Жоғары өнімді есептеу жүйелерінде Java пайдаланып 3D жылу өткізгіштік теңдеуін шешуге арналған параллельді есептеуді ұйымдастыру[1]

Б. Мәткерім, Д. Ж. Ахмед-Заки
Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
E-mail: bazargulmm@gmail.com

Аннотация

Бұл жұмыста үш өлшемді жылу өткізгіштік теңдеуін mpiJava технологиясын пайдаланып есептеуге арналған параллельді алгоритмді өңдеу технологиясы зерттеліп іске асырылды, сонымен қатар деректерді әр түрлі деңгейде декомпозициялаумен параллельді есептеу ұйымдастырылды. Алынған жуық шешімді аналитикалық шешіммен салыстыру арқылы жеткілікті дәлдіктегі жуық шешімге қол жеткізгеніміз белгілі болды.

1. Есептің қойлымы

Бұл мақалада үш өлшемді жылу өткізгіштік теңдеуін Java тілінде параллельді есептелуін қарастырамыз, қарастырылатын есептің қойлымы келесідей.

Айталық, $\Omega = [0,T] \times K\{0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1,\}$ кубында төмендегі есеп берілсін:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(\phi(x,y,z)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(\phi(x,y,z)\frac{\partial u}{\partial y}) + \frac{\partial}{\partial z}(\phi(x,y,z)\frac{\partial u}{\partial z}) + f(x,y,z) \tag{1}$$

бұнда $u = u(t,x,y,z)$ — кубтың t уақыттағы (x,y,z) нүктесінің температурасы;
$\phi(x,y,z)$ — жылу өткізгіштік коэффициенті [1].
$f(x,y,z)$ — жылу көзінің тығыздығы

Есептің бастапқы шарты

$$u(0,x,y,z) = \varphi(0,x,y,z) \tag{2}$$

Шекаралық шарттары:

$$u\big|_{\Gamma} = 0 \tag{3}$$

Мұндағы Г- Ω кубтың беттері. Осы Г - да және $t = 0$ – де (2) – (3) шарттарын және Ω-ның ішінде (1) теңдеуін қанағаттандыратын $u = u(x,y,z,t)$ функциясын анықтауымыз керек. Сандық әдістер арқылы (1) – (3) есебінің жуық шешімін анықтаймыз [2]. Есебтің шешімі u(t,x,y,z) кезкелген $0 \leq t \leq T$ уақыттағы кубтың әрбір (x,y,z) нүктесіндегі (x,y,z) температурасын береді. Кубтың шекаралық нүктелерінде 0 деген тұрақты

[1]Работа выполнена при поддержке грантового финансирования научно-технических программ и проектов Комитетом науки МОН РК, грант № 0729/ГФ, 2012г.-2014г.

температуралар берілген, ал бастапқы t=0 жағдайда әр (x,y,z) нүктесінің температурасын $\varphi(0,x,y,z)$ арқылы береміз. Мысалы $t=0.2$ уақытта $(x=0.3, y=0.5, z=0,2)$ нүктесіннің температурасы $u(0.2,0.3,0.5,0.2)$ арқылы анықталады.

Бұл есепте егер

$$\phi(x,y,z) = x^2 + y^2 + z^2$$

$$f(x,y,z) = e^{-bt}[3\pi^2 \sin ax \sin ay \sin az K(x,y,z) - 2\pi x \cos ax \sin ay \sin az - 2\pi y \sin ax \cos ay \sin az - 2\pi z \sin ax \sin ay \cos az - 3\pi^2 \sin ax \sin ay \sin az]$$

$$\varphi(0,x,y,z) = \sin ax \sin ay \sin az$$

$a=\pi$, $b=3\pi^2$ болған жағдайда аналитикалық дәл шешімі $u(t,x,y,z) = e^{-bt}\sin\pi x$ $\sin\pi y$ $\sin\pi z$, ал біз $u(t,x,y,z)$ функциясының жуық шешімін анықтаймыз.

