

| status_dt | question | no | ans |
|-----------------|--|--------------------------------------|--|
| 1 2019-01-22 | Выберите варианты правильного ответа. Что будет выдано фрагментом программы: MPI_Init(&argc,&argv); MPI_Comm_size(MPI_COMM_WORLD, &size); MPI_Comm_rank(MPI_COMM_WORLD, &rank); tag=rank; x=rank; if (!rank){ MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD); MPI_Recv(&x, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &stat); printf(" %dn",x); } if (rank == 1) { MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD); MPI_Recv(&x, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &stat); } | 1 2 3 4 | 0 1 Значение не определено Программа завершится с ошибкой |
| 2 2019-01-22 | Что будет выдано при выполнении программы на 4-ех процессорах: int size, rank, tag, count; int x=0,y; MPI_Status stat; MPI_Init(&argc,&argv); MPI_Comm_size(MPI_COMM_WORLD, &size); MPI_Comm_rank(MPI_COMM_WORLD, &rank); count=rank; x=rank; MPI_Allreduce (&x,&y,count, MPI_INT, MPI_SUM, MPI_COMM_WORLD); if (!rank) printf(" %dn",x); MPI_Finalize(); | 1 2 3 4 5 6 7 8 | 0 1 2 3 5 6 Значение не определено Программа завершится с ошибкой |
| 3 2019-01-22 | Что будет выдано фрагментом программы: int size, rank, tag; int x; MPI_Status stat; MPI_Request request; MPI_Init(&argc,&argv); MPI_Comm_size(MPI_COMM_WORLD, &size); MPI_Comm_rank(MPI_COMM_WORLD, &rank); tag=rank; x=rank; if (!rank) MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD); if (rank == 1) { MPI_Irecv(&x, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, &stat,&request); printf(" %dn",x); } MPI_Finalize(); } | 1 2 3 4 5 6 | 1 0 0 или 1 Программа ничего не выдаст Программа может завершиться с ошибкой Программа завершится с ошибкой |
| 4 2019-01-22 | Что будет выдано при выполнении программы на 4-ех процессорах: #include "mpi.h" #include <stdio.h> #include <stdlib.h> int main (int argc, char *argv[]) { int size, rank, tag, count,i; int x[100],y[100]; MPI_Status stat; MPI_Init(&argc,&argv); MPI_Comm_size(MPI_COMM_WORLD, &size); MPI_Comm_rank(MPI_COMM_WORLD, &rank); count=rank; root = 1; if (rank==root) for (i=0;i<size; x[i+]=i*size); MPI_Scatter(x, count, MPI_INT, y, count,MPI_INT, root,MPI_COMM_WORLD); if (rank==size-1) printf(" %dn",y[0]); MPI_Finalize(); } | 1 2 3 4 5 6 | 0 4 8 12 Значение не определено Программа завершится с ошибкой – правильный ответ |
| 5 2019-01-22 | Отметьте правильные утверждения. Функция MPI_Recv завершится успешно при выполнении следующих условий: | 1 2 3 4 5 6 7 | Размер буфера должен быть равен размеру отправленного сообщения Буфер памяти должен быть достаточным для приема сообщения Тип элементов передаваемого и принимаемого сообщения должны совпадать Тип принятого сообщения может быть равен MPI_ANY_TYPE Сообщение может быть принято только от определенного процесса Сообщение может быть принято от любого процесса Сообщение может быть принято от любого процесса, входящего в коммуникатор, заданный в функции |
| 6 2019-01-22 | Что будет выдано при выполнении фрагмента программы: MPI_Init(&argc, &argv); MPI_Comm_size(MPI_COMM_WORLD, &size); MPI_Comm_rank(MPI_COMM_WORLD, &rank); if (rank == 0){ strcpy(message, "Hello"); MPI_Recv(message, 30, MPI_CHAR, 1, type, MPI_COMM_WORLD, &status); MPI_Send(message, 30, MPI_CHAR, 1, type, MPI_COMM_WORLD); }else if (rank == 1){ MPI_Recv(message, 30, MPI_CHAR, 0, type, MPI_COMM_WORLD, &status); MPI_Send(message, 30, MPI_CHAR, 0, type, MPI_COMM_WORLD); } printf("%sn", message); MPI_Finalize(); | 1 2 3 4 | Процесс 0 выведет в stdout Hello Процесс 1 выведет в stdout Hello Программа завершится с ошибкой Оба процесса выведут в stdout Hello |
