
Введение в параллельные и распределенные вычисления

Судаков А.А.

ЮЦ КНУ
“Параллельные и распределенные
имені Тараса
вычисления” Лекція 1

Шевченка,

2005 р

Автор курса и преподаватель

Судаков Александр Александрович

кандидат физико-математических наук,
доцент радиофизического факультета Киевского национального
университета имени Тараса Шевченко,
руководитель лаборатории параллельных вычислений
информационно-вычислительного центра Киевского
национального университета имени Тараса Шевченко

ІОЦ КНУ

імені Тараса

Шевченка,

2005 р

Задачи курса

- Теоретические основы работы параллельных и распределенных систем
- Технологии построения параллельных и распределенных систем
- Практические навыки построения и работы с параллельными и распределенными системами
- Практические навыки разработки параллельных и распределенных программ

ІОЦ КНУ

імені Тараса

Шевченка,

2005 р

Для чего это нужно ?

- Все современные компьютерные системы используют элементы параллельной обработки информации
 - Многопроцессорность, конвейерная обработка ...
- Все современные компьютерные системы используют распределенные вычисления
 - Многозадачность, базы данных, файловые сервера...
- Пользователи привыкли к тому, что можно работать «сразу» с несколькими компьютерами и программами
 - *Интернет, локальные сети, связанные объекты...*
- Некоторые задачи можно сегодня решить только с помощью параллельных и распределенных вычислений
 - *получение «чрезвычайно» высокой производительности*
 - *получение высокой надежности и отказоустойчивости*
 - *Некоторые ресурсы распределены по определению*
- Специалисты по компьютерным системам должны в этом разбираться

Научные и промышленные задачи, требующие параллельных вычислений

- Квантовая физика, химия, молекулярная биология
- Микроэлектроника
- Статистическое моделирование (метод Монте-Карло)
- Ядерная физика

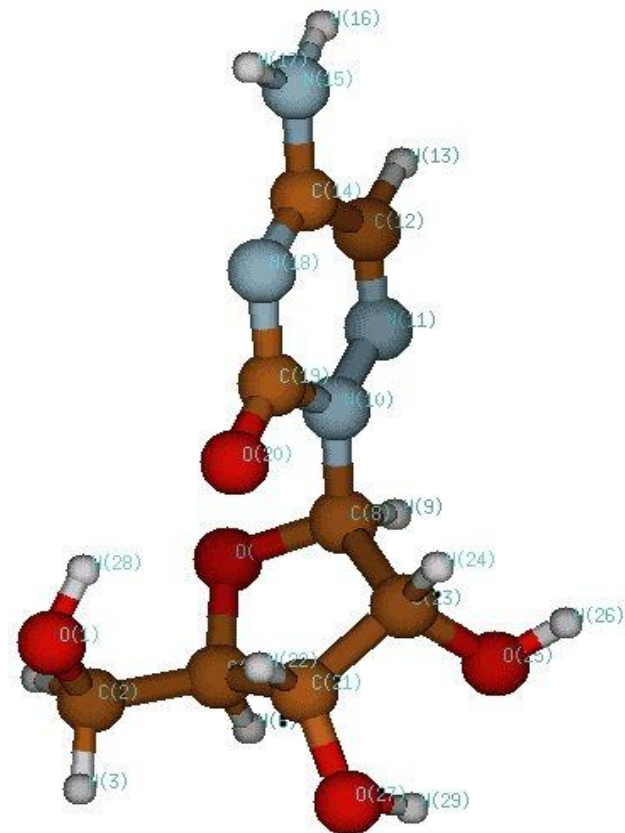
ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