2. Жылу өткізгіштік теңдеудің шешу әдісі

Берілген үш өлшемді жылу өткізгіштік теңдеуінің айқындалған схемасын құрамыз, ол үшін алдымен Ω - да айырымдық тор енгіземіз.

$$\omega_{\Delta s\tau} = \left\{ \begin{array}{l} (x_i, y_j, z_k, t_w) : x_i = i\Delta s, i = 0,1,...,N+1; \Delta s = \frac{1}{N+1}, \\ y_j = j\Delta s, j = 0,1,...,N+1; \Delta s = \frac{1}{N+1}, \\ z_k = z\Delta s, z = 0,1,...,N+1; \Delta s = \frac{1}{N+1}, \\ t_n = n\tau, n = 0,1,...,M; \tau = \frac{1}{M} \end{array} \right\}$$

сонда $\Omega_{\Delta s\tau} = \{(t_w, x_i, y_j, z_k)\}$ тор облысын аламыз, ол да

$$\Gamma_{\Delta s\tau} = \left\{ \begin{array}{l} (x_0, y_j, z_k, t_n), (x_{N+1}, y_j, z_k, t_n), (x_i, y_0, z_k, t_n), (x_i, y_{N+1}, z_k, t_n), (x_i, y_j, z_0, t_n), \\ (x_i, y_j, z_{N+1}, t_n), (x_i, y_j, z_k, t_n), i,j,k = \overline{0, N+1}; n = \overline{1, M} \end{array} \right\}$$

шекаралық және

$$\overline{\Omega}_{\Delta s} = \{(x_i, y_j, z_k, t_n), i,j,k = \overline{1,N}; n = \overline{1, M-1}\}$$

ішкі тор облыстарынан тұрады яғни $\Omega_{\Delta t\tau} = \Gamma_{\Delta t\tau} + \overline{\Omega}_{\Delta s\tau}$.

$\Omega_{\Delta t\tau}$ обылсының әрбір (x_i, y_j, z_k, t_n) нүктесінде айқындалған схемамен (1) – (3) есебін келесі айырымдық есебімен жуықтатамыз:

$$\begin{array}{l} \frac{u^{n+1}_{i,j,k} - u^n_{i,j,k}}{\tau} = \frac{\phi_{i+\frac{1}{2},j,k}(u^n_{i+1,j,k} - u^n_{i,j,k})}{(\Delta s)^2} - \frac{\phi_{i-\frac{1}{2},j,k}(u^n_{i,j,k} - u^n_{i-1,j,k})}{(\Delta s)^2} + \\ \frac{\phi_{i,j+\frac{1}{2},k}(u^n_{i,j+1,k} - u^n_{i,j,k})}{(\Delta s)^2} - \frac{\phi_{i,j-\frac{1}{2},k}(u^n_{i,j,k} - u^n_{i,j-1,k})}{(\Delta s)^2} + \\ \frac{\phi_{i,j,k+\frac{1}{2}}(u^n_{i,j,k+1} - u^n_{i,j,k})}{(\Delta s)^2} - \frac{\phi_{i,j,k-\frac{1}{2}}(u^n_{i,j,k} - u^n_{i,j,k-1})}{(\Delta s)^2} + f_{i,j,k}; \end{array} \tag{4}$$

$(i,j,k = 1,2,\ldots,N; n = 0,1,\ldots,M-1)$

Жоғарыдағы жылу коэффициенттері төмендегідей анықталады

$$\phi_{i+\frac{1}{2},j,k} = \frac{\phi_{i+1,j,k} + \phi_{i,j,k}}{2}; \quad \phi_{i-\frac{1}{2},j,k} = \frac{\phi_{i,j,k} + \phi_{i-1,j,k}}{2}; \quad \phi_{i,j+\frac{1}{2},k} = \frac{\phi_{i,j+1,k} + \phi_{i,j,k}}{2};$$

$$\phi_{i,j-\frac{1}{2},k} = \frac{\phi_{i,j,k} + \phi_{i,j-1,k}}{2}; \quad \phi_{i,j,k+\frac{1}{2}} = \frac{\phi_{i,j,k+1} + \phi_{i,j,k}}{2}; \quad \phi_{i,j,k-\frac{1}{2}} = \frac{\phi_{i,j,k} + \phi_{i,j,k-1}}{2}.$$