| 7 2019-01-22 | Параллельна структура модели прогноза погоды ПЛАВ реализуется на основе использования подхода(подходов): | 1 2 3 4 | 1D-MPI декомпозиции по долготе 2D-MPI декомпозиции по широте 3D-MPI декомпозиции по широте OpenMP по долготе |

| status_dt | question | no | ans |
|------------------|---|----|---|
| | | 5 | OpenMP по широте |
| | | 6 | 1D-MPI декомпозиции по широте |
| | | 7 | 2D-MPI декомпозиции по долготе |
| | | 8 | 3D-MPI декомпозиции по долготе |
| 8 2019-01-22 | Что означает ансамблевый подход к прогнозированию погоды: | 1 | Реализация расчетов выполняется с одновременным использованием параллельных технологий MPI и OpenMP |
| | | 2 | Проведение расчета нескольких десятков долгосрочных прогнозов, которые затем усредняются |
| | | 3 | Расчет прогноза выполняется ансамблем параллельных процессов |
| 9 2019-01-22 | Задача Неймана для уравнения Лапласа | 1 | имеет единственное решение |
| | | 2 | является невырожденной, поскольку в однородном случае имеет только тривиальное решение |
| | | 3 | имеет бесконечно много решений при любых значениях граничного условия |
| | | 4 | является условно разрешимой |
| 10 2019-01-22 | Указать неверное утверждение. Метод скорейшего спуска | 1 | является одношаговым итерационным методом решения системы линейных алгебраических уравнений (СЛАУ) |
| | | 2 | сходится к точному решению СЛАУ с квадратной симметричной положительно определенной матрицей при любом начальном приближении |
| | | 3 | является методом вариационного типа |
| | | 4 | сходится быстрее метода сопряженных градиентов, так как является его обобщением |
| 11 2019-01-22 | Какие обычно достигаются проценты от теоретического пика в тестах производительности суперкомпьютеров HPCG и HPL? | 1 | HPCG – несколько % от пика, HPL – 70-80% и более |
| | | 2 | HPL – несколько % от пика, HPCG – 70-80% и более |
| | | 3 | У HPCG и HPL примерно одинаковые уровни в несколько % от пика из-за нехватки пропускной способности памяти |
| | | 4 | У HPCG и HPL примерно одинаковые уровни в районе 70-80% от пика благодаря высокой интенсивности вычислений |
| 12 2019-01-22 | Выберите правильный ответ. Подход DES к моделированию турбулентных течений реализует: | 1 | Моделируются турбулентные структуры всех масштабов, от крупных вихрей до мелких |
| | | 2 | Крупные вихри воспроизводятся численно, мелкие вихри моделируются |
| | | 3 | Турбулентные структуры всех масштабов воспроизводятся численно |
| | | 4 | Крупные оторвавшиеся вихри воспроизводятся численно, мелкие вихри моделируются, но вблизи твердых поверхностей моделируются вихри всех масштабов |
| 13 2019-01-22 | Выберите правильный ответ. Подход RANS к моделированию турбулентных течений реализует: | 1 | Моделируются турбулентные структуры всех масштабов, от крупных вихрей до мелких |
| | | 2 | Крупные вихри воспроизводятся численно, мелкие вихри моделируются |
| | | 3 | Турбулентные структуры всех масштабов воспроизводятся численно |
| | | 4 | Крупные оторвавшиеся вихри воспроизводятся численно, мелкие вихри моделируются, но вблизи твердых поверхностей моделируются вихри всех масштабов |
| 14 2019-01-22 | Выберите правильный ответ. Подход LES к моделированию турбулентных течений реализует | 1 | Моделируются турбулентные структуры всех масштабов, от крупных вихрей до мелких |
| | | 2 | Крупные вихри воспроизводятся численно, мелкие вихри моделируются |
| | | 3 | Турбулентные структуры всех масштабов воспроизводятся численно |
| | | 4 | Крупные оторвавшиеся вихри воспроизводятся численно, мелкие вихри моделируются, но вблизи твердых поверхностей моделируются вихри всех масштабов |