ХИМИЯ

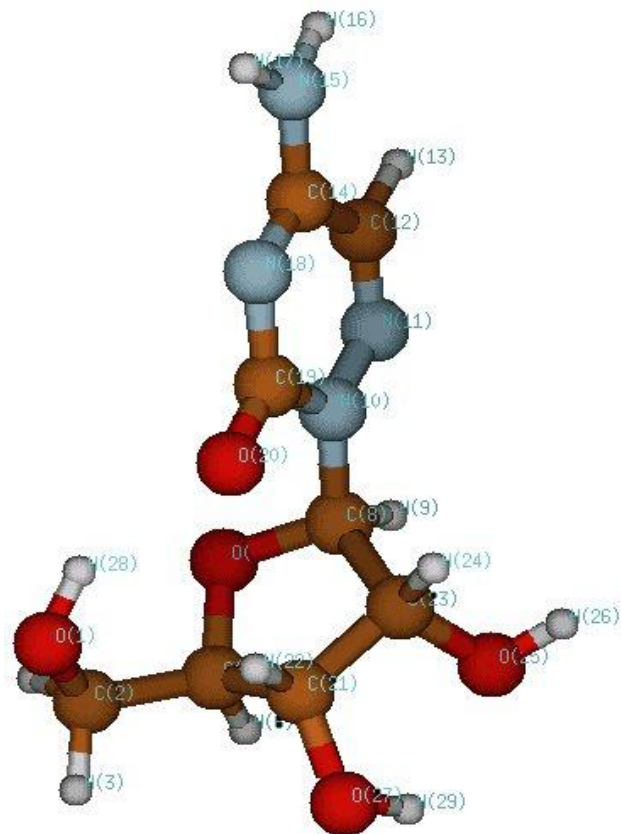
- Есть формула вещества (лекарственный препарат), найти, как это вещество вступает в реакцию, насколько оно устойчиво и как действует

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р



Как решается задача

- Свойства вещества определяются типом атомов, положением ядер и электронной конфигурацией
- Для нахождения электронной конфигурации необходимо решать уравнения квантовой физики
- Количество операций, и объем оперативной памяти, необходимые для решения определяются числом электронов молекулы N
- Количество операций пропорционально N^4-N^7
- Объем оперативной памяти пропорционален N^3-N^4