Бастапқы және шекаралық шарттардың айырымдық теңдеуі

$$u^0_{i,j,k} = \varphi_{i,j,k} \tag{5}$$

$$u^n_{i,j,k}\big|_\Gamma = 0 \tag{6}$$

(4)-(6) – айқындалған схема, оның реті $O(\tau + \Delta s^2)$. Схема шартты орнықты схема. Орнықтылық шарты $\tau \leq \Delta s^2$. Ал оның шешімі төмендегі айқын формуламен өрнектеледі:

$$\begin{aligned} u^{n+1}_{i,j,k} = \frac{\tau}{\Delta s^2}(&\phi_{i+\frac{1}{2},j,k}(u^n_{i+1,j,k} - u^n_{i,j,k}) - \phi_{i-\frac{1}{2},j,k}(u^n_{i,j,k} - u^n_{i-1,j,k}) \\ &+\phi_{i,j+\frac{1}{2},k}(u^n_{i,j+1,k} - u^n_{i,j,k}) - \phi_{i,j-\frac{1}{2},k}(u^n_{i,j,k} - u^n_{i,j-1,k}) + \phi_{i,j,k+\frac{1}{2}}(u^n_{i,j,k+1} - u^n_{i,j,k}) \\ &-\phi_{i,j,k-\frac{1}{2}}(u^n_{i,j,k} - u^n_{i,j,k-1})) + u^n_{i,j,k} + f_{i,j,k} \end{aligned}$$

(i,j,k=1,2,...,N; n=0,1,...,M-1)

$$u^0_{i,j,k} = \varphi_{i,j,k}(\mathrm{i,j,k} = \overline{1,N;}\ ), u^n_{i,j,k}\big|_{\Gamma_{\Delta s\tau}} = 0$$

3. Java тілінде параллельді есептеу

Java тілінің көрнекті қасиеттері оның объектіге бағытталатындығы, үлестірімділігі, платформаға тәуелсіздігі және қауыпсыздығы болып саналады. C және Fortran тілінде MPI стандарттарын пайдаланып параллельді есептеу кеңінен таралған, бірақ Java тілінің жоғарыда аталған ерекшеліктері оның параллельді есептеуді пайдаланып қолданбалы қосымшаларды өңдеуде үлкен сұранысқа ие болып отыр.

Java тілінде берілген есептті параллелдеу үшін біз mpijava бағдарламалар кітапханасын пайдаландық. Бұл бағдарламалар кітапхананың негізгі мақсаты Java бағдарламашыларына хабарламаларды айырбастау интерфесін(MPI) пайдалану арқылы параллельді бағдарламалар жазуға мүмкіндік туғызу. MPI(Message Passing Interface) C/C++, Fortran (MPICH, LAM, т.б.) тілдеріне арналған параллельді бағдарламалауға интерфейсімен кеңінен еркін таралған. mpiJava ортасы Java тіліне MPI бағдарламалық кітапханасындағы іске асырылатын барлық мүмкіндіктерін JNI(Java native methods interfaces) интерфейсінің көмегімен байлайды(binding), бағдарламалық қабаттар төмендегі сүретте бейнеленгендей [3]:

Әдетте есепті параллельді бағдарламалаудың екі түрі болады біріншісі есептің деректерін параллельдеу, ал екіншісі есептің өзін параллелдеу. Біз бірінші түрін пайдаланамыз, оның қасиеттері төмендегідей:

- Бір операция массивтің бірнеше элементінде бірдей орындалады. Бұл массивтің фрагменттері векторлық процессорда немесе параллельді машинадағы әр түрлі процессорда өңделеді.
- өңделетін деректерді бір бағдарлама басқарады.
- Есептеу кеңістігінің аты глабальды болады.

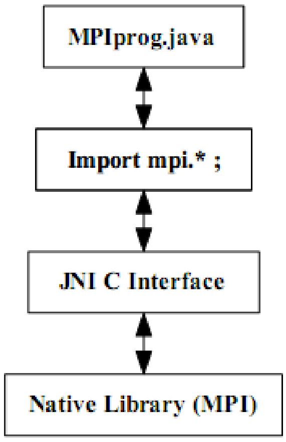


Рис. 1: Бағдарламаның қабаттары

- Массивтің элементіне жүргізілетін параллельді операциялар осы бағдарламаның барлық қолжетімді процессорларында бір уақытта орындалады.

Есепті параллельді есептеу үшін есептеу облысын декомпозициялау қажет. Біз есептеуді бір өлшемді және үш өлшемді декомпозициялау арқылы іске асырдық(1-сурет).