| 15 2019-01-22 | Выберите правильный ответ. Подход DNS к моделированию турбулентных течений реализует: | 1 | Моделируются турбулентные структуры всех масштабов, от крупных вихрей до мелких |

| status_dt | question | no | ans |
|------------------|--|----|---|
| | | 2 | Крупные вихри воспроизводятся численно, мелкие вихри моделируются |
| | | 3 | Турбулентные структуры всех масштабов воспроизводятся численно |
| | | 4 | Крупные оторвавшиеся вихри воспроизводятся численно, мелкие вихри моделируются, но вблизи твердых поверхностей моделируются вихри всех масштабов |
| 16 2019-01-22 | Каковы современные тенденции в развитии вычислительных устройств – процессоров и ускорителей? | 1 | пиковая производительность растет значительно быстрее, чем пропускная способность памяти |
| | | 2 | пропускная способность памяти растет значительно быстрее, чем пиковая производительность |
| | | 3 | пропускная способность памяти и производительность растут примерно одинаково, чтобы устройства были хорошо сбалансированы |
| | | 4 | рост производительности и пропускной способности практически прекратился из-за достижения предела увеличения тактовой частоты |
| 17 2019-01-22 | Чем графический процессор GPU отличается от центрального процессора CPU? Выберите правильные варианты | 1 | У GPU существенно выше пропускная способность памяти |
| | | 2 | У CPU существенно выше пропускная способность памяти |
| | | 3 | У GPU обычно намного больше памяти, чем у CPU |
| | | 4 | GPU работает так же, как и CPU, но у GPU намного больше ядер, а сами ядра проще устроены – меньше регистров, проще конвейер. |
| | | 5 | GPU реализует принципиально иной тип параллелизма, ядра CPU и ядра GPU – это совсем разные вещи |
| 18 2019-01-22 | <div style="text-align: left;"> Что такое конвекция Рэлея – Бенара? | 1 | вынужденная конвекция, вызванная внешним перепадом давления |
| | | 2 | естественная конвекция, вызванная горизонтальным градиентом температуры |
| | | 3 | вынужденная конвекция, вызванная колебательным движением твердой поверхности |
| | | 4 | естественная конвекция, вызванная вертикальным градиентом температуры |
| 19 2019-01-22 | Чему равна ширина бисекционной плоскости 2D-тора 4x3? | 1 | 12 |
| | | 2 | 6 |
| | | 3 | 4 |
| | | 4 | 3 |
| 20 2019-01-22 | Как обеспечивается выполнение длительного крупного расчета на суперкомпьютере? | 1 | Путем периодической записи на файловую систему полного дампа данных процесса, находящихся в оперативной памяти, чтобы работа процесса могла быть возобновлена после остановки. |
| | | 2 | Путем периодической записи на файловую систему минимального набора данных, необходимых для восстановления расчета. С целью экономии дискового пространства хранится только один последний набор данных. |
| | | 3 | Путем периодической поочередной записи в два различных набора файлов минимального набора данных, необходимых для восстановления расчета. |
| | | 4 | Путем периодической записи каждый раз в новый набор файлов минимального набора данных, необходимых для восстановления расчета. |
| 21 2019-01-22 | Какое количество коммутаторов необходимо для построения высокоскоростной сети Mellanox Infiniband EDR с половиной от максимально возможной бисекционной пропускной способностью (half bisection) для суперкомпьютера из 96 вычислительных узлов, если для этого используются только 36-портовые коммутаторы? | 1 | 3 |
| | | 2 | 4 |
| | | 3 | 5 |
| | | 4 | 6 |
| | | 5 | 7 |
| | | 6 | 8 |
| 22 2019-01-22 | Какое количество коммутаторов необходимо для построения высокоскоростной сети Intel OmniPath с половиной от | 1 | 3 |