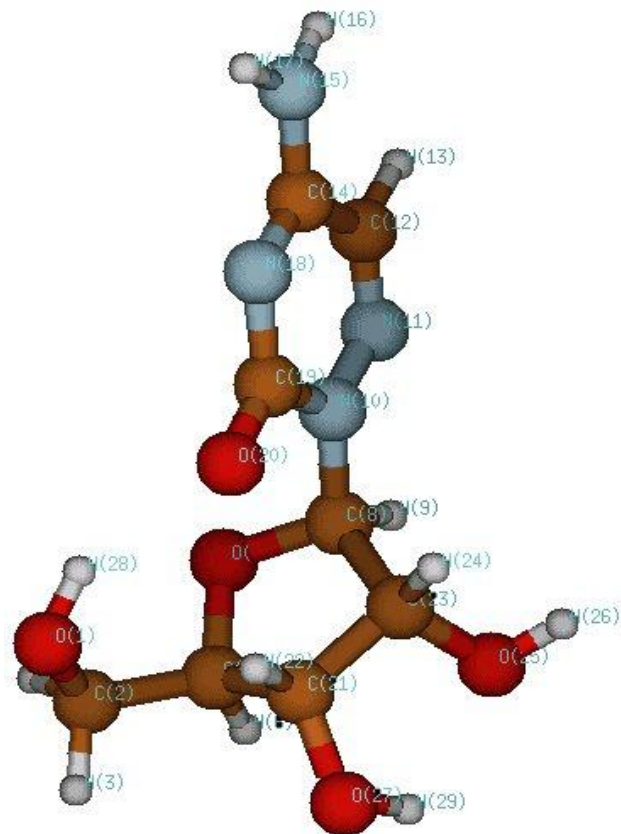


ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Оценка времени и ресурсов

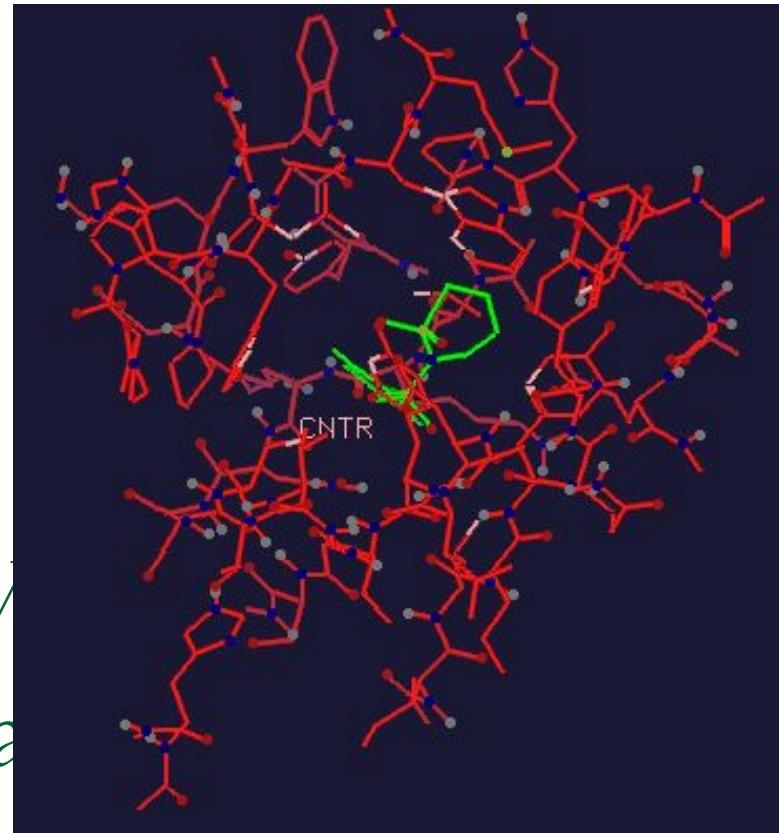
- Количество атомов 29
- Количество электронов $N=130$
- Количество базисных функций 280
- Количество операций $N^5 \sim 10^{13}$
- Задачу необходимо решать десятки/сотни раз $\sim 10^{15}$
- Время на процессоре производительностью 1 млрд. операций в секунду
- Памяти около 4 Гбайт
- Необходимо несколько процессоров



ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Молекулярная биохимия

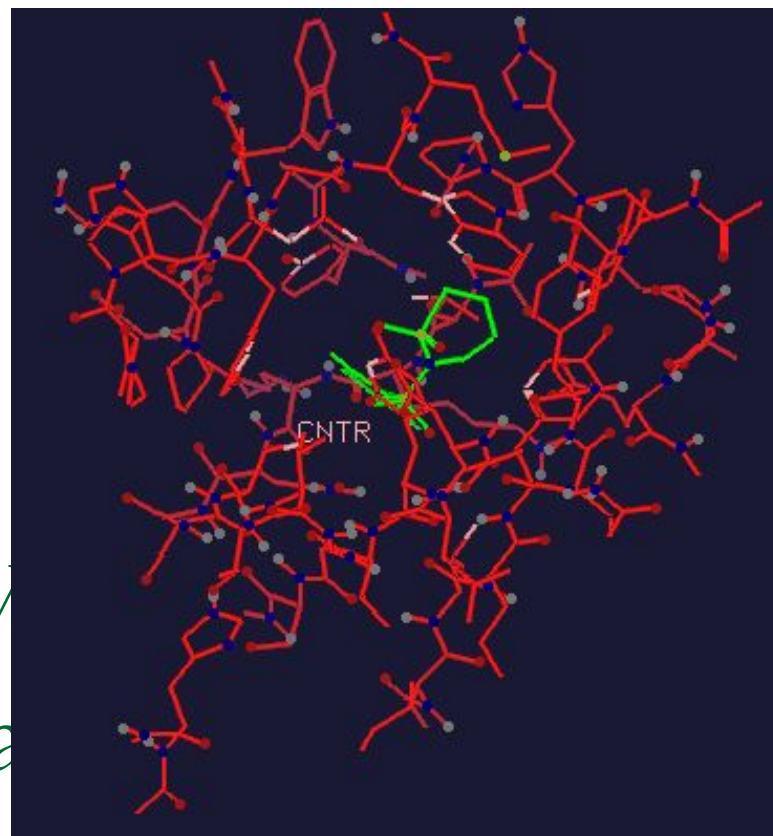
- Есть вирусный белок для которого нужно подобрать лекарственный препарат, который будет на него действовать
- Количество атомов несколько тысяч



ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Как решается задача

- Используются приближенные методы классической физики
- количество операций MN^2 , где M количество итераций, N количество атомов
- Требуется интенсивный обмен между процессорами
- Время расчета несколько недель



ІОЦ КНУ

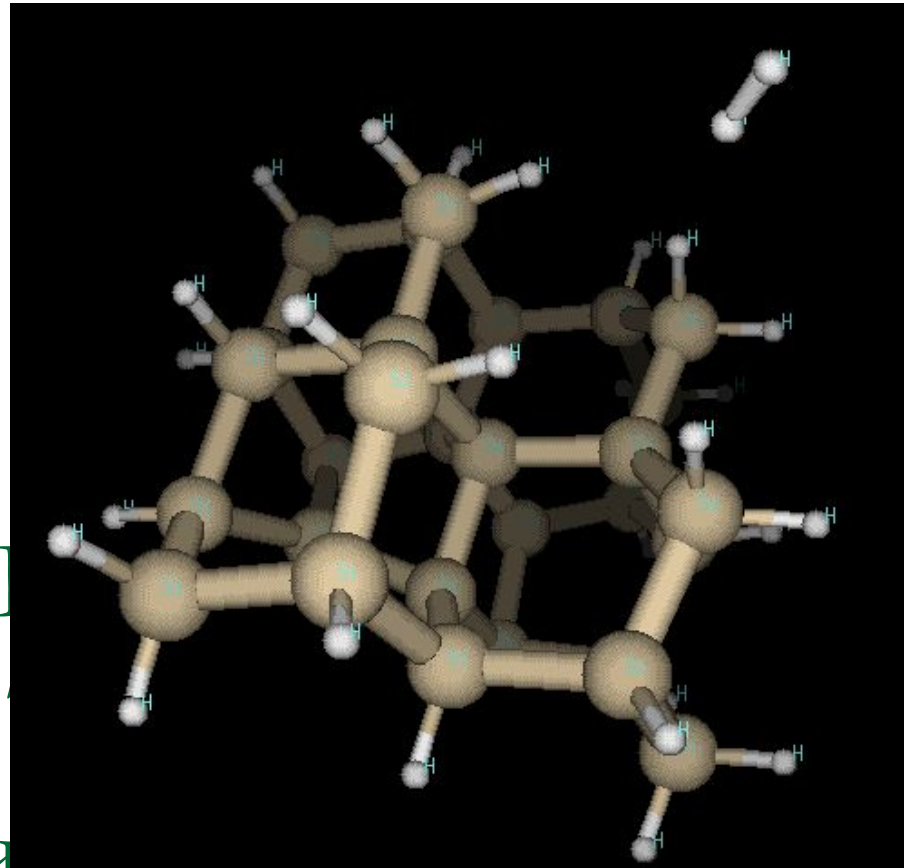
імені Тара

Шевченка,

2005 р

Микро (нано) электроника

- Исследование поведения атомов на поверхности кремния для создания новых технологий
- Требуется квантово-физических расчетов



ЮЦ
імені
Шевченка,
2005 р

Ядерная физика

- Взаимодействие ионизирующего излучения с веществом
- Моделируется поведение большого количества частиц

- Обработка данных с ускорителей

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р



Использование распределенных вычислений

- Интернет приложения
- Высоконадежные системы
- Параллельные вычисления

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Программа курса

- Лекции (40 часов)
- Семинарские занятия (30 часов)
- Практические занятия (30 часов)
- Лабораторные занятия (40 часов)

