ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ КАЗАХСТАН

ҚАЗ НТУ

ҚазҰТУ ХАБАРШЫСЫ

ВЕСТНИК КазНТУ

VESTNIK KazNTU

№3 (103)

АЛМАТЫ 2014 МАЙ

Учредитель:

Казахский национальный технический университет
имени К.И. Сатпаева

Регистрация:

Министерство культуры, информации и общественного согласия
Республики Казахстан № 951 – Ж "25" 11. 1999 г.

Основан в августе 1994 г. Выходит 6 раз в год

Адрес редакции:

г. Алматы, ул. Сатпаева, 22,
каб. 904, тел. 292-63-46
n. fedossenko @ ntu. kz

шешуге мүмкіндік беретін өрнектер нұсқауышын қолданған жөн. Нәтижесінде математикалық амалдардың орындау барысындағы көптеген қателіктерге жол берілмейді.

Өзге де .NET объектіге бағытталған тілдер секілді C# тілі жалпы программалау тілдерінің орындалу ортасында (CLR) жүзеге асырылатын MSIL (Microsoft intermediate language)–де жүктеледі. CLR қысқаша тиімділеуші JIT жүктеуішінің және қалдықтар жиынтығының кешені түрінде көрсетуге болады. C# тілі CLR функционалдылығының басым бөлігін қолданады және көрсетеді, сондықтан да орындалушы ортада қандай амал орындалып жатқанын бөліктеп қарастырған маңызды.

Ғылыми программалау жұмыстарында бұл ыңғайлы болуы мүмкін. Ғылыми есептеулер коды негізінде сандық амалдарды орындайды. Мұндай сандық амалдарды аз ғана уақытта орындау үшін біз аппаратты жабдықтарды тиімді пайдалана білуіміз керек.

SciMark эталонды тесттері ғылыми қосымшаларда кең таралған есептеу амалдарын пішіндейтін бірнеше өзектерден тұрады. Олардың әрқайсысының өз ерекшеліктері болады. Атап кеткеніміздей есептеуіш кодтар қалдықтар жиынтығын оқшауламайды. Бұл қағидалар жадыда көп орынды алматын өте қарапайым есептеу жұмыстарында жеңіл жүзеге асырылады. Барлығы орындалатын аймаққа және амалдардың өңделу үдерісіне байланысты орындалады.

Келесі мысалда матрицаларды бір-біріне көбейту және қалдықтар жиынтығын оқшаулау программасы көрсетілген. Біз мысал ретінде матрицаларды таңдадық, себебі олар көптеген ғылыми жобаларда жиі қолданылады. Матрицалар компьютерлік графика, томография генетика, криптограмма электр желілері және экономика салаларында көптеген құбылыстардың тәжірибелік шешімдерін алу үшін қолданылады.

Мысал:

```
using System;
class Matrix
{
    double[,] matrix;
    int rows, columns;
                {
                Console.WriteLine("Finalize");
    }
    public Matrix(int sizeA, int sizeB)
    {
                rows = sizeA;
                columns = sizeB;
                matrix = new double[sizeA, sizeB];
    }
                public double this[int i, int j]
    {
                set { matrix[i,j] = value; }
                get { return matrix[i,j]; }
    }
                public int Rows
    {
                get { return rows; }
    }
                public int Columns
    {
                get { return rows; }
    }

}
class MatMulTest
{
    static void Main(string[] args)
    {
```

```
int i, size, loopCounter;
Matrix MatrixA, MatrixB, MatrixC;
size = 200;
MatrixA = new Matrix(size,size);
MatrixB = new Matrix(size,size);
MatrixC = new Matrix(size,size);
for (i=0; i<size; i++)
{
        for (int j=0; j<size; j++)
        {
                MatrixA [i,j]= (i + j) * 10;
                MatrixB [i,j]= (i + j) * 20;
        }
    }
    loopCounter = 1000;
    for (i=0; i < loopCounter; i++) Matmul(MatrixA,
            MatrixB, MatrixC);
    Console.WriteLine("Done.");
    Console.ReadLine();
}
public static void Matmul(Matrix A, Matrix B, Matrix C)
{
    int i, j, k, size;
    double tmp;

    size = A.Rows;
    for (i=0; i<size; i++)
    {
        for (j=0; j<size; j++)
        {
            tmp = C[i,j];
            for (k=0; k<size; k++)
            {
                tmp += A[i,k] * B[k,j];
            }
            C[i,j] = tmp;        }
    }
} }
```

