0) Задание состоит в том, чтобы расширить имеющуюся программу в 3м задании с использованием CUDA/OpenACC/DVMH.

1) Срок **начала** сдачи программы – не позднее **7 декабря,** срок **окончания** сдачи – **25 декабря**
2) Программа может быть написана

- с использованием MPI + CUDA
- с использованием MPI + OpenACC
- с использованием DVMH модели

Если выбирается более низкоуровневая модель CUDA, то программа должна быть оптимизирована достаточно хорошо. К директивным моделям требования к сдаче могут быть снижены.
3) Если используется MPI+CUDA, то программа должна собираться через makefile, в котором обязательно должны быть две переменные **ARCH**=sm_N (N = 35 / 60), обозначающая архитектуру ГПУ, и **HOST_COMP=mpicc,** обозначающая хост компилятор. Эти переменные должны использоваться как минимум для **nvcc.** Запрещается использовать возможности CUDA > cc 3.5.
4) Отчет о выполнении должен содержать в себе этап выполнения 3 задания, то есть распараллеливание на MPI, MPI + OpenMP и на кластере GPU с выбранной моделью.
5) В отчете должны содержаться все времена запусков задачи на том количестве процессоров, которое требуется. Также должны быть получены графики ускорения и эффективности, по отношению к **последовательной** (исходной!) программе без MPI / OpenMP.
*Опционально можно посчитать ускорения различных параллельных версий между собой.*
6) Программа на MPI + CUDA/OpenACC или DVMH должна работать **НЕ медленнее**, чем MPI, и MPI + OpenMP и тем более последовательная версия. Если количество данных не хватает для получения «хороших» цифр на GPU, следует увеличить исходные размеры массивов.
7) Отчет должен содержать пояснение по тем результатам, которые были получены – каков характер ускорений, эффективности, каковы причины такого поведения, если ожидаемые цифры не совпадают с реальными.
8) В отчете должно быть указано, каким образом производилась оценка корректности выполнения параллельных версий, в особенности MPI + CUDA/OpenACC.
9) В отчете должны содержаться не только общее время работы программы, а также времена всех параллельных циклов, времена инициализации и завершения работы программы, времена копирования данных с GPU на хост и обратно, времена коммуникационных обменов (если они асинхронные, то демонстрация того, что они не занимают времени) как в исходной, последовательной программе, так и в параллельной. Таким образом, таблица, содержащая времена запусков, должна содержать помимо общего времени, времена всех затрат на коммуникации и обмены между GPU, а также времена параллельных циклов.
10) Для сдачи необходимо прислать исходный код программы и готовый отчет по 3 заданию. Исходный код должен содержать исходную (не испорченную) версию программы, и параллельную.
11) **Запрещается** использование разделяемой памяти для реализации редукции. Использование разделяемой памяти где-либо требует обоснования в отчете.
12) Предпочтительнее использовать thrust в варианте MPI+CUDA.

Невыполнение **хотя бы одного** из этих пунктов приведет к дополнительной итерации сдачи 3го задания по реализации не выполненных пунктов.

PS: Производительность IBM Power 8 DP – 0.3 Tflops, Tesla P100 – 4.7 Tflops
Скорость памяти IBM Power 8 – for 1 TB RAM – 230 GB/s, Tesla P100 – 700 GB/s
Из этих соотношений ясно, что для ГПУ может быть получена достаточно хорошая программа при условии оптимальности кода.