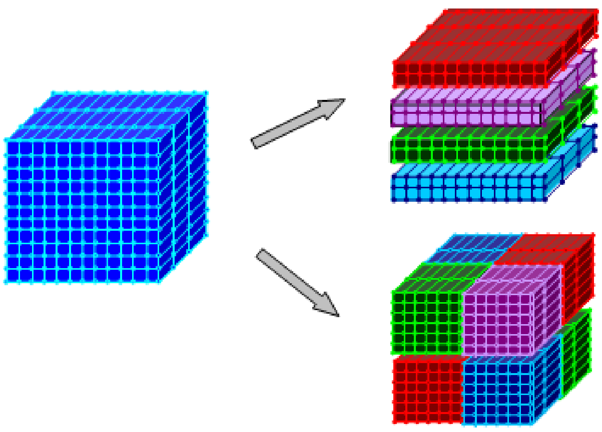
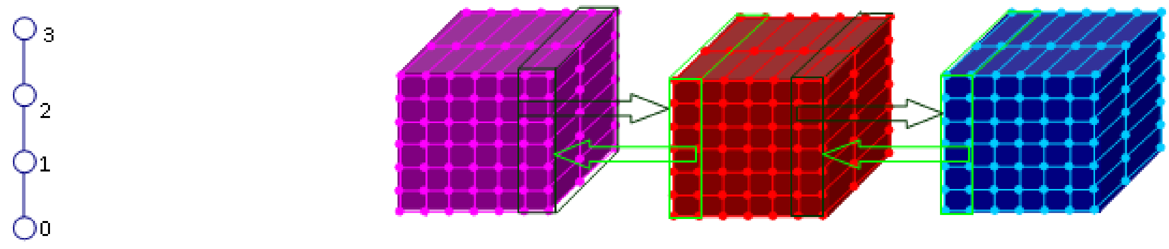


Рис. 2: Есептеу облысын бір өлшемді және үш өлшемді декомпозициялау

Бір өлшемді декомпозициялауда облысты x, y немесе z өсінің бойымен n процессорлар санына тең бөлеміз(Сурет 3-a), мысалға 1-суретте облыс z өсінің бойымен 4 бөлікке бөлінген, сонда әр процессор облыстың 1/4 бөлігін жеке дара есептеп әр итарация сайын асты және үстінде орналасқан көрші процессорлармен шекаралық нүктедегі мәндерін өз ара алмастырып жаңартып келесі итерацияға көшіп отырады(Сурет 3-b).

Ал үш өлшемді декомпозициялауда облысты x, y және z өстерінің бойымен n процессорлар санына тең бөлеміз(Сурет 4-a), мысалға 2-суретте облыс z өсінің бойымен 8 бөлікке бөлінген, сонда әр процессор облыстың 1/8 бөлігін жеке дара есептеп әр итерация сайын астында және үстінде, алдында және артында, оңында және солында орналасқан



a) b)

Рис. 3: a. Есептеу облысының процессорларға бөлінуі. b. Процессорлар ара итарацияның соңында шекаралық нүктедегі мәндерінің өзара алмасып жаңаруы

көршілес процессорлармен шекаралық нүктедегі мәндерін өз ара алмастырып жаңартып келесі итерацияға көшіп отырады(Сурет 4-b).

a) b)

Рис. 4: a. Есептеу облысының үш процессорларға бөлінуі. b. Процессорлар ара итерацияның соңында шекаралық нүктедегі мәндерінің өзара алмасып жаңаруы

Үш өлшемді жылу өткізгіштік теңдеуге параллелдеу әдісін пайдалану келесі қадамдардан тұрады:

- Декомпозициялау. Есептеу облысына байланысты деректерді декомпозициялау, бір өлшемді, екі өлшемді немесе үш өлшемді, сол арқылы берілген есепті кішкене блоктарға бөліп жеке дара процессорларда есептеліну үшін жағдай жасау. Есептеудің жалпы саны n3.

- Байланыстарды ұйымдастыру. Шешімді анықтау теңдеуінен u(i,j,k) мәнін анықтау үшін алдынғы уақыт бірлігіндегі x,y және z өсідері бойынша көршілес үш нүктенің мәндері пайдаланылады, сондықтан шекарадағы нүктелерді есептеу үшін көрші блоктағы шекаралас нүктедегі мәндер жоғарыдағы сүретте көрсеткендей өзар ауысып жаңарып отырады.

- Біріктіру әр процессорда есептелген есептеу блоктың нәтижесін түбір процессорға яғни 0- ші процессорға жинақтау, блоктардан бастапқы декомпозициялан есептеу облысын қайта қалпына келтіру, есептеу нәтижесін файлға жазу.

