

Численная модель мелкой воды: использование графических процессоров NVIDIA CUDA*

Т.А. Дьяконова¹, А.В. Хоперсков¹, С.С. Храпов¹

Волгоградский государственный университет¹

Обсуждаются особенности реализации программного комплекса для численного моделирования динамики поверхностных вод в приближении мелкой воды на основе параллельных технологий с использованием графических процессоров NVIDIA CUDA. В основе численного алгоритма лежит комбинированный Лагранжево-Эйлеровый метод (CSPH-TVD). Основное внимание в работе уделяется особенностям параллельной реализации на графических процессорах линейки Tesla: C2070, K20, K40, K80. Иерархическая система пространственных сеток разных масштабов дает возможность эффективно использовать вычислительные ресурсы и ускоряет расчеты при решении различных задач затопления территорий.

Ключевые слова: численное моделирование, параллельные технологии, графические процессоры, уравнения мелкой воды

1. Введение

Современная вычислительная техника позволяет решать самые различные задачи гидрологии с учетом реального рельефа местности $b(x, y)$ и всех значимых физических факторов для больших территорий [1]. Особо выделим проблему управления гидрологическим режимом пойменных ландшафтов во время весеннего паводка на крупных реках [2]. Решение этой задачи требует максимально эффективных численных моделей на основе параллельных технологий [3]. Для эколого-экономического управления территорией, например, Волго-Ахтубинской поймы, необходимо решать задачу оптимизации гидрографа для конкретных условий в каждом году, и делать экспертные заключения для различных режимов работы десятков гидросооружений в пойме и по строительству новых [4]. Каждая такая задача требует проведения сотен численных экспериментов по прямому гидродинамическому моделированию весеннего гидрологического режима для территории площадью от 2 тыс. до 20 тыс. кв. км.

Наша практика использования больших суперкомпьютеров (в частности, на вычислительных ресурсах Суперкомпьютерного комплекса МГУ [5]) для проведения массовых гидродинамических расчетов выявила ряд проблем, связанных с необходимостью проводить за короткий период большое число расчетов и передач через сеть больших объемов данных для последующей обработки и анализа. Оперативность проведения расчетов и обработки данных моделирования являются важнейшим фактором применения такого рода моделей в практической деятельности. Дополнительной проблемой является визуализация проведенных расчетов, что представляется общей трудностью для очень высокопроизводительных машин [6]. Переход на вычислительные ресурсы класса «персональные суперкомпьютеры» на основе GPU позволяет отчасти решить указанные проблемы. В работе рассмотрены результаты создания пакета программ для параллельных расчетов на вычислительных узлах C2070, K20, K40, K80.

*Исследование выполнено при финансовой поддержке РФФИ и Администрации Волгоградской области в рамках научных проектов № 16-07-01037, № 15-02-06204, № 15-45-02655 р_а. Работа выполнена с использованием ресурсов суперкомпьютерного комплекса МГУ имени М.В. Ломоносова.

2. Математическая и численная модели

2.1. Основные уравнения

Для описания динамики поверхностных вод будем использовать уравнения Сен-Венана (приближение мелкой воды) в виде:

$$\frac{\partial H}{\partial t} + \frac{\partial HU_x}{\partial x} + \frac{\partial HU_y}{\partial y} = \sigma, \quad (1)$$

$$\frac{\partial U_x}{\partial t} + U_x \frac{\partial U_x}{\partial x} + U_y \frac{\partial U_x}{\partial y} = -g \frac{\partial \eta}{\partial x} + F_x + \frac{\sigma}{H}(V_x - U_x), \quad (2)$$

$$\frac{\partial U_y}{\partial t} + U_x \frac{\partial U_y}{\partial x} + U_y \frac{\partial U_y}{\partial y} = -g \frac{\partial \eta}{\partial y} + F_y + \frac{\sigma}{H}(V_y - U_y), \quad (3)$$

где H — глубина воды, U_x, U_y — горизонтальные компоненты вектора скорости воды \vec{U} , усредненные по вертикальной координате, σ — поверхностная плотность источников и стоков воды [м/с], g — ускорение свободного падения, $\eta(x, y, t) = H(x, y, t) + b(x, y)$ — уровень свободной поверхности воды, V_x, V_y — горизонтальные компоненты вектора скорости воды в источнике или стоке \vec{V} , F_x, F_y — горизонтальные компоненты вектора внешних и внутренних сил \vec{F} , действующих на слой воды. Суммарная плотность сил

$$\vec{F} = \vec{F}^{fric} + \vec{F}^{visc} + \vec{F}^{cor} + \vec{F}^{wind}, \quad (4)$$

где $\vec{F}^{fric} = -\frac{\lambda}{2}\vec{U}|\vec{U}|$ — сила придонного трения, $\lambda = \frac{2gn_M^2}{H^{4/3}}$ — коэффициент гидравлического сопротивления, n_M — коэффициент шероховатости дна по Маннингу, $\vec{F}^{visc} = \nu(\frac{\partial^2 U_x}{\partial x^2} + \frac{\partial^2 U_y}{\partial y^2})$ — сила внутреннего (вязкого) трения между слоями жидкости, ν — коэффициент кинематической турбулентной вязкости, $\vec{F}^{cor} = 2[\vec{U} \times \vec{\Omega}]$ — сила Кориолиса, $\vec{\Omega}$ — угловая скорость вращения Земли, $\vec{f}^{wind} = C_a \frac{\rho_a}{\rho H} (\vec{W} - \vec{U})|\vec{W} - \vec{U}|$ — сила ветра, C_a — параметр, зависящий от состояния водной поверхности, ρ_a и ρ — плотность воздуха и воды соответственно, \vec{W} — вектор скорости ветра в горизонтальном направлении.