Енді CLR-ден C# тіліне көшсек. Атап кеткеніміздей C# программалау тілі бірнеше программалық тілдердің жиынтығы. C# - объектіге бағытталған тіл. Біздің заманымыз өзара тығыз байланысқан объектілерден тұратын болғандықтан, объектіге бағытталған программалау тілдері ғылыми есептеу жұмыстары үшін өте ыңғайлы прогаммалау тілі болып табылады. Сонымен қатар, құрылымдық объетіге бағытталған программалық код ғылыми модельдердің өзгерістеріне байланысты ішкі әдістер жеңіл және жылдам өзгертіле береді. Бірақ барлық ғылыми құбылыстарды объектілер және олардың өзара байланысы арқылы көрсетуге келмес – мұндай жағдайда объектілерге нұсқау күрделі қиындықтар туындатуы мүмкін. Мысал ретінде молекулярлық динамиканы алуға болады. Молекулярлық динамика - есептеуіш химия, физика, биология және материалтану ғылымдарында кең қолданылады. Ол компьютерлерді ғылыми қолданудың алғашқы аймағы болды: 1957 жылы Элдер (Alder) және Уэйнрайт (Wainwright) аргондардың шамамен 150 атомының қозғалысын пішіндеген. Молекулярлық динамикада ғалымдарды атомдардың өзара қатынасын жұптық потенциалдар (pairwise potentials) арқылы модельдеу қызықтырады, бұған мысал ретінде күннің, планетаның жұлдыздардың өзара қатынасына гравитацияның әсерін алуға болады. Нәтижесінде жұптық күш пен энергияның есептеуге арналған теңдеуі күрделі болады, сондықтан оны

есептеу үшін объектіге бағытталған жүйелерді қолданудан бас тартып дәстүрлі есептеу үрдісін пайдалануға тура келеді. Ал дәстүрлі үрдістер коды да өндірісте қолайсыз болуы мүмкін. Мұның барлығы мәліметтерді сақтау және қолдану алгоритмдерінен тәуелді болады. Осы есептеу жұмыстарын C# тілінде шығаруға болады. Әрине объектіге бағытталған программалау тілін пайдалану тиімді шешім болмауы мүмкін, бірақ ғылыми программалау бастамалары үшін қолайлы болып келеді. Нәтижелерді алу және есептеу жұмыстарын жүргізуге арналған барлық айнымалылар мен әдістерге жалғыз класс қолданылады.

C# тілінде сілтемелік (reference types) («тяжеловесные» объекты) және шамалар типі немесе мәндік типтер (value types) («облегченные» объекты) болып бөлінетін екі негізгі категория бар. Сілтемелік типтерге жады үйінділерден (жалғыз ерекшелік(қателік) — stackalloc кілттік сөзін қолдану) бөлінеді; олар абстракцияның қосымша деңгейлерін құрады, яғни олардың сақталған орнына сілтеме арқылы енуді талап етеді. Бұл типтерге тікелей қатынас жасауға болмайтындықтан сілтемелік типтің айнымалылары дерлік барлық кезде сілтемені нақты объектте(немесе null) сақтайды. Жады үйіндіден бөлінетін болғандықтан есептеу жұмыстары жүргізілетін орта әрбір ерекшелінген сұраныстың дұрыстығына сенімді болуы керек. Жадыны сәтті ерекшелейтін келесі мысалды қарастырайық:

Matrix m = new Matrix(100, 100);

CLR жадысының байланыс құралын басқарушы ерекшелеуге сұраныс алады, объетілерді, класс айнымалыларын сақтауға арналған жады көлемін есептейді. Содан соң жады байланыс құралын басқарушы үйіндіде есептеу жұмыстарын сақтауға арналған орын бар жоғын тексереді. Егер нысандарды сақтауға арналған орын жеткіліксіз болса, онда орынды босату және үйінділерді сығу үшін қалдықтар жиынтығы іске қосылады.