Таблица 1: Қателікті анықтау

Есептеу түрі	$\Delta u1_{i,j,k}$	$\Delta u2_{i,j,k}$
Тізбектелген бағдарламада	2.2422862330381983E-5	1.3837427552296915E-13
1D параллельді бағдарламада	8.413022901795377E-5	1.3832589343445153E-13
3D параллель бағдарламада	2.8897862813792585E-7	5.055380255129835E-14

4. Алынған нәтиже

Бұл есеп сегіз ядролық процессоры бар компьютерде тізбектелген, 1-D және 3-D декомпозиция бойынша параллельді бағдарламасы құрылды және тәтижелірei алынды.

Есептеудің қателігі төмендегідей:

$$\Delta u1_{i,j,k} = MAX(\left|u_{i,j,k}^{n+1} - u_{i,j,k}^{n}\right|); \Delta u2_{i,j,k} = MAX(\left|Eu_{i,j,k}^{n+1} - u_{i,j,k}^{n+1}\right|).$$

бұнда $\Delta u1_{i,j,k}$ салыстырмалы қателік болып ол (i,j,k) нүктесінің алдынғы уақыт бірлігімен осы уақыт бірлігіндегі есептелген u(i,j,k) жуық мәндерінің айырымының максимумы, ал $\Delta u2_{i,j,k}$ абсолютты қателік болып ол (i,j,k) нүктесінің t уақыт бірлігіндегі дәл шешімі мен осы уақыт бірлігінде есептелген u(i,j,k) жуық шешімінің айырымының максимумы. Төмендегі кестеде N=168*168*168 дағы есептеу қателіктерінің кестесі.

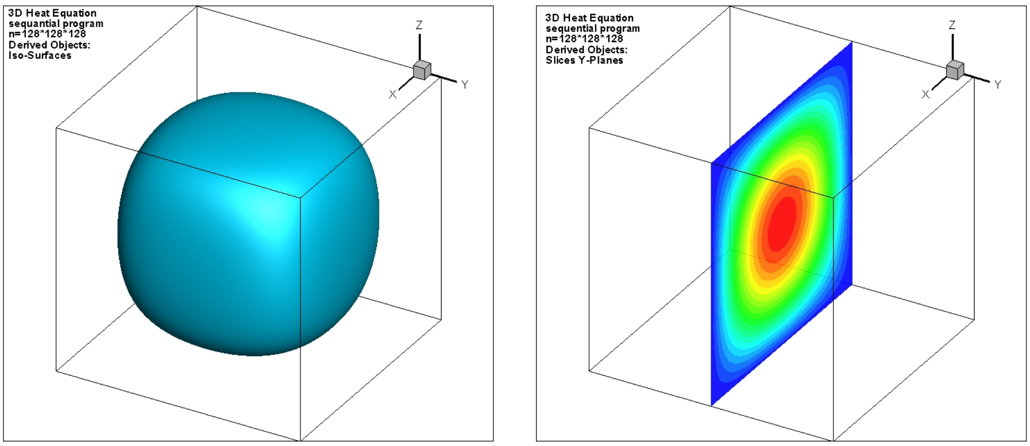


Рис. 5: 3D жылу өткізгіштік теңдеуінің тізбектелген бағдарламасының нәтижесі

Жоғары суреттерден байқағанымыз есептеуден алынған Iso беттері бір біріне ұқсас шар, ал центріндегі x,y және z бойынша алынған қимадан температураның әр бағытқа бірдей таралғанын көреміз, бұдан үш түрлі есептеулеріміздің нәтижелері деңгейлес келетінін білеміз және параллелдеу алгориимның дұрыстығы дәлелденеді.

Параллельді есептеудің өнімділігін бағау мақсатында біз параллельді есептеудің үдеуін және тиімділігін анықтаймыз.