Модель (1)–(3) учитывает следующие факторы [7]: неоднородный рельеф местности $b(x, y)$; взаимодействие потока с подстилающей поверхностью, причем свойства этой поверхности неоднородны; вращение Земли; взаимодействие потока жидкости с ветровым переносом приземного слоя атмосферы; источники, связанные с работой гидротехнических сооружений и осадками; фильтрация в грунтовый слой и испарение; внутреннее трение, обусловленное турбулентным переносом.

При построении вычислительных моделей в работе [3] предложен подход так называемого Co-design, учитывающего при разработке программного кода архитектуру суперкомпьютера. Повышение эффективности основано на максимизации независимых вычислений с учетом особенностей численных алгоритмов решения уравнений Сен-Венана на неоднородном рельефе местности.

2.2. Система сеток и матрицы цифрового рельефа местности

Использование декартовой системы координат и равномерной сетки $\Delta x_i = \Delta y_j = \Delta x = \Delta y$, несмотря на сложную неоднородность рельефа местности в виде русел крупных рек (Волга, Ахтуба) и мелких ериков и протоков, позволяет упростить процесс декомпозиции расчетной области и повысить эффективность CSPH-TVD метода (см. п. 2.3) при сквозном расчете границ вода–сухое дно.

Уменьшение характерных размеров ячеек ℓ имеет ограничение, обусловленное применением уравнений мелкой воды (1)–(3), в силу которых величина ℓ не может быть меньше

глубины жидкости H . Из-за сильной нестационарности данной задачи максимальная глубина жидкости в расчетных ячейках может варьироваться от 10 м на затопляемых участках суши до 30 м в русле Волги. Поэтому для применимости модели мелкой воды необходимо использовать крупномасштабные сетки при моделировании на глубоких русловых участках и мелкомасштабные на затопляемых участках суши.

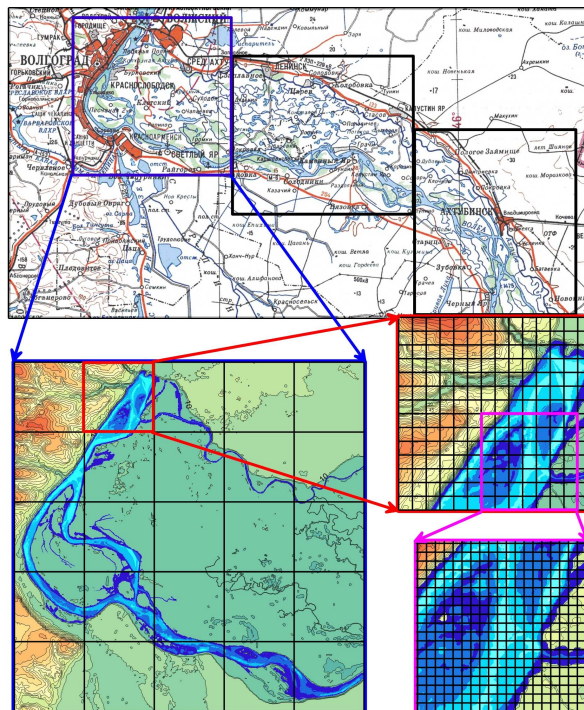


Рис. 1: Иерархическая система сеток для территории междуречья Волги и Ахтубы.

На рисунке 1 показана структура сеток, позволяющая существенно экономить ресурсы при решении гидрологических задач, когда изучаемые процессы (протекание жидкости) происходят только в небольшом числе ячеек. Иерархическая система сеток (ИСС) последовательно описывает динамику поверхностных вод от мелких масштабов до наиболее крупных с неравномерно распределенными источниками. Для наиболее важных зон применяется технология zoom-in [8] без обратного влияния и zoom-in с учетом обратного влияния на динамику воды на мелкомасштабной сетке со стороны расчета на крупномасштабной сетке.

2.3. Численная гидродинамическая схема

На рисунке 2 изображена вычислительная схема метода CSPH-TVD [9, 10] ($h = \Delta x = \Delta y$ — пространственный шаг, $0 < K < 1$ — число Куранта, $U_p = \max[|U_x^n + \text{sign}(F_x)\sqrt{hF_x}|, |U_y^n + \text{sign}(F_y)\sqrt{hF_y}|]$, $U_s = \max(|U_x^n| + \sqrt{gH^n}, |U_y^n| + \sqrt{gH^n})$). Можно выделить четыре основных этапа вычислений на одном временном слое t_n . На I и II этапе применяется Лагранжев подход, а на III и IV — Эйлеров подход. Первоначально вычисляются источниковые слагаемые \bar{Q} , определяющие действие внешних и внутренних сил, а также связанные с наличием величины σ . Здесь же вычисляется шаг интегрирования по времени τ_n . Затем вычисляются изменения величины \bar{q} (II этап) под действием рассчитанных на первом этапе факторов, что позволяет определить смещение лагранжевых частиц \bar{r} внутри ячеек. На данном этапе для продвижения по времени использовалась схема «предиктор-корректор» II порядка точности. После вычисления потоков массы и импульса через границы эйлеровых ячеек с использованием приближенного решения задачи Римана (III этап) на последнем IV этапе рассчитываем итоговые значения величины \bar{q} на следующем временном слое t_{n+1} , и возвращаем частицы в исходное положение в центры эйлеровых ячеек (x_i, y_j) .