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Лекции

- Введение (1 лекция)
- Средства параллельных и распределенных вычислений (9 лекций)
- Теоретические основы параллельных вычислений (5 лекций)
- Разработка параллельных и распределенных систем (5 лекций)
- Контрольные работы (2 занятия)
- Зачет

ІОШ КНУ
Імени Тараса
Шевченка,

2005 р

Семинарские занятия (15 занятий)

- Распределенные операционные системы
- Задачи, требующие параллельных вычислений и соответствующее программное обеспечение: 3D анимация, математические пакеты, физические, химические, экономические и д.р. задачи
- WWW технология, Java и их применение
- MS Windows домен, Active directory, NetBios
- Средства коммуникации для параллельных кластеров: Myrinet, SCI
- Промышленные высоконадежные кластеры
- Промышленные высокопроизводительные системы (суперкомпьютеры, кластеры)
- Распределенные файловые системы (NFS, AFS, GFS), SAN
- Метакомпьютеры и GRID системы, globus, condor
- Компиляторы и реализации библиотек для разработки параллельных и распределенных программ
- Распределенные и параллельные системы управления базами данных
- Средства создания параллельных программ для MS Windows (COM, Corba, .NET)
- Rear-to-rear системы
- Параллельные алгоритмы поиска, шифрования и дешифрования (2 години)
- Языки программирования с внутренним параллелизмом и поддержкой распределенных систем

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Лабораторные работы (6 работ)

- Работа в командной строке Linux (15.05.2012)
- Работа на удаленных машинах по SSH, RSH (16.05.2012)
- Распределенные системы имен (NIS) (17.05.2012)
- Сетевые файловые системы (NFS, amd) (18.05.2012)
- Менеджер ресурсов и менеджер заданий, кластер типа Beowulf (21.05.2011-22.05.2012)
- Запуск PVM и MPI на кластере Beowulf (23.05.2012)

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Практические занятия (6 занятий)

- Разработка программ на основе интерфейса socket
- Разработка многопоточных программ
- Разработка с использованием RPC
- Разработка расчетных MPI программ
- Разработка расчетных OpenMP программ
- Измерение производительности

ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

Литература

- Параллельные вычисления в России <http://www.parallel.ru>
- Обчислювальний кластер Київського національного університету імені Тараса Шевченка <http://www.cluster.kiev.ua>
- В.П. Гергель, Р.Г. Стронгин Основы параллельных вычислений для многопроцессорных вычислительных машин. Нижний новгород: Изд-во ННГУ им. Лобачевского, 2000, 176 с.
- К. Хьюз, Т. Хьюз. Параллельное и распределенное программирование с использованием C++. Перс. с англ. М: Издательский дом «Вильямс», 2004, 672 с.
- И. Н Молчанов. Введение в алгоритмы параллельных вычислений. — К.: Наукова Думка, 1990. — 128 с.
- Distributed information systems <http://www.iks.inf.ethz.ch/education/ws04/eai/>

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Что такое параллельные и распределенные вычисления?

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Определение параллельных и распределенных вычислений

- *Параллельные вычисления* – для вычисления **одновременно** используется **несколько физических устройств**
- *Распределенные вычисления* – вычисление выполняется в **нескольких адресных пространствах** (с помощью **нескольких процессов**)

ЮЦ КНУ
імені Тараса
Шевченка,

2005 р

Особенности параллельных вычислений

- Одновременная и не одновременная работа нескольких устройств
- Основное использование параллелизма
- Уровни параллелизма
- Сложности, связанные с параллелизмом
- Истинный и псевдопараллелизм