| status_dt | максимально возможной бисекционной пропускной способностью (half bisection) для суперкомпьютера из 128 вычислительных узлов, если для этого используются только 48-портовые коммутаторы? | no | ans |
|------------------|---|----|---|
| | | 2 | 4 |
| | | 3 | 5 |
| | | 4 | 6 |
| | | 5 | 7 |
| | | 6 | 8 |
| 23 2019-01-22 | Какое количество нерегулярных обращений к памяти произойдет при последовательной обработке (см. код ниже) одним программным потоком всего графа $G(V =n, E =m)$, который хранится в формате CRS? for (int u = 0; u < G->n; u++) { for (int j = G->rowsIndices[u]; j < rowsIndices[u+1]; j++) { const int v = G->endV[j]; // обработка ребра u->v } | 1 | n |
| 24 2019-01-22 | Перечислите эффекты, которые обычно возникают при обработке разреженной матрицы на вычислительной системе после применения алгоритма Reverse Cuthill-McKee. | 2 | m |
| | | 3 | n+m |
| | | 4 | 2n |
| | | 5 | 2n+m |
| 25 2019-01-22 | Выберете все верные утверждения относительно следующего кода, при условии, что st1, st2 отличны от потока по умолчанию, а ядро меняет массив arr1: строка1- cudaMemcpyAsync(arr1, arr2, count, cudaMemcpyHostToDevice, st1); строка2- kernel<<<count / 256, 256, 0, st2 >>>(arr1, arr3, count); строка3- cudaMemcpyAsync(arr2, arr1, count, cudaMemcpyDeviceToHost, st1); | 1 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1 , полученные после завершения функции в строке1 |
| | | 2 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1 , полученные после завершения функции в строке2 |
| | | 3 | ядро выполнится параллельно с копированием в строке1 и строке3 |
| | | 4 | ядро выполнится только после завершения копирования в строке1 |
| | | 5 | копирование в строке3 выполнится только после копирования в строке1 |
| 26 2019-01-22 | Выберете все верные утверждения относительно следующего кода, при условии, что st1, st2 отличны от потока по умолчанию, а ядро меняет массив arr1: строка1- cudaMemcpy (arr1, arr2, count, cudaMemcpyHostToDevice); строка2- kernel<<<count / 256, 256, 0, st2 >>>(arr1, arr3, count); строка3- cudaMemcpyAsync(arr2, arr1, count, cudaMemcpyDeviceToHost, st1); | 1 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке1 |
| | | 2 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке2 |
| | | 3 | ядро выполнится параллельно с копированием в строке3 |
| | | 4 | ядро выполнится только после завершения копирования в строке1 |
| | | 5 | копирование в строке3 выполнится только после копирования в строке1 |
| 27 2019-01-22 | Выберете все верные утверждения относительно следующего кода, при условии, что st1, st2 отличны от потока по умолчанию, а ядро меняет массив arr1: строка1- cudaMemcpyAsync (arr1, arr2, count, cudaMemcpyHostToDevice); строка2- kernel<<<count / 256, 256, 0, st2 >>>(arr1, arr3, count); строка3- cudaMemcpyAsync(arr2, arr1, count, cudaMemcpyDeviceToHost, st1); | 1 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке1 |
| | | 2 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке2 |
| | | 3 | ядро выполнится параллельно с копированием в строке3 |
| | | 4 | ядро выполнится только после завершения копирования в строке1 |
| | | 5 | копирование в строке3 выполнится только после копирования в строке1 |
| 28 2019-01-22 | Выберете все верные утверждения относительно следующего кода, при условии, что st1, st2 отличны от потока по умолчанию, а ядро меняет массив arr1: строка1- cudaMemcpyAsync(arr1, arr2, count, cudaMemcpyHostToDevice, st1); строка2- kernel<<<count / 256, 256, 0, st2 >>>(arr1, arr3, count); строка3- cudaMemcpyAsync(arr2, arr1, count, cudaMemcpyDeviceToHost, st1); | 1 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке1 |