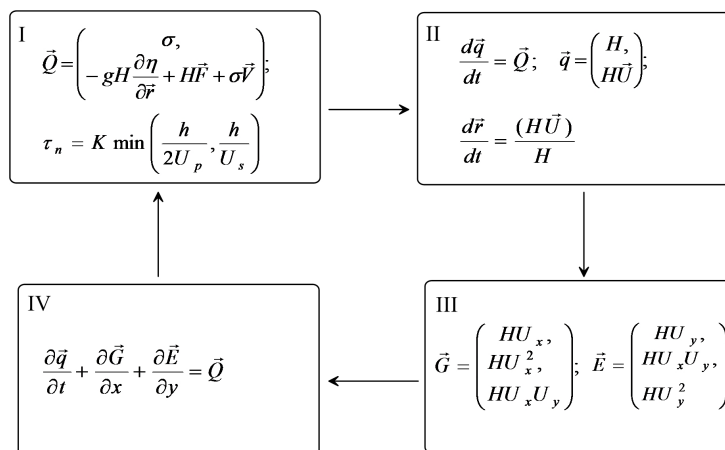


Рис. 2: Основные этапы вычислительной схемы для решения уравнений мелкой воды

Среди положительных свойств численной схемы выделим второй порядок точности по пространственной и временной координатам, консервативность, хорошую сбалансированность, возможность сквозного счета на динамических границах вода–сухое дно без регуляризации.

3. Параллельная реализация численного алгоритма

Для распараллеливания численной схемы CSPH-TVD применена технология CUDA, позволяющая эффективно реализовать подход иерархической системы сеток (см. пп. 2.1). Это обусловлено тем, что блоки ИСС являются своеобразным аналогом потоковых блоков CUDA, на которых основано выполнение CUDA-ядер (рис. 3). Отличительной особенностью реализуемого подхода является использование динамического параллелизма CUDA, что позволяет для наиболее важных областей проводить детализированный расчет гидродинамических течений на мелкомасштабных сетках с помощью дополнительных нитей.

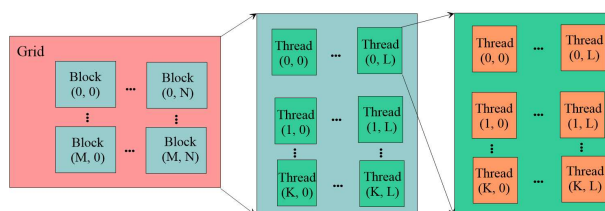


Рис. 3: Иерархия нитей на GPU с учетом динамического распараллеливания

Описанный в п. 2 вычислительный алгоритм распараллелен с использованием технологии OpenMP–CUDA. На рисунке 4 изображена диаграмма деятельности основных этапов численного алгоритма, где введены следующие обозначения для вычислительных ядер CUDA: K1 — kernel_Index_block предназначено для определения блоков, заполненных водой; K2 — kernel_forces_predictor вычисляет силы в момент времени t_n на лагранжевом этапе; K3 — kernel_dt вычисляет шаг интегрирования по времени в зависимости от текущих параметров течения; K4 — kernel_SPH_predictor рассчитывает новые положения частиц и интегральные характеристики в момент времени $t_{n+1/2}$; K5 — kernel_forces_corrector определяет силы на промежуточном временном слое $t_{n+1/2}$; K6 — kernel_SPH_corrector рассчитывает положения частиц и интегральные характеристики для следующего временного слоя t_{n+1} ; K7 — kernel_TVD_flux для времени t_{n+1} вычисляет потоки физических величин через границы ячеек; K8 — kernel_Final определяет итоговые гидродинамические