Егер есептеу үрдісі сәтті өтсе, онда жады байланыс құралын басқарушы объект жадыға сақталмас бұрын өте маңызды амалды қолдануы қажет. Ол қалдықтар жиынтығын ұрпақтан ұрпаққа (generational garbage collection) қолдану үшін қажетті және жазбалар тоскауылында (write barrier) блок коды туындалады. Өз кезегінде амалдарды орындаушы орта жазбалар тоскауылын объект жадыдағы нақты адреске жазылған сайын немесе объект жадыдағы басқа объектке сілтеме жасалған кезде генерациялайды. Ұрпақтар бойынша қалдықтар жиынының бұзылмауын есте сақтау қажетті бөліктердің бірі – өзге объектілерден осы объектке сілтеме жасалған уақытта объектілерде осы жазбаның қателіксіз жинақталған болуы өте маңызды. Жазбалар тоскауылы амалдардың орындалу кезеңінде азғана кідірісте болады, сондықтан бірнеше нысандарды құру ғылыми қосымшалар үшін тиімсіз болып табылады.

Мәндік типтер стекте сақталады. Мұндай типтерге абстракцияның қосымша деңгейлері қажет емес, сондықтан мәндік типтердің айнымалылары барлық тек сан мәнін сақтайды (сондықтан null мәніне ие болмайды). Мәндік типтердің сілтемелік типтермен салыстырғандағы басты ерекшелігі онда құрылған қосымшалар кідірісте болмайды. Оларға жады стектің қарапайым көрсеткіші арқылы стектен бөлінеді. Бұл объектілер ешқашан қалдықтар жиынтығына бастамашылық етпейді. Сонымен қатар мәндік типтер үшін жазбалар тоскауылы туындалмайды.

C# тілінде мәндік типтердің мысалы болып – мәліметтердің қарапайым типтері(int, float, single, byte), санауыштар және құрылымдар болып табылады. Сілтемелік типтің ішінде орналасқан мәлімет типі стекте сақталады. Мысалы, келесі кодқа назар аударайық:

```
class Point
{
   private double x, y;

   public Point (double x, double y)
   {
      this.x = x; this.y = y;
   }
}
```

Бұл класс көшірмесі 24 байттан тұрады, оның ішінде 8 байт объекттің бас тақырыбына бөлінеді, ал қалған 16 байты - double: x және y типіндегі екі айнымалыға бөлінеді. Сонымен қатар сілтемелік типті мәндік тип объектісінің ішіне орнатқанымен барлық нысан үйіндіге орнатылмайды. Үйіндіде тек массив қана ерекшеленеді, сілтеме стекте орналасады.

мі
Сондықі
құрылыш
сонымен
типтерде
Ғы
тиімдірек
немесе те
жұмсады
(CTS) спе
Өнд
қолданадь
пайдалану]

1. П
М.: Издател
2. *Ъ*
СПб.:Питес
3.
I Электрона
ru/library/dc
4. II
2004. - 752 I

1 . Petj
- M : . Izdateli
2 . Tro
R.Mikheyev. I
3 . Fakh
I Elcktronnyy
in_library/dd353
4 . Shild
752 s . . I

Ғылым
Түйіндеі
кодтарды жеңіл
ғылыми жобала
Сонымсн катар,
жылдам орындал
Туйін сөз
(CLS) жэнс Comr

Применен)
Резюме,
генерации кода,
научном сообщесі
которые в C # *явз*
тестов и сравнить
C # с точки произв
Ключевые
Specification (CLS

Мәндік типтер өз кезегінде System.Object мұраланатын System.ValueType мұраланады. Сондықтан мәндік типтер кластарда бар функционалдылықты қамтамасыз етеді. Онда құрылымдауыштар, әдістер, индексаторлар және артық салмақтанған операторлар болуы мүмкін, они сонымен қатар интерфейстер жүзеге асырылуы мүмкін. Бірақ олардан мұраланбайды, және олар өзге типтерден мұраланбайды.

Ғылыми есептеу жұмыстары үшін мәндік типтер сілтемелік типтерге қарағанда жылдамырақ, тиімдірек орындалады. Көптеген өндірісшілер .NET Framework пайдаланып кітапханаларды құруға немесе технологиялар және программалық тілдер негізінде шешуге уақыттарын және еңбек қорларын жұмсады. .NET қосымшасындағы Common Language Specification (CLS) және Common Type System (CTS) спецификасының арқасында программалық тілдердің арақатынасын біріктіреді.