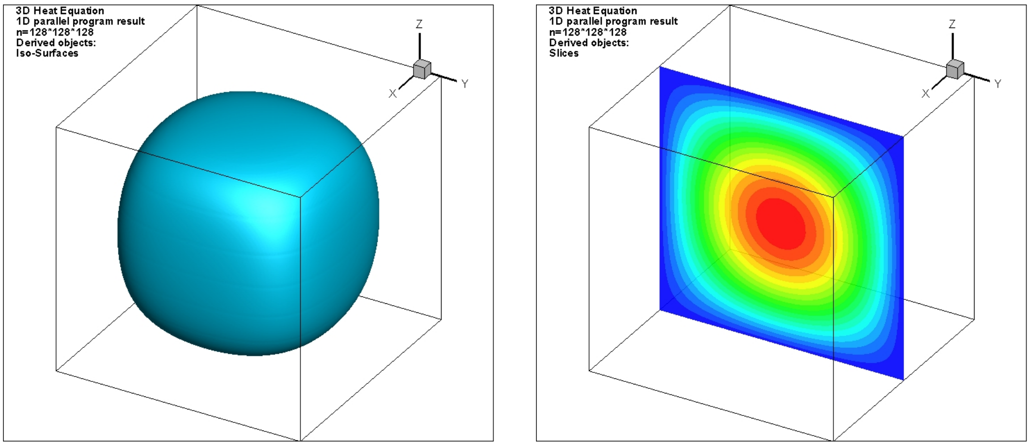


Рис. 6: 3D жылу өткізгіштік теңдеуінің 1D декомпозициялық параллельді бағдарламасының нәтижесі

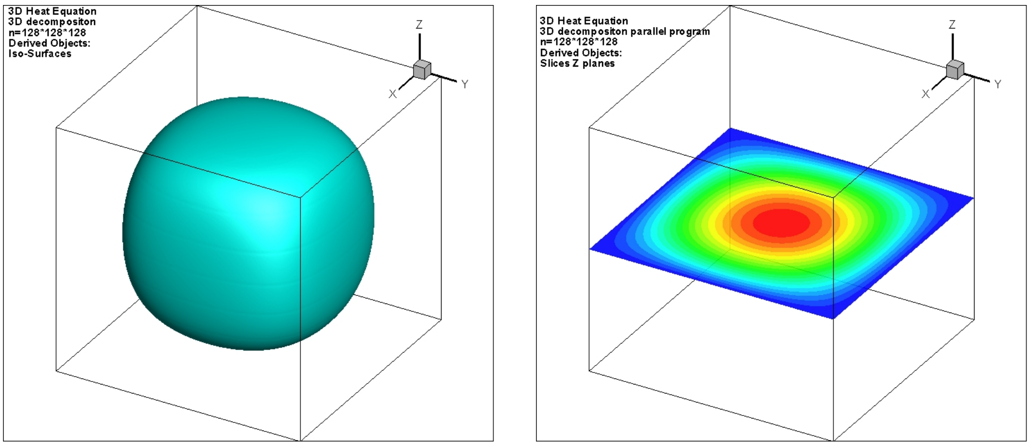


Рис. 7: 3D жылу өткізгіштік теңдеуінің 3D декомпозициялық параллельді бағдарламасының нәтижесі

- Параллельді есептеудің үдеуі

  Бағдарлама кодының *үдеуі* сіздің бағдарламаңыздың мултипроцессорда параллельді орындалғанда қаншалықты өнімділікке жеткендігін айқындайды.

  – Қарапайым анықтамасы ол бір бағдарламаның бір процессорда орындалған уақытының ұзындығын көп процессорда орындалған уақыттың ұзындығына бөлінуі яғни

    $S_p = \frac{T_1}{T_p}$

  – Үдеудің мәні әдетте 0 және p аралығында болады, p процессордың санына тең.

    Сызықты үдеу

  – Егер бір процессорда t уақыт көлемінде бір есепті шығарса және p процессорда бұл есепті t/p уақытта есептесе онда біз кемелді немесе сызықты үдеуді аламыз (Sp=p). Егер бағдарлама 4 процессорда орындалса онда уақыты 4 есе тездетіледі, ал 8 процессорда орындалса онда уақыты 8 есе тездейді. Сызықты үдеуді төмендегі суреттен көре аламыз.

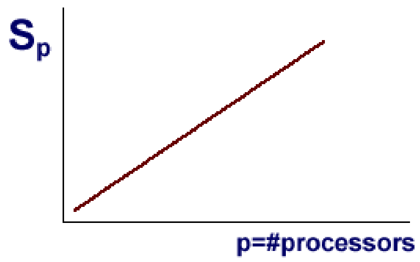


Рис. 8: Сызықты үдеу.