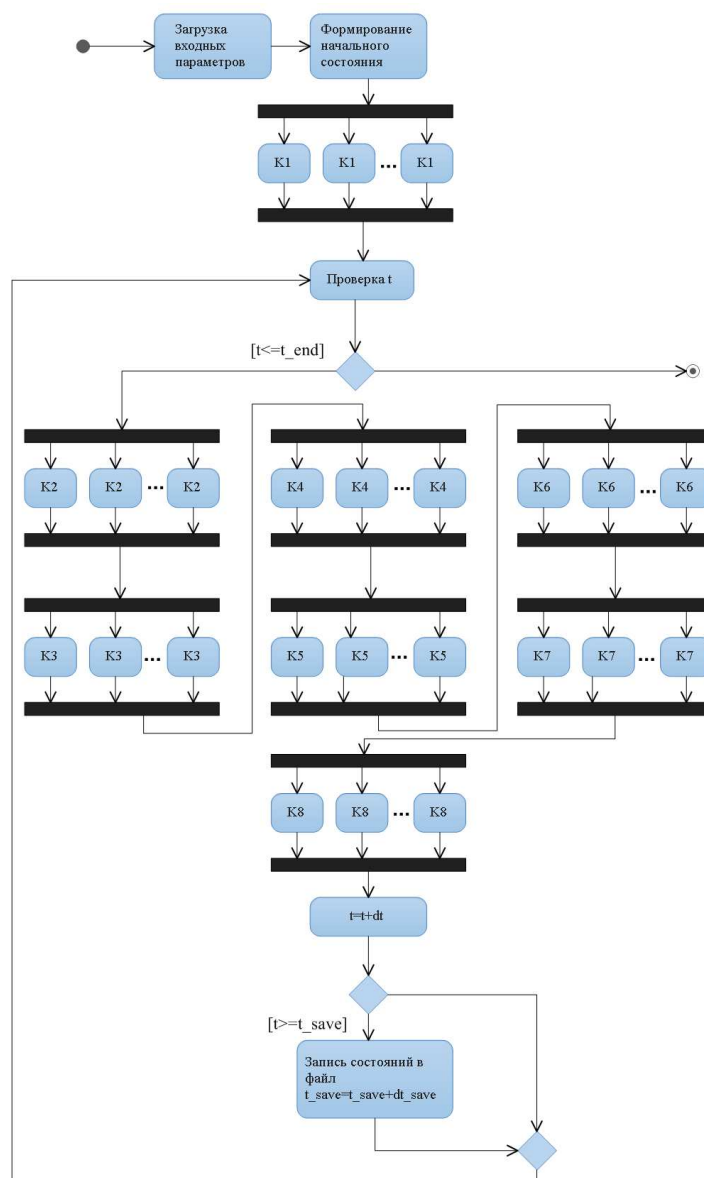


Рис. 4: Диаграмма деятельности расчетного модуля

параметры в момент времени t_{n+1} .

Данная диаграмма отражает особенности CSPH-TVD. Предложенное разделение вычислительного алгоритма на 8 CUDA-ядер позволяет оптимально использовать ресурсы GPU.

На рисунке 5 показаны доли времени выполнения соответствующих CUDA-ядер, приходящиеся на выполнение основных этапов расчетной схемы. Почти 60 % приходится на вычисление потоков TVD-этапа kernel_TVD_flux (см. рис. 5). Это обусловлено тем, что на этом этапе решается задача Римана. С учетом вычисления сил kernel_forces_predictor и kernel_forces_corrector вклад от TVD-этапа увеличивается и является доминирующим. Лагранжев этап в сумме (kernel_SPH_predictor и kernel_SPH_corrector) превносит чуть более 8 %. Таким образом, незначительное увеличение времени расчета на SPH-этапе существенно улучшает свойства численной схемы.

На рисунке 6 приведен фрагмент программы, реализующий заключительный этап расчета (CUDA-ядро K8) с учетом проверки на наличие жидкости в расчетном CUDA-блоке размером 16×16 ячеек. Если в запускаемом блоке жидкость отсутствует (параметр Index_block

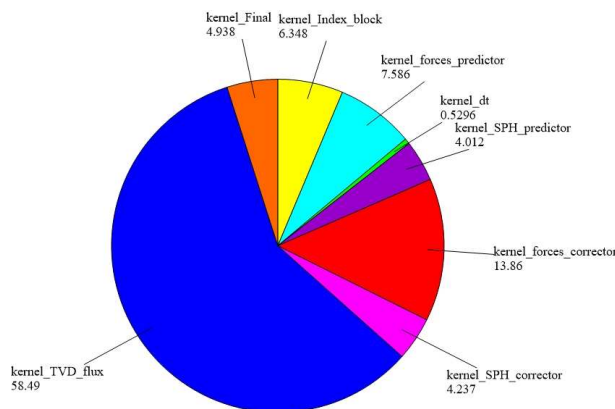


Рис. 5: Вклады различных этапов численной схемы CSPH-TVD на одном временном слое

равен нулю), то для всех нитей в блоке вычисления пропускаются. Аналогичный подход реализован и для других CUDA-ядер (K2, K4, K5, K6 и K7).

```

__global__ void kernel_Final(double *H, double2 *HV, double *Ht,
double2 *HVt, double *Fh, double2 *Fv, int2 *Index_block,
double tau){
    int ib=blockIdx.x+blockIdx.y*gridDim.x;
    int x = threadIdx.x + blockIdx.x * blockDim.x, y = threadIdx.y +
        blockIdx.y * blockDim.y;
    int ind = x + y * blockDim.x * gridDim.x;
    if(Index_block[ind].x > 0 || Index_block[ind].y > 0){
        double dt_h = tau/dd.hp, Eps=dd.Eps, ht, Flux_h;
        double2 hv=make_double2(0,0);
        ht = Ht[ind]; Flux_h=Fh[ind];
        if( ht>Eps || fabs(Flux_h)>Eps){
            ht = dev_h(ht + dt_h*Flux_h);
            hv.x = dev_hv(HVt[ind].x + dt_h*Fv[ind].x,ht);
            hv.y = dev_hv(HVt[ind].y + dt_h*Fv[ind].y,ht);
        }
        H[ind]=ht; HV[ind]=hv; Ht[ind]=0; HVt[ind]=make_double2(0,0);
    }else {Ht[ind]=0; HVt[ind] = make_double2(0,0);}
}
    
```

Рис. 6: Фрагмент кода, реализующий CUDA-ядро K8.

Определение параметра Index_block происходит в CUDA-ядре K1 (рис. 7). Переменная Index_block является структурой целого типа (int2) с двумя полями Index_block.x и Index_block.y. Значение параметра Index_block.x > 0 указывает на наличие воды хотя бы в одной ячейке рассматриваемого CUDA-блока. Условие Index_block.y > 0 означает, что хотя бы в одной из ячеек, граничащих с данным CUDA-блоком, имеется вода. Необходимость проверки наличия воды в приграничных ячейках соседних CUDA-блоков обусловлена потоками жидкости через границы CUDA-блоков на эйлеровом этапе в CUDA-ядре K7.