| status_dt | question | no | ans |
|------------------|--|----|--|
| | | 2 | Значения элементов массива arr2 после завершения функции в строке3 будут совпадать со значениями элементов массива arr1, полученные после завершения функции в строке2 |
| | | 3 | ядро выполнится параллельно с копированиями в строке1 и строке3 |
| | | 4 | ядро выполнится только после завершения копирования в строке1 |
| | | 5 | копирование в строке3 выполнится только после копирования в строке1 |
| 29 2019-01-22 | Отметьте все верные факты про вызов данной функции cudaMemcpyAsync(array1, array2, count * 1024, cudaMemcpyDeviceToDevice); | 1 | Происходит копирование данных с ГПУ на ЦПУ |
| | | 2 | Происходит копирование данных с ЦПУ на ГПУ |
| | | 3 | Происходит копирование данных с ГПУ на ГПУ |
| | | 4 | Происходит копирование данных из массива array2 в массив array1 |
| | | 5 | Происходит копирование данных из массива array1 в массив array2 |
| | | 6 | Копируется count байт |
| | | 7 | Копируется count килобайт |
| | | 8 | Операция асинхронна |
| 30 2019-01-22 | Отметьте все верные факты про технологию CUDA: | 1 | Это программно-аппаратная архитектура параллельных вычислений, позволяющая значительно ускорить код с использованием GPU NVidia |
| | | 2 | CUDA является расширением стандартных языков C/C++ |
| | | 3 | CUDA является новым языком программирования на базе C/C++ |
| | | 4 | CUDA является расширением стандартного языка Fortran |
| | | 5 | CUDA является расширением стандартных языков Java/C#/Python |
| | | 6 | Это программно-аппаратная архитектура параллельных вычислений, позволяющая значительно ускорить код с использованием любых GPU. |
| 31 2019-01-22 | Отметьте все верные факты в отношении GPU Nvidia: | 1 | Скорость глобальной памяти ГПУ выше, чем RAM память ЦПУ |
| | | 2 | Скорость вычислений ГПУ в двойной точности такая же, как и скорость вычислений в одинарной точности |
| | | 3 | Размер кэш памяти ГПУ соответствует размеру кэша ЦПУ последних поколений |
| | | 4 | ГПУ является сопроцессором |
| 32 2019-01-22 | Максимальное число процессов, которые можно запустить на одном узле Blue Gene/P: | 1 | 1 |
| | | 2 | 2 |
| | | 3 | 4 |
| | | 4 | 8 |
| | | 5 | 16 |
| | | 6 | 32 |
| | | 7 | Число процессов не ограничено |
| 33 2019-01-22 | На суперкомпьютере Blue Gene/P можно использовать следующие технологии параллельного программирования: | 1 | только MPI |
| | | 2 | MPI+OpenMP |
| | | 3 | MPI+Pthreads |
| | | 4 | MPI+CUDA |
| 34 2019-01-22 | Какое максимальное число потоков можно использовать в параллельной программе для Blue Gene/P при запуске на 1024 узлах : | 1 | 32 |
| | | 2 | 512 |
| | | 3 | 1024 |
| | | 4 | 2048 |
| | | 5 | 4096 |

| status_dt | question | no | ans |
|------------------|--|----|--|
| | | 6 | 3072 |
| | | 7 | 8096 |
| | | 8 | 128 |
| | | 9 | Число потоков неограничено |
| 35 2019-01-22 | Пересылка данных в программе для Blue Gene/P, выполненная с использованием функции MPI_Bcast, будет выполнена с использованием сети «дерево» при выполнении следующих условий: | 1 | Объем передаваемых данных должен превышать 512 байт |
| | | 2 | Для любых объемов передаваемых данных |
| | | 3 | Программа должна быть запущена на числе процессоров, кратных 512 |
| | | 4 | Для коммуникатора MPI_COMM_WORLD |
| | | 5 | Для любых коммуникаторов |
| 36 2019-01-22 | Какой объем оперативной памяти, доступной процессу, достигается при запуске параллельных программ на Blue Gene/P в режиме VN: | 1 | 512 MB |
| | | 2 | 1024 MB |
| | | 3 | 2 GB |
| | | 4 | 4 GB |
| | | 5 | 1024 GB |
| 37 2019-01-22 | Каков порядок вычислительной мощности алгоритма перемножения плотных квадратных матриц? | 1 | O(1) |