Өндірісшілер интернет компоненттерін және кең таралған қосымшаларды құру үшін C# қолданады, ал бұл программалау тілінің мүмкіндіктері өте көп, сондықтан оны ғылыми тіл ретінде пайдалануға болады.

ӘДЕБИЕТТЕР

1. Петгольц Ч. Программирование в тональности C#/ Ч.Петгольц; пер. с англ. под ред. Ю.П. Леоновой. - М.: Издательско-торговый дом «Русская редакция», 2004. – 512 с.: ил.

2. Троелсен Э. C# и платформа .NET. Библиотека программиста/ Э.Троелсен; пер. а англ. Р.Михеев. – СПб.:Питер, 2004. – 796 с: ил.

3. Фахад Г. C# и наука: применение языковых средств C# в проектах для научных вычислений [Электронный ресурс] // MSDN Magazine. Электрон. дан. URL: http://msdn.microsoft.com/ru-ru/library/dd353137.aspx, свободный. Яз. рус. (дата обращения 12.12.2011).

4. Шилдт Г. Полный справочник по C#/ Г. Шилдт : пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 752 с.:ил.

REFERENCES

1 . Petgol'ts CH . Programmirovaniye v tonal'nosti 100 # / CH.Petgol'ts ; per . s angl . pod red . YU.P. Leonovoy. - M : . Izdatel'sko - torgovyy dom " Russkaya redaktsiya " 2004 - 512 s . Il . .

2 . Troyelsen E . 100 # i platforma . Chistaya . Biblioteka programmista / E.Troyelsen ; per . a angl . R.Mikheyev . - SPb : . Piter , 2004 - 796 s : . Il .

3 . Fakhad G . 100 # i nauka : primeneniye yazykovykh sredstv 100 # v proyektakh dlya nauchnykh vychisleniy [Elektronnyy resurs] / / MSDN Magazine . Elektron . dan . URL: http://msdn.microsoft.com/ru-ru/library/dd353137.aspx, svobodnyy . YAz . rus . (Data obrashcheniya 12.12.2011) .

4 . Shildt G . Polnyy spravochnik po C # / G . Shildt , per . s angl . - M : Izdatel'skiy dom " Vil'yams " 2004 - 752 s . . Il . .

Мирзахмедова Г.А.

Ғылыми есептеу жұмыстарына C# программалау тілін қолдану

Түйіндеме. Мақалада C# программалау тілінің өңдеу жылдамдығы қиынға соғатын программалық кодтарды жеңіл құруға мүмкіндік беретін кейбір ішкі ерекшеліктері қарастырылған. C# программалау тілінің ғылыми жобаларды құру, сандық есептеулер жүргізу барысында маңызы зор екеніне көз жеткізуге болады. Сонымен қатар, жадыны басқару кезінде жүктелу жылдамдығы баяу орындалатын жобалардың жеңіл және жылдам орындалатынына көз жеткізуге болады.

Түйін сөздер: C# программалау тілі, JIT жүктемесі, NET қосымшасы, Common Language Specification (CLS) және Common Type System (CTS).

Мирзахмедова Г.А.

Применение языка программирования C# для научных вычислений

Резюме. В статье рассмотрены некоторые внутренние C #, что позволяет легко и практично для генерации кода, чувствительного к скорости исполнения. Вы можете увидеть, что большую роль играет C # в научном сообществе, открывая дверь в численных расчетах следующего поколения. Я исследую те качества, которые в C # являются хорошей альтернативой в мире численных расчетов, а также результаты нескольких тестов и сравнить их с результатами неуправляемого C + +, для того, чтобы понять, на каком уровне находится C # с точки производительности и эффективности.

Ключевые слова: Язык программирования C#, JIT компилятор, .NET Framework, Common Language Specification (CLS) и Common Type System (CTS).

Mirzahmedova G.A.

Application C # programming language for scientific computing

Summary. Here we consider some internal C #, that makes it easy and practical to generate code that is sensitive to the speed of execution. You can see what a big role to play C # in the scientific community, opening the door to the next generation of numerical calculations. I explore the qualities that in C # is a good alternative in the world of numerical calculations and the results of several tests and compare them with the results of an unmanaged C + +, in order to understand at what level is C # in terms of productivity and efficiency.