Для гибридных систем вида CPU+n×GPU целесообразно использовать двухуровневую схему распараллеливания (рис. 8 а). Для обеспечения быстрого обмена данными на различных GPU применяется технология Direct Access. Последнюю можно использовать только для GPU, подключенных к шинам PCI Express, находящимся под управлением одного CPU (рис. 8б).

4. Сравнение эффективности для различных GPU

На рисунке 9 а для четырех графических карт Tesla приведена диаграмма производительности программы. В процентах указаны значения загрузки GPU. Профилирование про-

```

__global__ void kernel_Index_block(int2 *Index_block, int *Index_Q,
double *H){
    __shared__ int2 Sij[ithbx*ithby];
    int ind_thb = threadIdx.x + ithbx*threadIdx.y;
    int ib=blockIdx.x+blockIdx.y*gridDim.x;
    int x = threadIdx.x + blockIdx.x*blockDim.x, y = threadIdx.y +
        blockIdx.y*blockDim.y;
    int tid, xx, yy, i, j, si, sj, Ni, Nj, isi, jsj, m;
    double2 D=make_double2(0,0); int2 iD=make_int2(0,0);
    if(threadIdx.x == 0){Ni=1; si=-1;}
    else if(threadIdx.x == ithbx-1){Ni=1; si=1;}
    else {Ni=0; si=0;}
    if(threadIdx.y == 0){Nj=1; sj=-1;}
    else if(threadIdx.y == ithby-1){Nj=1; sj=1;}
    else {Nj=0; sj=0;}
    for(i=0; i<=Ni; i++){
        isi = i*si;
        if(x==0 && isi<=-1) xx = x;
        else if(x==dd.Nx-1 && isi>=1) xx = x;
        else xx = x + isi;
        if(i==0) m=Nj; else m=0;
        for(j=0; j<=m; j++){
            jsj = j*sj;
            if(y==0 && jsj<=-1) yy = y;
            else if(y==dd.Ny-1 && jsj>=1) yy = y;
            else yy = y + jsj;
            tid = xx + yy*dd.Nx;
            if(i>0 || j>0) {D.y += H[tid]; iD.y += Index_Q[tid]; }
            if(i==0 & j==0) {D.x += H[tid]; iD.x += Index_Q[tid]; }
        }
    }
    if(D.x>dd.Eps || iD.x>0) Sij[ind_thb].x=1;
    else Sij[ind_thb].x=0;
    if(D.y>dd.Eps || iD.y>0) Sij[ind_thb].y=1;
    else Sij[ind_thb].y=0;
    __syncthreads();
    int k = ithbx*ithby/2;
    while(k != 0){
        if(ind_thb < k) {Sij[ind_thb].x += Sij[ind_thb+k].x;
            Sij[ind_thb].y += Sij[ind_thb+k].y; }
        __syncthreads();
        k /= 2; }
    if(ind_thb == 0) {Index_block[ib].x = Sij[0].x;
        Index_block[ib].y = Sij[0].y; }
}

```

Рис. 7: Фрагмент кода, реализующий CUDA-ядро K1.

граммы проводилось при помощи NVIDIA Parallel Nsight. Уменьшение процента загрузки при переходе на более производительные GPU с большим числом скалярных ядер связано с числом расчетных ячеек 1072×1248 (при $\Delta x = \Delta y = 50$ м). При переходе к сеткам с шагом (10 – 25) м процент загрузки увеличивается. Однако использование шага < 10 м может нарушать приближение мелкой воды. Таким образом, для моделирования затоплений на территориях площадью порядка 10 тыс. кв. км с учетом фактора загрузки целесообразно использовать персональные суперкомпьютеры с несколькими GPU.

На рисунке 9б приведена диаграмма распределения регистров памяти на поток для CUDA-ядер. Параметры программы подобраны с учетом особенностей используемых графических ускорителей таким образом, чтобы избежать спилинга регистров.

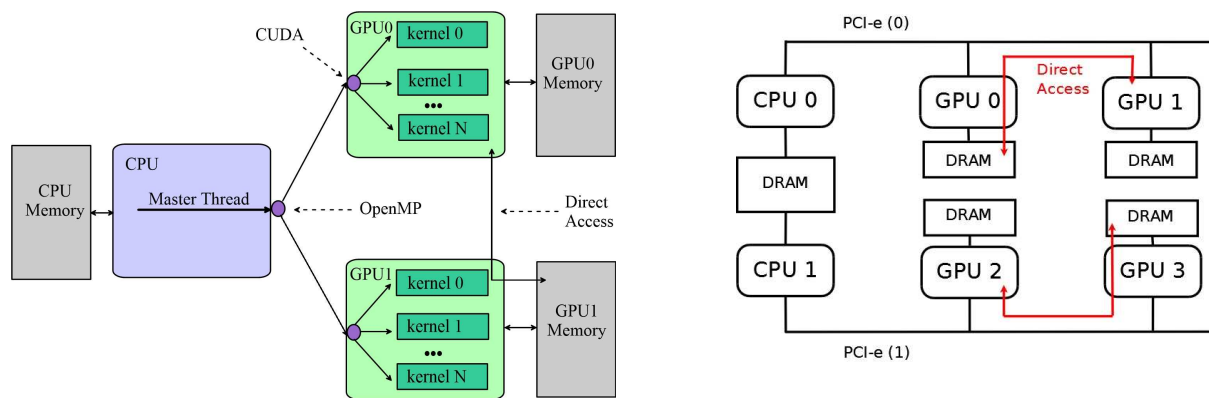


Рис. 8: а) Двухуровневая схема распараллеливания OpenMP – CUDA б) Архитектура 2×CPU+4×GPU