ІОЦ КНУ
імені Тараса

Шевченка,

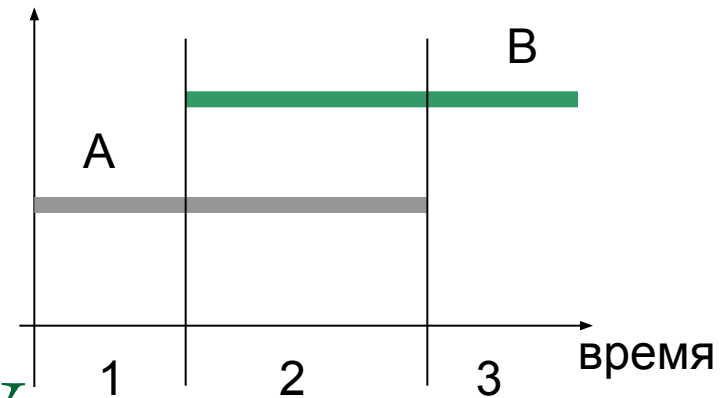
2005 р

Параллельно – значит одновременно

- В промежутки времени 1 и 2 «происходит» процесс А

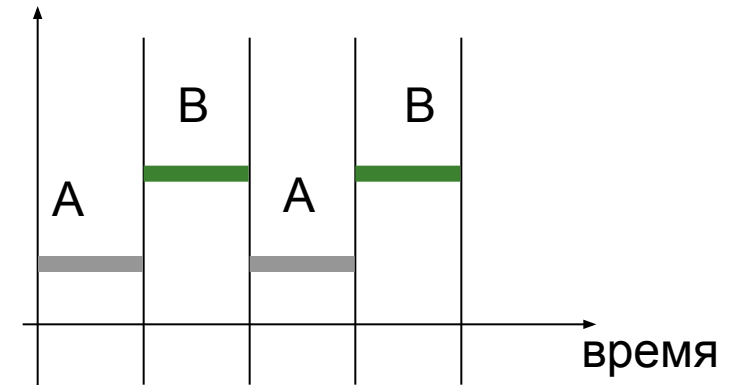
- В промежутки времени 2 и 3 «происходит» процесс В

- В промежуток времени 2 процессы А и В «происходят» одновременно, то есть параллельно



Не одновременно – значит не параллельно

- Когда процесс А выполняется, процесс В – не выполняется



- Процессы А и В выполняются не параллельно

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Примеры параллельного выполнения

- Двухпроцессорный компьютер
- Дисковый массив из нескольких дисков
- Заводской конвейер
- Бригада рабочих, которые копают яму
- Одновременная работа диска и процессора (ассинхронный режим)
- Параллельная работа нескольких видеоадаптеров

