Проект

по Системи за Паралелна Обработка  
на тема  
  
  
 Паралелен Алгоритъм за Изчисляване на  
Неперово число

Изготвен от: Диана Иванова

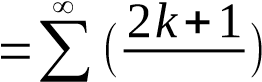
Ръководители: проф. Васил Георгиев, ас. Христо Христов

Проверил: ........................

ФН:81426, 2 група  
Компютърни науки

Условие на задачата

Едно важно за математиката число е Неперовото число (Ойлеровото число), тоест числото e. Използвайки сходящи редове, можем да сметнем стойността на e с произволно висока точност. Един от сравнително бързо сходящите към e редове е:

*е* *k*=0 (2*k*)!

Вашата задача е да напишете програма за изчисление на числото e използвайки цитирания ред, която използва паралелни процеси (нишки) и осигурява пресмятането на e със зададена от потребителя точност. Изискванията към програмата са следните:

(o) Команден параметър задава точността на пресмятанията. По Ваше желание, точността се изразява или в брой цифри след десетичната запетая или в брой членове на реда. Командният параметър задаващ точността има вида „**-p 10000**”;

(о) Друг команден параметър задава максималния брой нишки (задачи) на които разделяме работата по пресмятането на e – например “**–t 1**” или “**–tasks 3**”; (о) Програмата извежда подходящи съобщения на различните етапи от работата си, както и времето отделено за изчисление и резултата от изчислението

(стойността на ***e***);

(o) Записва резултата от работа си (стойността на ***e***) във изходен файл, зададен с подходящ параметър, например “**-o result.txt**”. Ако този параметър е изпуснат, се избира име по подразбиране;

(o) Да се осигури възможност за „**quiet**“ режим на работа на програмата, при който се извежда само времето отделено за изчисление на ***e***, отново чрез подходящо избран друг команден параметър – например “**-q**”;

# 2. Описание на реализацията на задачата

За паралелното решаване на задачата е избран подходът на многонишковото програмиране на eзика за програмиране Java.

Програмата се резалиризира чрез два класа: Main и еThread.

В Main класа се извършват подготвката за изчислението, разделянето на задачите за всяка нишка, обработката на аргументите, допълнителни функции.

Аргументите зададени при стартиране на програмата са обработени чрез CommandLineParser.   
 Измерването на времената започва преди началото на изчислителните процеси.

Всяка нишка прави изчисление на сумата на няколко определени последователни членове от сходящия ред.

Главната нишка създава останалите и се грижи за тяхното “събиране”.   
След като зададе работа на максималният брой нишки, тя ги изчаква да приключат, след което събира сумите, които всяка от тях е акумулирам.   
 Нишките се пазят в масив, чрез който лесно може да се достъпят техните даннил   
 След като мастър-нишката е събрала сумите, тя прави проверка. Новата получена сума сравнява с по-стара предварително запазена сума. Ако са равни, това значи че е достигната нужната прецизност и алгоритъмът завършва. \*\*Ако сумите не са равни, главната нишка отново задава работа на slave-нишките и стъпките се повтарят. Този цикъл продължава докато не е достигната нужната прецизност.  
  
\*\*Забележка: след като е достигната нужната прецизност, каквато и достатъчно малка сума да добавим(както е в случая), няма да има разлика между новата и старата сума, тъй като числото се закръгля след тази точка на прецизност, а малките суми които добавяме са всеки път по-малки.

Нишките са реализирани като наследници на класа Thread.   
В метода run() се реализира изчислението на членовете на сходящият ред.

Като се пресмята знаменател и числител и по този начин се оформя дроб, която се добавя към член-данната сума на съответната нишка.

В края на метода run(), ако не е вдигнат флагът q за тих режим   
Спомагателната функция factorial(BigInteger n) – имплементира пресмятането на факториел.  
 В края на метода run(), ако не е вдигнат флагът за тих режим q – се извежда съобщение с информация за нишката.

Резултати

Таблици и графики:

Sp = T1 / Tp = ускорение (speed-up) на програмата при p на брой нишки

Ep = Sp / p = ефективност на програмата при p на брой нишки

Tp = време за изпълнение на програмата с p на брой нишки