Key words: Programming language C#, JIT compiler, .NET Framework, Common Language Specification (CLS) and Common Type System (CTS).

ӘОЖ(378.016.02:004.032.6:574)

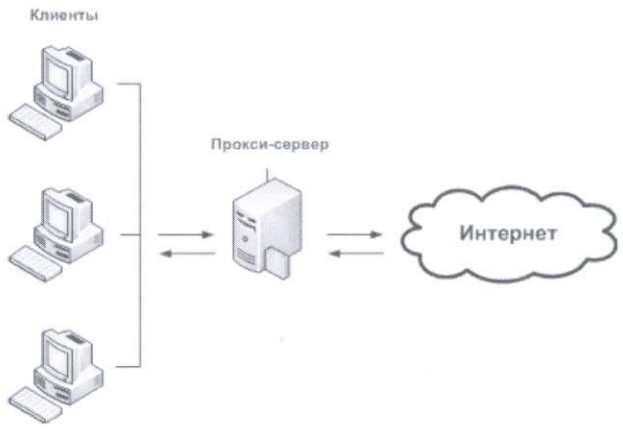
Г.А. Мирзахмедова, А.Қ. Самбетбаева
(Әл-Фараби атындағы Қазақ ұлттық университеті
Алматы, Қазақстан Республикасы, gulban84@mail.ru, erulan72@mail.ru)

C# ТІЛІНДЕ ПРОКСИ СЕРВЕРДІ ӨҢДЕУ

Аннотация. Прокси – сервер клиенттер мен өзге серверлер және желілік қызметтер арасында мәліметтерді тасымалдауды ұйымдастыруға мүмкіндік береді. Прокси – серверлерді Интернет желісіне қосылу мүмкіндігін алу, желілік қосымшаларды ретке келтіру және т.с.с. қызметтер үшін қолдануға болады. Бұл мақалада C# программалау тілінің көмегімен сокеттерді пайдалану арқылы жеке *HTTP* прокси серверін құруды қарастырамыз.

Түйін сөздер: C# программалау тілі, *HTTP* прокси сервері, System.Text.Encoding NET қосымшасы, ItemBase.

Прокси – серверлер тура және кері болып екіге бөлінеді. Тура прокси – сервер қолданушы серверін өзге серверлермен байланыстырушы құрылғы болып табылады, қолданушының сұранысын желілерге тікелей аударады және қайта қолданушыға хабарлама жібереді. Кері прокси серверлер желідегі жүктемелерді тарату, қолданушының сұранысын бір желінің ішінде бірнеше серверлерге қайта аудару үшін қолданылады.



1-сурет. Прокси сервердің қызметінің сұлбасы.

Интернет желісінде көпке танымал ол **HTTP** хаттамасы. Прокси – серверде бір мезгілде бір немесе бірнеше хаттамалармен жұмыс жасауға болады. Прокси-серверді құру үшін «Windows консольді қосымшасы» типіндегі жобаны қолданған жөн. Жобаны «**Прокси_Сервер**» деп атайық. Прокси сервер нақты портта тыңдалуы қажет. Порттарды тыңдау үшін стандартты кластарды қолданамыз. NET **Framework** –та портқа кластың атын **TcpListener** деп береміз. Инициализациялау барысында **TcpListener** класы тыңдалуға қажетті екі парметрді IP-адрес және портты қабылдайды. Жергілікті машинада IP-адрес **127.0.0.1** –ге тең, порт ретінде кез келген бос портты қолдануға болады.

TcpListenerinyTC

*Циклдщ деносія^* орналастырылуы қажет *функциясътың* көмегім

}
if (myTCP.Pendm

using (Socket myf
if (myClient.Conr

Жауапты баш түрінде берілуі мүмк қолдану қажет. Қось болады, сондықтан қабылдап, оларды ба

private *static Щ*
{
byte[] b = new
int len = 0;

using (Memoi

while (mySod
my S ocket. ReceiveB

m.Wnte(b, 0,

return ***m.Tol***

byte[] *htipl*

Содан ***coi*** кажеттігі турал) катарларды алй мазмұнынан ер рсгулярлы өрне сұранысы байт жасайтын бол функциясының

Regex ш
RcgcxOptionsJ
Match m
string he
int port
if (lint'

IP-адре

ҚазҰТУ