- **Параллельді есептеудің тиімділігі**

Тиімділік параллелдеудің өнімділігін анықтайтын тағы бір параметр болып ол үдеумен өте тығыз байланыста болады. p процессордың тиімділігі p процессордың үдеуімен p – ның қатынасына тең болады.

$$E_p = \frac{S_p}{p}$$

Тиімділік 0 мен 1 аралығындағы кішкентай үзінді. Ep=1 болса Sp= p кемел үдеу алынғандығын білдіреді. Тімділікті әр процессордың ортақ үдеуін өрнектейді деп қарауға болады. Берілген үш өлшемді есеп нүкелер саны N=64x64x64, N=128x128x128, N=168x168x168, N=184x184x184 болып әр алгоритмға жеке-жеке төрт реттен есептелді. Біздің есебіміздің нәтижесінен алынғанесептеудің уақыты, үдеуі және тиімділіктері төмендегі кестелерде көрсетілген.

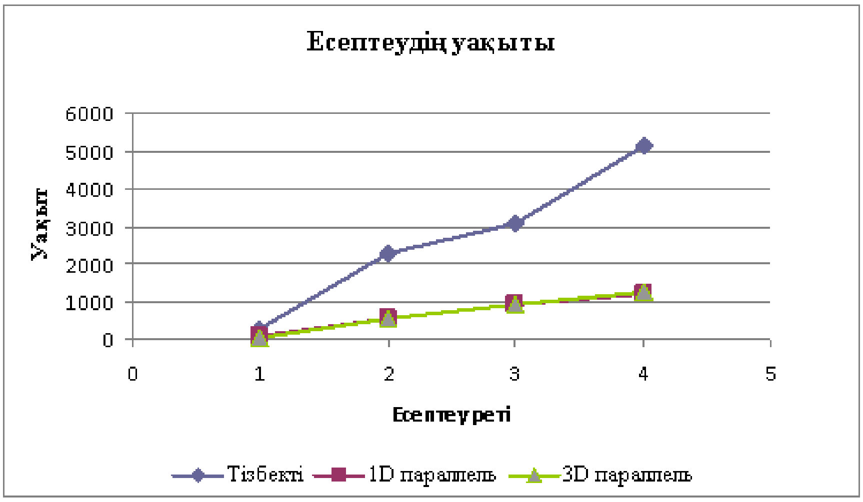


Рис. 9: Тізбекті және параллельді алгоритм бойынша есептелу уақыты кестесі

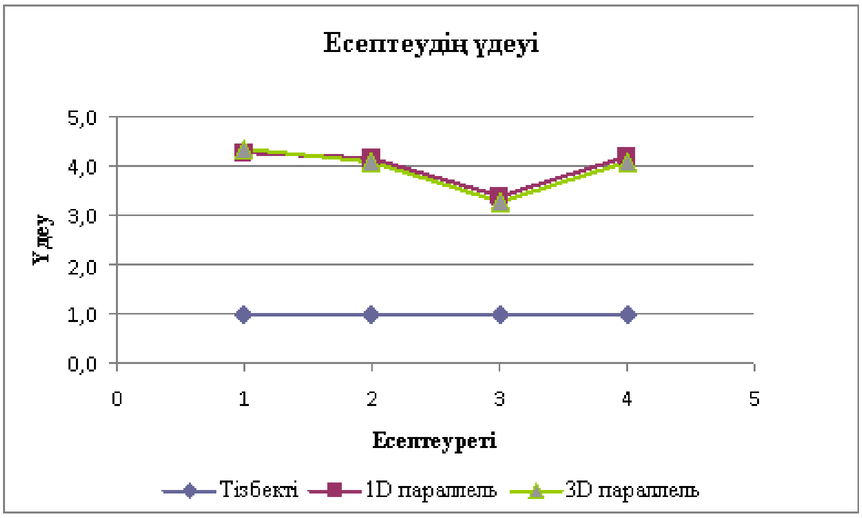


Рис. 10: Параллельді есептеулердің үдеу кестесі

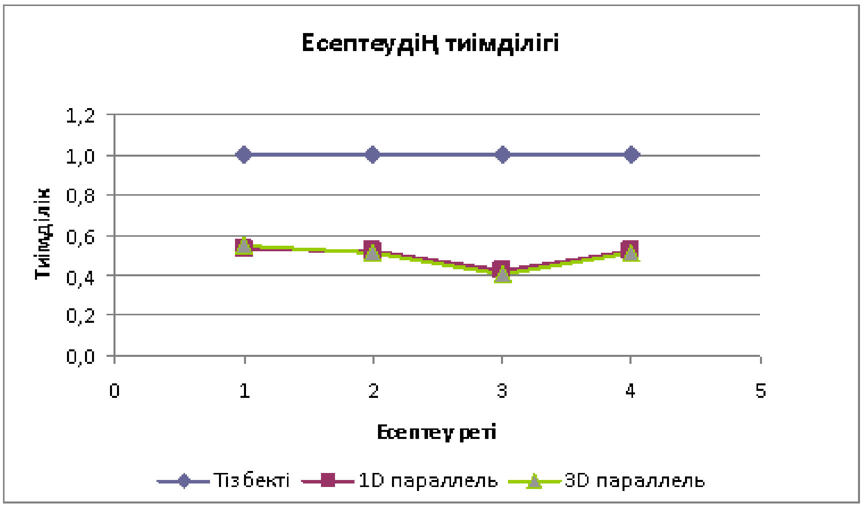


Рис. 11: Тиімділікті салыстыру кестесі

Кестелерден көргеніміздей параллельді алгоритмі бойынша есептеу нәтижесі тізбектелген алгоритммен салыстырғанда тез екендігін бірден байқай аламыз, ал параллельді есептеудің үдеуі және тиімдігі кестесінен параллельді есептеудің тізбектелгенге қарағанда өнімді екені тағы бір рет дәлелденеді.

5. Қорытынды

Үш өлшемді жылу өткізгіштік теңдеуді mpiJava технологиясы бойынша деректердің 1D және 3D декомпозициялану тәсілдерімен параллельдеп есептеу алгоритмі құрылды және бағдарламасы арқылы нақты нәтижелер алынды. Нәтижелерді салыстыру арқылы параллельді есептеудің тиімділігі айқындалды. Пайдаланылған алгоритм бойынша жылу өткізгіштік теңдеудің классына кіретін басқада осы тектес есептерді шығаруға болады, алдағы уақытта осы есептеуден жинақтаған тәжірибе кеуек ортадағы мұнайдың тұтқырлығын, қысымын, температурасы, қанықтылығын анықтайтын үш өлшемді есептерде пайдаланылады, ал ол жоғарғы өнімді жүйеде мұнайкен орындарын өңдеу технологиясын модельдеуге қолданылады.