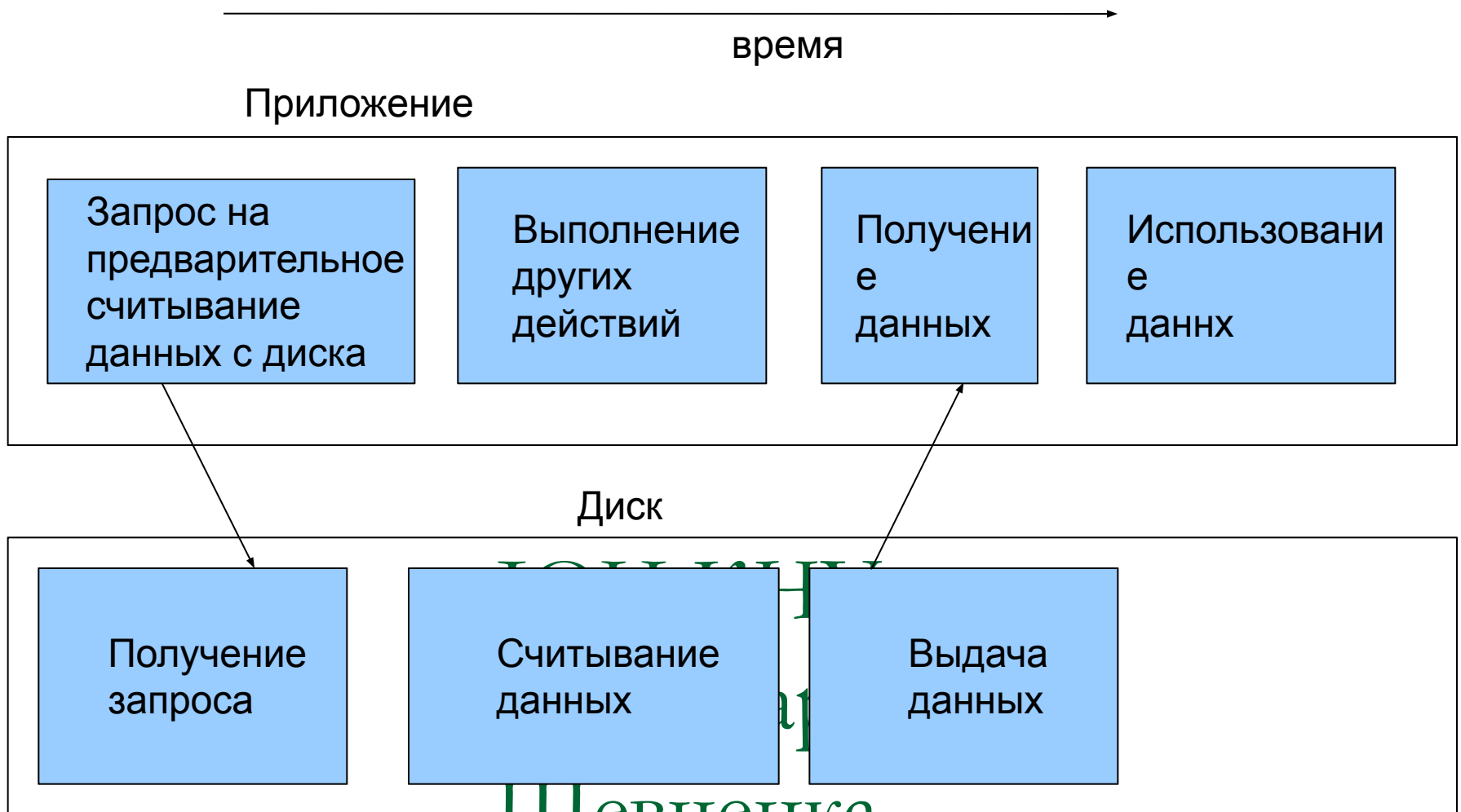
Двухпроцессорный компьютер

```
18:49:49 up 90 days, 23:22, 6 users, load average: 1,97, 2,
167 processes: 161 sleeping, 3 running, 1 zombie, 2 stopped
CPU states:  cpu    user    nice  system  irq  softirq  iowa
              total  197,0%  0,0%   2,2%   0,0%   0,0%    0,
```

Каждый процессор выполняет свою программу

ЮЦКНУ
імені Тараса
Шевченка,
2005 р

Асинхронный режим чтения диска



Шевченка,

2005 р

Примеры не параллельного выполнения

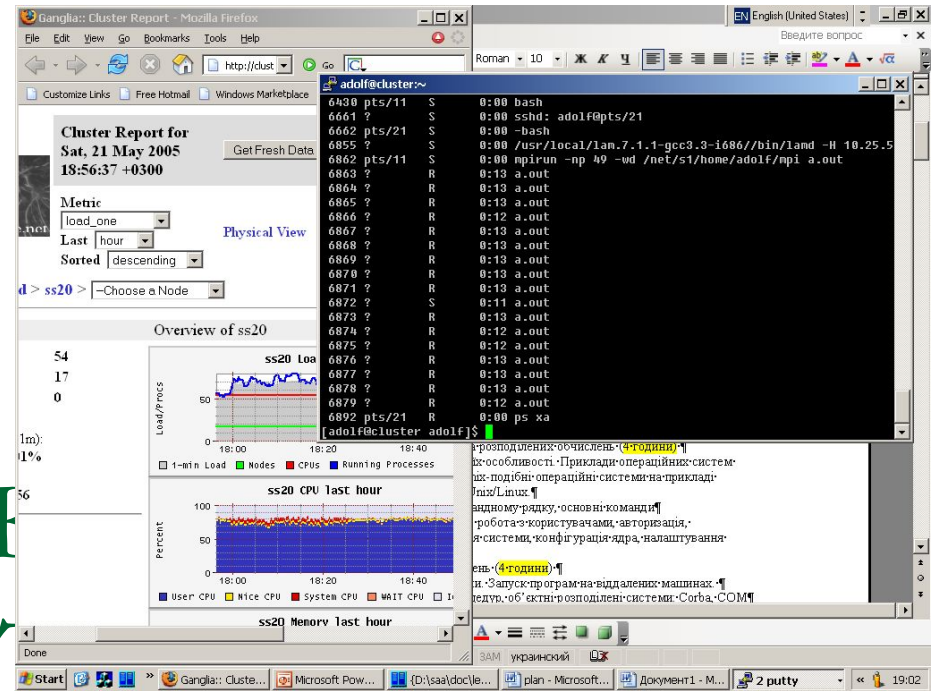
- Многозадачная операционная система с разделением времени
- Сеть Ethernet с общей средой передачи данных (СМАСD)
- Синхронный режим доступа к жесткому диску