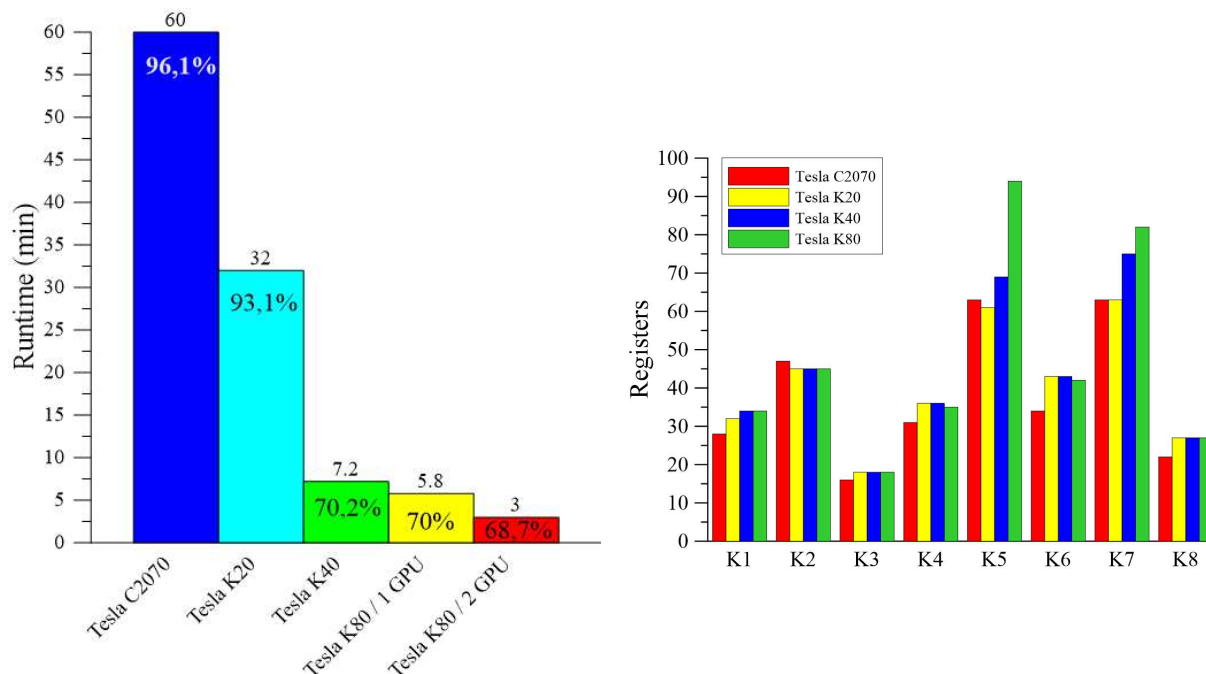


Рис. 9: а) Время расчета динамики затопления северной части ВАП на протяжении 10 часов на различных GPU для сетки 1072×1248. б) Диаграмма распределения регистров памяти по CUDA-ядрам для различных GPU.

5. Результаты моделирования

В качестве примера расчетов рассмотрим задачу аварийного сброса с суммарным расходом $100000 \text{ м}^3/\text{с}$. Моделирование проводилось для северной части Волго-Атубинской поймы размером $53600 \text{ м} \times 62400 \text{ м}$ ($\Delta x = \Delta y = 50 \text{ м}$). На рисунке 10 представлено гидрологическое состояние территории через 5 часов после начала моделирования. Средняя скорость течения в пойме 5 м/с , средняя глубина – 6 м . Соотношение воды и суши на всей области моделирования $\sim 35\%$, в максимуме достигая $\sim 60\%$. Для таких задач целесообразно использовать изложенный в п. 3 подход, основанный на проверке наличия жидкости в расчетных CUDA-блоках, который позволяет производить вычисления только в блоках с водой.

Результаты сравнения наших численных моделей с данными наблюдений за динамикой уровней воды на гидропостах [1] и площади затопления в мае 2011 год [11] показали удовлетворительное согласие.