| | | 2 | O(n) |
| | | 3 | O(n^2) |
| | | 4 | O(n^3) |
| | | 5 | правильного ответа нет |
| 38 2019-01-22 | Каким параллелизмом обладает фрагмент программы: for(i=1; i<=n; ++i) for(j=1; j<=m; ++j) A[i][j] = (A[i-1][j] * A[i][j-1])/2; | 1 | конечным |
| | | 2 | координатным |
| | | 3 | скошенным |
| | | 4 | другим |
| | | 5 | не обладает |
| 39 2019-01-22 | Отметьте верные утверждения про локальность данных: | 1 | свойство локальности данных характеризует качество алгоритма |
| | | 2 | свойство локальности данных характеризует качество программной реализации |
| | | 3 | пространственная локальность показывает, насколько близко друг к другу расположены различные данные |
| | | 4 | временная локальность показывает, насколько часто происходят обращения к одним и тем же данным |
| | | 5 | тест Random Access обладает хорошей пространственной локальностью |
| 40 2019-01-22 | Какой будет архитектура большинства вновь создаваемых суперкомпьютеров? Почему? | 1 | Традиционные монолитные, типа Blue Gene - они проверены практикой и хорошо себя зарекомендовали. |
| | | 2 | Однородные двухсокетные кластеры - они дешевые, простые и понятные. |
| | | 3 | Гибридные, содержащие процессоры разных архитектур - это модно и интересно. |
| 41 2019-01-22 | Какое свойство процессора наиболее важно для вычислительных задач? | 1 | Главное - максимальная теоретическая производительность в Гфлопс-ах. |
| | | 2 | Иметь как можно больше однородных ядер - чтобы можно было эффективно распараллеливать алгоритм. |
| | | 3 | Обеспечивать широкие возможности SIMD-ификации кода - использование векторизации дает большой реальный эффект. |
| | | 4 | Иметь высокую пропускную способность подсистемы памяти - память должна успевать предоставлять процессору данные, над которыми он работает. |
| | | 5 | это зависит от конкретного приложения - у всех разные требования. |
| 42 2019-01-22 | Максимальное число потоков, которые можно запустить на одном узле Blue Gene/P: | 1 | 2 |
| | | 2 | 4 |
| | | 3 | 8 |
| | | 4 | 16 |

| status_dt | question | no | ans |
|------------------|---|----|---|
| | | 5 | 32 |
| | | 6 | Число потоков не ограничено |
| 43 2019-01-22 | Суперкомпьютер Blue Gene/P является: | 1 | многопроцессорным кластером с однородной архитектурой |
| | | 2 | массивно-параллельной вычислительной системой |
| | | 3 | высокопроизводительной гибридной вычислительной системой |
| 44 2019-01-22 | С каким количеством процессоров непосредственно связан каждый вычислительный узел Blue Gene/ коммуникационной сетью «решетка» : | 1 | 4 |
| | | 2 | 6 |
| | | 3 | 8 |
| | | 4 | 16 |
| | | 5 | 128 |
| 45 2019-01-22 | Коммуникационная сеть «дерево» Blue Gene/P используется для: | 1 | выполнения 2-ух точечных передач большого объема |
| | | 2 | выполнения коллективных операций MPI при соблюдении определенных условий |
| | | 3 | выполнения любых коллективных операций MPI |
| | | 4 | для доступа к файловой системе |
| 46 2019-01-22 | Максимальный объем доступной памяти достигается при запуске параллельных программ на Blue Gene/P в режиме: | 1 | DUAL |
| | | 2 | SMP |
| | | 3 | VN |
| | | 4 | не зависит от режима запуска |
| 47 2019-01-22 | Почему явные итерационные методы успешны для решения задачи на собственные значения на суперкомпьютере BlueGene/P? | 1 | Неявные итерационные методы показывают более низкую точность вычислений. |
| | | 2 | Для прямых методов нельзя эффективно использовать параллельное быстрое преобразование Фурье. |
| | | 3 | Существуют эффективные методы распараллеливания перемножения матрицы на вектор с учетом параллельного быстрого преобразования Фурье. |