

hipSolverSP库外包项目

项目介绍

SOLVER库主要由两部分构成hipSOLVER库和rocSOLVER库，其中具体的算法实现在rocSOLVER库中，hipSOLVER库是对rocSOLVER库的接口封装，同时提供与cuSOLVER库相同风格的API，方便用户进行对已有的cuda程序进行移植。

与cuSOLVER库相比，hipSOLVER库缺失hipSolverSP（稀疏求解器），现计划将hipSolverSP模块进行外包。

hipSolverSP的需要开发的接口在[hipSOLVER外包API列表](#)文件中说明。

开发要求

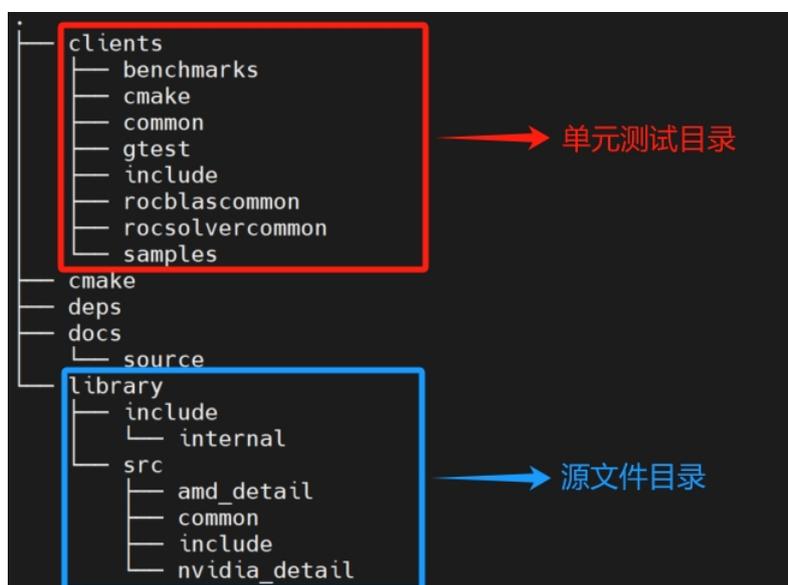
hipSolverSP模块在现有hipSOLVER库和rocSOLVER库的基础上进行开发，具体的算法功能在rocSOLVER库中实现，hipSOLVER库中对各API进行封装，新增API需要提供与cuSOLVER相同风格的接口，即接口名称的后缀相同，接口参数数量、类型相同（若是cuda中自定义类型，hipSOLVER中也需要新增对应的类型），具体的接口定义参考[cuSolver接口文档](#)，其他开发要求如下：

- 新增的API不可影响已有API的功能和性能。
- 新增API的声明和具体实现需新建文件，例如：新增API声明在新建的头文件hipsovlerSp.h中。
- 若需要对库中已有的公共文件进行修改，并在验收文档中进行特殊说明。
- 对于多种数据类型使用相同算法逻辑进行实现的情况，使用C++模板进行开发，减少重复代码，与已有API保持相同实现风格。
- **新增API开发完成后，需完成函数单元测试代码编写，并执行测试通过。**求解器类的API的单元测试程序需要集成到hipSOLVER和rocSOLVER的benchmark程序中，能够进行正确性测试和精度测试，精度验证采用求解结果带入原式进行验证的方式，并使用[SuiteSparse矩阵库](#)中的真实矩阵作为算例进行测试。对benchmark程序的修改和新增文件同样需要满足上述要求。

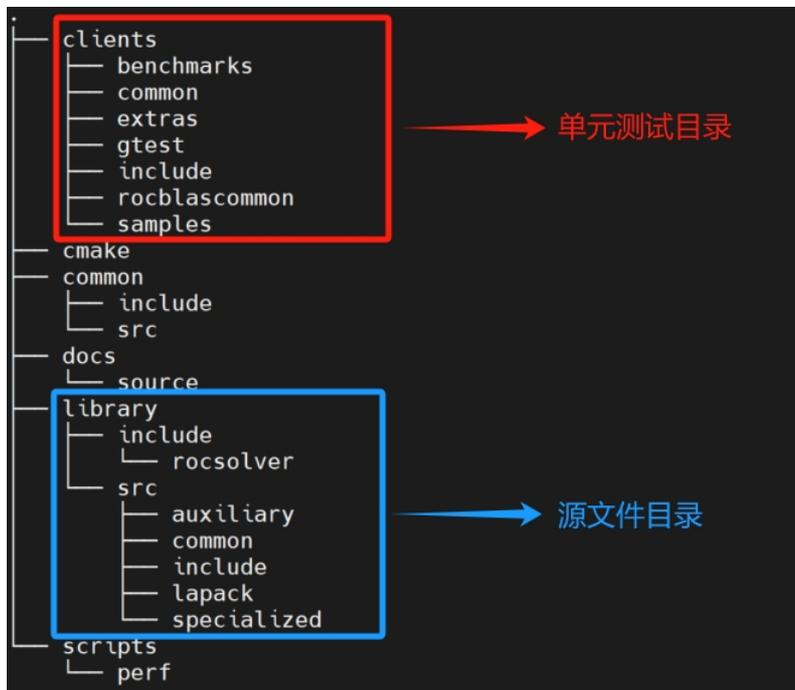
编码规范

- 目录结构规范

hipSOLVER库目录结构



rocSOLVER库目录结构



• 函数定义规范

◦ 函数定义需包含：

- 函数简介：一句话概括函数功能。
- 函数详细说明：详细介绍函数的功能，必要时借助公式进行说明。
- 参数介绍：逐个解释参数的含义，并标注为 入参 或 出参 或 同时具备。
- 函数声明：根据开发要求声明函数。