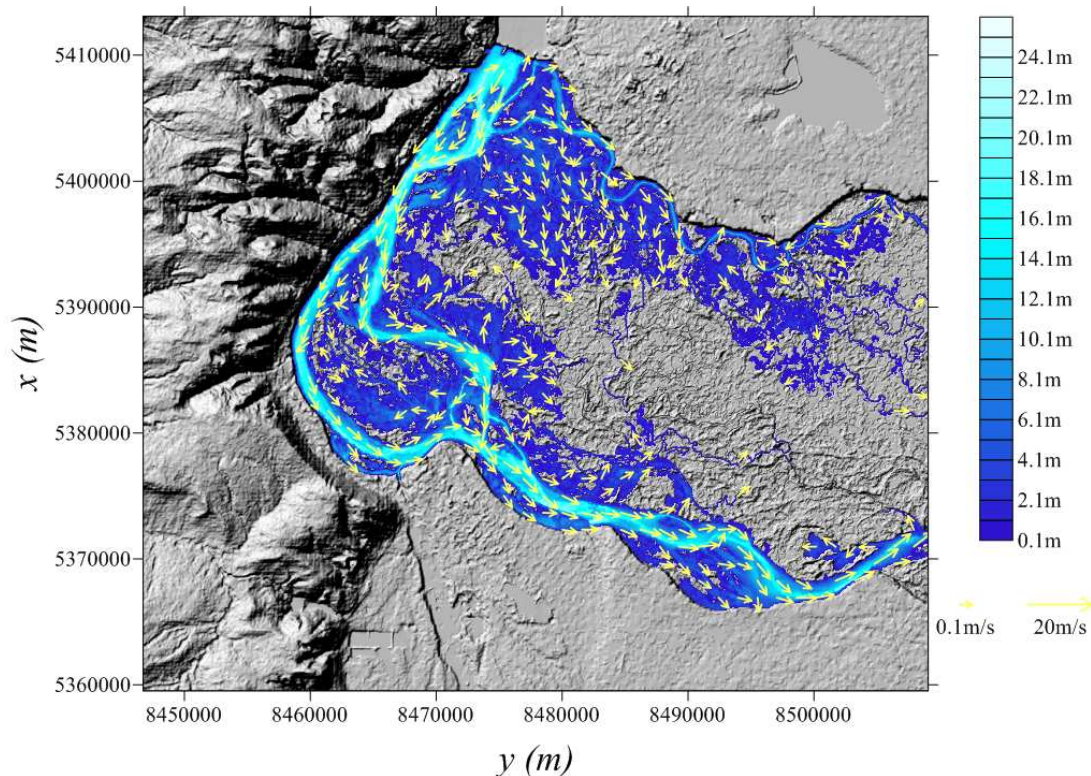


Рис. 10: Результаты вычисления структуры течения на двух сетках. Показаны распределения глубин (цветом) и поля скоростей (стрелки). Хорошо видны вихревые структуры.

6. Заключение

В работе рассмотрены некоторые особенности параллельной реализации численных моделей на основе уравнений Сен-Венана в случае, когда площадь затопления изменяется со временем в широких пределах. Например, для Волго-Ахтубинской поймы площадь под водой в период весеннего паводка может увеличиваться в десятки раз. Для повышения эффективности таких расчетов применен гибридный подход распараллеливания OpenMP-CUDA с оптимизацией на основе выделения CUDA-блоков и контроля наличия воды в расчетных ячейках внутри и на границах блоков. Переход с последовательной версии на параллельную ускорил расчет в 100 – 1200 раз в зависимости от используемого GPU.

Литература

1. Писарев А.В., Храпов С.С., Агафонникова Е.О., Хоперсков А.В. Численная модель динамики поверхностных вод в русле Волги: оценка коэффициента шероховатости // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2013. № 1. С. 114–130.
2. Воронин А.А., Елисеева М.В., Храпов С.С., Писарев А.В., Хоперсков А.В. Задача управления гидрологическим режимом в эколого-экономической системе «Волжская ГЭС – Волго-Ахтубинская пойма». Ч. 2. Синтез системы управления // Проблемы управления. 2012. № 6. С. 19–25.

3. Glinskiy B., Kulikov I., Snytnikov A., Romanenko A., Chernykh I., Vshivkov V. Co-design of Parallel Numerical Methods for Plasma Physics and Astrophysics // *Supercomputing frontiers and innovations*. 2014. Vol. 1. P. 88–98.
 4. Воронин А.А., Васильченко А.А., Писарева М.В., Писарев А.В., Хоперсков А.В., Храпов С.С., Подципкина Ю.Е. Проектирование системы эколого-экономического управления территорией Волго-Ахтубинской поймы на основе гидродинамического и геоинформационного моделирования // *Управление большими системами*. 2015. Выпуск 55. С. 79–102.
 5. Sadovnichy V., Tikhonravov A., Voevodin Vl., Opanasenko V. "Lomonosov": Supercomputing at Moscow State University. In *Contemporary High Performance Computing: From Petascale toward Exascale* // Chapman & Hall/CRC Computational Science. Boca Raton, USA, CRC Press. 2013. P. 283–307.
 6. Moreland K., Larsen M., Childs H. Visualization for Exascale: Portable Performance is Critical // *Supercomputing Frontiers and Innovations*. 2015. Vol. 2. No. 3. P. 67–75.
 7. Дьяконова Т.А., Писарев А.В., Хоперсков А.В., Храпов С.С. Математическая модель динамики поверхностных вод // *Вестник Волгоградского государственного университета. Серия 1: Математика. Физика*. 2014. № 1 (20). С. 35–44.
 8. Bonoli Silvia, Mayer Lucio, Kazantzidis Stelios, Madau Piero, Bellovary Jillian, Governato Fabio Black hole starvation and bulge evolution in a Milky Way-like galaxy // *Monthly Notices of the Royal Astronomical Society*. 2016. Vol. 459. P. 2603–2617.
 9. Khrapov S., Pisarev A., Kobelev I., Zhumaliev A., Agafonnikova E., Losev A., Khoperskov A. The Numerical Simulation of Shallow Water: Estimation of the Roughness Coefficient on the Flood Stage // *Advances in Mechanical Engineering*. 2013. Vol. 2013. Article ID 787016. 11 p.
 10. Храпов С.С., Хоперсков А.В., Кузьмин Н.М., Писарев А.В., Кобелев И.А. Численная схема для моделирования динамики поверхностных вод на основе комбинированного SPH-TVD подхода // *Вычислительные методы и программирование*. 2011. Т. 12. № 2. С. 282–297.
 11. Писарев А.В., Храпов С.С., Воронин А.А., Дьяконова Т.А., Циркова Е.А. Особенности динамики затопления Волго-Ахтубинской поймы в зависимости от режимов испарения и инфильтрации // *Вестник Волгоградского государственного университета. Серия 1. Математика. Физика*. 2012. № 1 (16). С. 36–41.
-