Список литературы

[1] *Самарский А.А.* Введение в численые методы Москва: Наука P. 230.

[2] *Атанбаев С.* Сандық әдістердің алгоритмдері «Білім» баспасы. 87 б.

[3] *Mark Baker* mpiJava: A Java interface to MPI University of Portsmouth Southsea

[4] *Томас Б.* VI Российско-Германская школа по параллельным вычислениям на высокопроизводительных вычислительных системах Новосибирск 2011 P.81.

[5] *Букатов А. А., Дацюк В. Н., Жегуло А. И.* Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Издательство ООО «ЦВВР», 2003, 208 с.

[6] http://www.hpjava.org/courses/arl/lectures/mpi.ppt

[7] http://www.hpjava.org/reports/mpiJava-spec/mpiJava-spec.pdf

B. Matkerim, D.Zh. Akhmed-Zaki, *Organization of parallel computations to calculate the 3D heat equation using Java on High Performance Computing Systems*, The Bulletin of KazNU, ser. math., mech., inf. 2012, №1(72), 54 – 64

We carried out a study of technology development of the parallel algorithm for computing the 3-D heat equation using mpiJava technologies, as well as the organization of parallel computations with different levels of decomposition. The obtained approximate solution is compared with the analytical solution, while achieving sufficient accuracy of solutions.

Б. Маткерим, Д.Ж. Ахмед-Заки, *Организация параллельных вычислений для расчета 3D уравнения теплопроводности с использованием Java на высокопроизводительных вычислительных системах*, Вестник КазНУ, сер. мат., мех., инф. 2012, №1(72), 54 – 64

В работе осуществляется исследование технологий разработки параллельного алгоритма для вычисления решения 3D уравнения теплопроводности с использованием mpiJava технологий, а также организация параллельных вычислений с разными уровнями декомпозиций. Найденное приближенное решение сравнивается с аналитическим решением, при этом достигается достаточная точность расчетов.