示例：

```

/*! @brief LACGV conjugates the complex vector x.
 *
 * @details
 * It conjugates the n entries of a complex vector x with increment incx.
 *
 * @param[in]
 * handle      roclblas_handle.
 * @param[in]
 * n           roclblas_int. n >= 0.\n
 *             The dimension of vector x.
 * @param[inout]
 * x           pointer to type. Array on the GPU of size at least n (size depends on the value of incx).\n
 *             On entry, the vector x.
 *             On exit, each entry is overwritten with its conjugate value.
 * @param[in]
 * incx       roclblas_int. incx != 0.\n
 *             The distance between two consecutive elements of x.
 *             If incx is negative, the elements of x are indexed in
 *             reverse order.
 *
 * *****/
ROCSOLVER_EXPORT roclblas_status rocsolver_clacgv(roclblas_handle handle,
                                                  const roclblas_int n,
                                                  roclblas_float_complex* x,
                                                  const roclblas_int incx);

```

• 单元测试规范

- 函数开发完成后，针对每一类求解器函数，都需要在单元测试目录增加对应的单元测试程序。
- 根据当前单元测试程序架构，新增各类稀疏求解器的测试程序，具体做法是：

对于每一类稀疏求解器，都需要在client/include目录下新增一个testing_xxxx.hpp测试文件，其中包含参数检查、精度验证、性能测试等测试函数，并将该测试程序加入到client/include/目录下的hipsolver_dispatcher.hpp或rocsolver_dispatcher.hpp文件中，并可以通过hipsolver-bench和rocsolver-bench测试程序对该接口进行测试。

验收标准

- 功能、正确性
 - 求解器API支持四种数据类型（实数单精度、实数双精度、复数单精度、复数双精度）。
 - 精度验证采用求解结果带入原式进行验证的方式，单精度误差 $<10^{-6}$ ，双精度误差 $<10^{-12}$ 。（精度验证方法需根据不同API进行针对性调研）
 - 将新增API运用到实际应用程序中，替换应用程序相同功能的接口，保证该应用程序正常运行，且运行结果正确。（“API—实际应用程序”的测试组合待确定）
- 性能
 - 求解器类API的性能指标以V100上cuda 11.2版本的性能作为参照，DCU1上的hipsolverSP的性能需要在80%覆盖率的算例中，性能达到cusolver对应API的60%；DCU2上的hipsolverSP的性能需要在80%覆盖率的算例中，性能达到cusolver对应API的80%；（KM-E待确定）。

考虑到不同求解器的算法和算例特征不同，可能需要针对不同API设计不同的算例集合，设置不同的验收标准。

ex: hipsolverSp<t>csrslvlu

LU分解求解稀疏线性方程组 $A \cdot x = b$ ， A 是CSR格式的 $n \times n$ 稀疏矩阵， b 是右端向量， x 是待求解的结果向量。在SuiteSparse稀疏矩阵集中选择20个不同特征（矩阵规模、非零元数量）的稀疏矩阵，并随机生成右端向量，组成测试算例，进行精度验证和性能测试。

精度标准：

将调用API求解出的 x 向量代入 $A \cdot x$ ，求出 b' ，误差 $error = \max(|b-b'|) / \text{norm}(b)$ ， $error$ 需要满足：单精度API小于 10^{-6} ，双精度小于 10^{-12} 。20个算例全部满足以上的精度标准才算通过精度验证。

性能标准：

针对每个测试算例连续运行10次，取性能的平均值作为性能结果，hipsolver库和cusolver库对应的API进行性能对比，20个算例中，有16个及以上算例达到cusolver库性能的80%及以上，则判断该API性能达标；否则性能不达标。

验收流程

- 采用阶段性验收的方式，每完成一类求解器API，即可进行该求解器的验收测试，然后进行代码集成。
- 验收材料
 - 各API的设计文档
 - hipSolverSP开发完成的hipSOLVER代码
 - hipSOLVER代码文件修改说明
 - 算法实现和优化文档
 - 单元测试文档

- 代码检查
 - 数学库开发组的人员负责对代码实现进行检查，结合算法实现文档确认两者是否匹配。
 - 检查单元测试程序的精度验证和性能测试方法是否正确并符合约定。
- 验收测试
 - 单元测试中，核心用例与基本用例一次性通过率100%，非关键性用例一次性通过率不低于95%。
 - 测试组人员结合[SuiteSparse矩阵库](#)中的真实矩阵设计测试用例，覆盖多种特征的矩阵算例，进行功能验证和性能测试，整理测试数据。
 - 将开发完成后的API运用到合适的应用程序中，测试精度和性能表现。

可能存在的问题

- hipSolverSP的API实现可能会依赖hipSparse库，功能、性能方面的实现和优化都会受影响
 - hipSparse库目前性能比较一般，hipSolverSP外包项目依赖hipSparseAPI时，所依赖的API优化实现放在hipSolverSP中，两个库就会有重复。
 - hipSparse库的优化计划或外包计划，直接影响hipSolverSP外包项目。