Numerical model of shallow water: the use of GPUs NVIDIA CUDA

T.A. Dyakonova¹, A.V. Khoperskov¹, S.S. Khrapov¹
Volgograd State University¹

Features of the software package for the numerical simulation of the shallow water dynamics are discussed. We use parallel technologies for graphic processors NVIDIA CUDA. Combined Lagrangian-Euler method (CSPH-TVD) is the basis of our numerical algorithm. We considered characteristics of parallel implementation on GPUs Tesla range: C2070, K20, K40, K80. The hierarchical system of spatial grids of different scales makes it possible to efficiently use computing resources and speeds up the calculations in solving various problems of flooding areas.

Keywords: numerical simulation, parallel technology, graphics processors, shallow water equations

References

1. Pisarev A.V., Khrapov S.S., Agafonnikova E.O., Khoperskov A.V. Chislennaya model' dinamiki poverkhnostnykh vod v rusle Volgi: otsenka koeffitsienta sherokhovatosti [Numerical model of shallow water dynamics in the channel of the Volga: estimation of roughness] // Vestnik Udmurtskogo universiteta. Matematika. Mekhanika. Komp'yuternye nauki [Bulletin of Udmurt University. Mathematics, Mechanics]. 2013. No. 1, P. 114–130.
2. Voronin A.A., Eliseeva M.V., Khrapov S.S., Pisarev A.V., Khoperskov A.V. Zadacha upravleniya gidrologicheskim rezhimom v ekologo-ekonomicheskoy sisteme «Volzhskaya GES – Volgo-Akhtubinskaya poyma». Ch. 2. Sintez sistemy upravleniya [The Regimen Control Task in The Eco-Economic System «Volzhskaya Hydroelectric Power Station – the Volga-Akhtuba Floodplain». II. Synthesis of Control System] // Problemy upravleniya [Control Sciences]. 2012. No. 6. P. 19–25.
3. Glinskiy B., Kulikov I., Snytnikov A., Romanenko A., Chernykh I., Vshivkov V. Co-design of Parallel Numerical Methods for Plasma Physics and Astrophysics // Supercomputing frontiers and innovations. 2014. Vol. 1. P. 88–98.
4. Voronin A., Vasilchenko A., Pisareva M., Pisarev A., Khoperskov A., Khrapov S., Podshipkova J. Proektirovanie sistemy ekologo-ekonomicheskogo upravleniya territoriy Volgo-Akhtubinskoy poymy na osnove gidrodinamicheskogo i geoinformatsionnogo modelirovaniya [Designing a system for ecological-economical management of the Volga-Akhtuba floodplain on basis of hydrodynamic and geoinformational simulation] // Upravlenie Bol'shimi Sistemami. 2015. 55. P. 79–102
5. Sadovnichy V., Tikhonravov A., Voevodin V.I., Opanasenko V. "Lomonosov": Supercomputing at Moscow State University. In Contemporary High Performance Computing: From Petascale toward Exascale // Chapman & Hall/CRC Computational Science. Boca Raton, USA, CRC Press. 2013. P. 283–307.
6. Moreland K., Larsen M., Childs H. Visualization for Exascale: Portable Performance is Critical // Supercomputing Frontiers and Innovations. 2015. Vol. 2, No. 3, P. 67–75.
7. Dyakonova T.A., Pisarev A.V., Khoperskov A.V., Khrapov S.S. Matematicheskaya model' dinamiki poverkhnostnykh vod [Mathematical model of surface water dynamics] // Vestnik

- Volgogradskogo gosudarstvennogo universiteta. Seriya 1: Matematika. Fizika [Science Journal of VolSU. Mathematics. Physics]. 2014. 1 (20). P. 35–44.
8. Bonoli Silvia, Mayer Lucio, Kazantzidis Stelios, Madau Piero, Bellovary Jillian, Governato Fabio Black hole starvation and bulge evolution in a Milky Way-like galaxy // Monthly Notices of the Royal Astronomical Society. 2016. Vol. 459, P. 2603–2617.
 9. Khrapov S., Pisarev A., Kobelev I., Zhumaliev A., Agafonnikova E., Losev A., Khoperskov A. The Numerical Simulation of Shallow Water: Estimation of the Roughness Coefficient on the Flood Stage // Advances in Mechanical Engineering. 2013. Vol. 2013. Article ID 787016. 11 p.
 10. Khrapov S.S., Khoperskov A.V., Kuz'min N.M., Pisarev A.V., Kobelev I.A. Chislennaya skhema dlya modelirovaniya dinamiki poverkhnostnykh vod na osnove kombinirovannogo SPH-TVD podkhoda [A numerical scheme for simulating the dynamics of surface water on the basis of the combined SPH-TVD approach] // Vychislitel'nye metody i programmirovaniye [Numerical Methods And Programming]. 2011. Vol. 12. No. 2, P. 282–297.
 11. Pisarev A.V., Khrapov S.S., Voronin A.A., Dyakonova T.A., Tsyrkova E.A. [The role of infiltration and evaporation in the flooding dynamics of the volga-akhtuba floodplain] // Vestnik Volgogradskogo Gosudarstvennogo Universiteta. Seriya 1, Matematika. Fizika [Science Journal of VolSU. Mathematics. Physics]. 2012. No. 1 (16). P. 36–41.