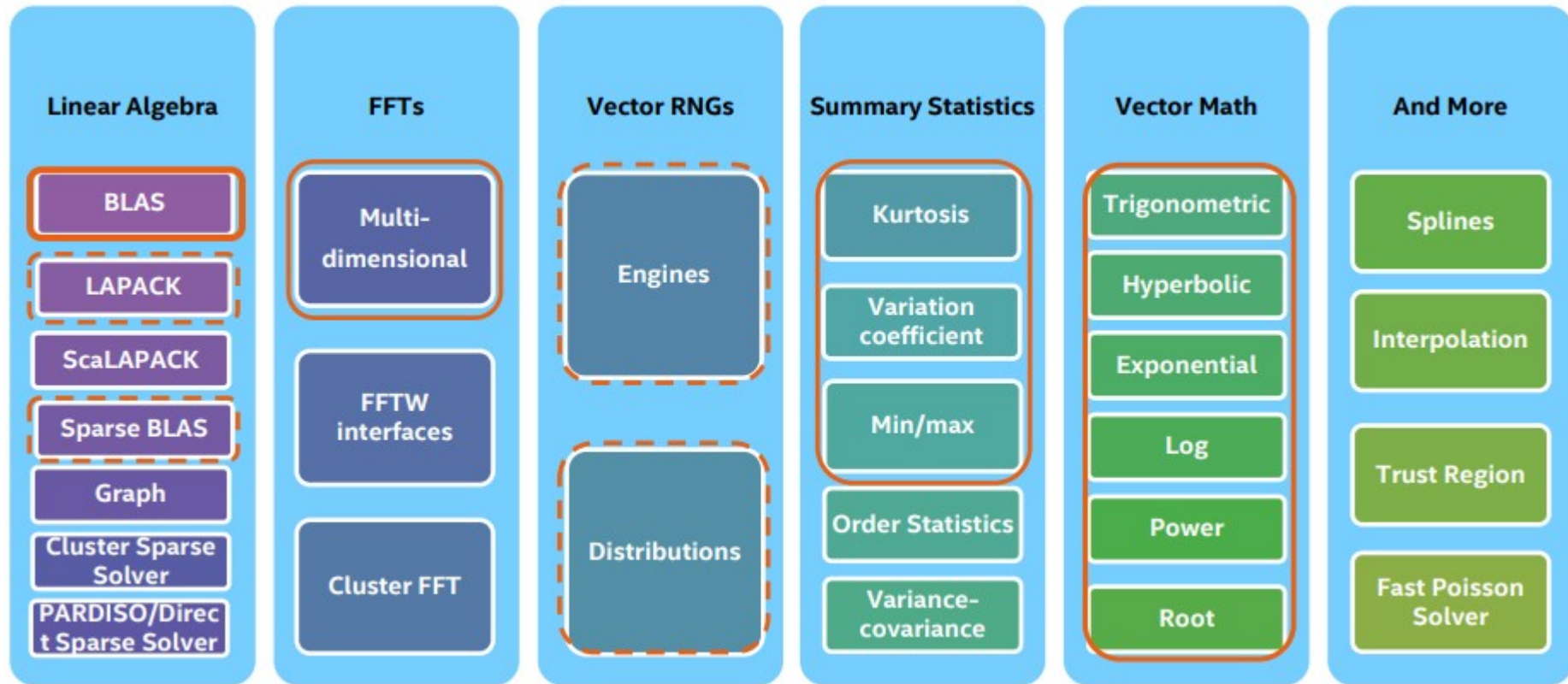


Intel(R) oneMKL

Introduction aux nouveautés de la librairie
Math Kernel Library
Support SYSL

Que contient oneMKL



Utilisable pleinement sur GPU



Utilisable partiellement sur GPU

Nouveautés

- Support des GPU Intel
- Support de DPC++
 - BLAS – Support complet CPU & GPU
 - LAPACK – support partiel via les Buffers pour CPU & GPU
 - Vecteurs – support partiel CPU & GPU

NVIDIA, AMD

OneMKL permet également l'utilisation d'autres types de GPU

User Application	oneMKL Layer	Third-Party Library	Hardware Backend
oneMKL interface	oneMKL selector	Intel(R) oneAPI Math Kernel Library for x86 CPU	x86 CPU
		Intel(R) oneAPI Math Kernel Library for Intel GPU	Intel GPU
		NVIDIA cuBLAS for NVIDIA GPU	NVIDIA GPU
		NVIDIA cuSOLVER for NVIDIA GPU	NVIDIA GPU
		NVIDIA cuRAND for NVIDIA GPU	NVIDIA GPU
		NETLIB LAPACK for x86 CPU	x86 CPU
		AMD rocBLAS for AMD GPU	AMD GPU
		AMD rocRAND for AMD GPU	AMD GPU

<https://github.com/oneapi-src/oneMKL>

GEMM avec oneMKL

```
int main() {  
  
    int64_t m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;  
    int64_t sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;  
    double alpha = 1.0, beta = 0.0;  
  
    // Allocate matrices  
    double *A = (double *) mkl_malloc(sizeof(double) * sizea);  
    double *B = (double *) mkl_malloc(sizeof(double) * sizeb);  
    double *C = (double *) mkl_malloc(sizeof(double) * sizec);  
  
    // Initialize matrices [...]  
    ...  
    cblas_dgemm(CblasColMajor, CblasNoTrans, CblasNoTrans, m, n, k,  
                alpha, A, lda, B, ldb, beta, C, ldc);  
    ...  
}
```

GEMM avec DPC++

```
int main() {  
    using namespace oneapi::mkl;  
  
    int64_t m = 10, n = 6, k = 8, lda = 12, ldb = 8, ldc = 10;  
    int64_t sizea = lda * k, sizeb = ldb * n, sizec = ldc * n;  
    double alpha = 1.0, beta = 0.0;  
  
    sycl::queue Q{sycl::gpu_selector{}};  
  
    // Allocate matrices  
    double *A = malloc_shared<double>(sizea, Q);  
    double *B = malloc_shared<double>(sizeb, Q);  
    double *C = malloc_shared<double>(sizec, Q);  
  
    // Initialize matrices [...]  
    ...  
  
    auto e = blas::gemm(Q, transpose::N, transpose::N, m, n, k,  
                        alpha, A, lda, B, ldb, beta, C, ldc));  
    ...  
}
```

Utilisation du namespace oneapi::mkl

Initialisation d'une queue rattachée au GPU

Allocation de mémoire partagée entre le CPU et le GPU

Nouveau paramètre : queue SYCL

e est un objet de type Event de SYCL qui représente l'exécution de la commande
e.wait() permet d'attendre la fin de l'exécution