
ignore <- "__FALLBACK__"

A <- matrix(runif(n*n), nrow=n)
B <- matrix(runif(n*n), nrow=n)

# load backends
backends <- setdiff(flexiblas_list(), ignore)
idx <- flexiblas_load_backend(backends)

# benchmark
timings <- sapply(idx, function(i) {
  flexiblas_switch(i)

  # warm-up
  C <- A[1:100, 1:100] %*% B[1:100, 1:100]

  unname(system.time({
    for (j in seq_len(runs))
      C <- A %*% B
  })[3])
})

results <- data.frame(
  backend = backends,
  `timing [s]` = timings,
  `performance [GFlops]` = (2 * (n / 1000)^3) / timings,
  check.names = FALSE)

results[order(results$performance),]
#>            backend timing [s] performance [GFlops]
#> 1           NETLIB     56.776            0.2818092
#> 5            ATLAS      5.988            2.6720107
#> 2     BLIS-THREADS      3.442            4.6484602
#> 8      BLIS-OPENMP      3.408            4.6948357
#> 4      BLIS-SERIAL      3.395            4.7128130
#> 6  OPENBLAS-SERIAL      3.206            4.9906425
#> 7 OPENBLAS-THREADS      0.773           20.6985770
#> 3  OPENBLAS-OPENMP      0.761           21.0249671
```