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Операционная система с разделением времени

- Каждая программа получает свой квант времени
- Переключение между программами происходит быстро
- Кажется, что все программы выполняются одновременно
- На самом деле параллельно



ІОЦ Р
імені
Шевченка,
2005 р

Использование параллелизма

- Единственная цель - увеличение производительности

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Производительность

- Производительность – количество операций, которые выполняются в единицу времени
- Чем сложнее задача, тем большая производительность системы нужна для ее решения в обозримом времени
- Если увеличить количество операций, которые выполняются одновременно, то возрастет производительность системы

Пути повышения производительности

■ Интенсивные:

- Использование новых физических принципов построения компьютерных систем (оптические компьютеры, наноэлектроника, высокомолекулярная электроника)

■ Экстенсивные:

- Увеличение тактовой частоты устройств
- Использование параллельной обработки

ІОІІКНУ
імені Тараса

Шевченка,

2005 р

Новые технологии

- Наилучший вариант, но...
- Физические основы современных компьютерных технологий были разработаны лет 30 назад (физика полупроводников и диэлектриков)
- Новые физические методы станут технологиями примерно лет через 30

ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

Увеличение тактовой частоты

- Производительность пропорциональна тактовой частоте
- Увеличение тактовой частоты приводит к увеличению потребляемой мощности и к необходимости усиленного охлаждения
- Увеличение тактовой частоты приводит к возрастанию влияния паразитных обратных связей и к необходимости введения новых технических решения

Параллельные вычисления

- Если один рабочий выкопает яму за 1 час, то 2 рабочих – за 30 минут
- Если один процессор медленно..., то можно поставить 2, 3, 100 ... и будет быстро
- Можно повышать производительность без введения принципиально новых физических и технических решений
- Никаким другим методом сегодня нельзя достичь такого повышения производительности, как за счет параллельной обработки

ЮЦКНУ
имени Габаса
Шевченка,

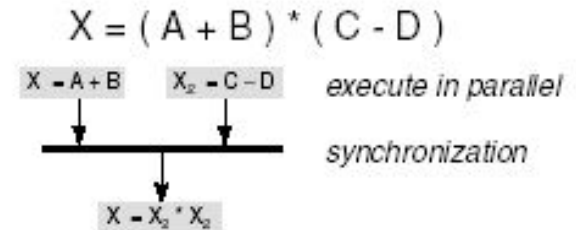
2005 р

Уровни параллелизма

- Уровень мелких структурных единиц (fine graine)
 - уровень инструкций
- На уровне средних структурных единиц
 - Уровень подпрограмм
- На уровне крупных структурных единиц (course graine)
 - Уровень объектов
 - Уровень приложений

Параллелизм на уровне машинных инструкций

- Две (или больше) машинных инструкций выполняется одновременно
- Суперскалярные и векторные процессоры
- Конвейеры
- Есть во всех современных процессорах (SSE, MMX)



ЮЦКНУ
імені Тараса

Шевченка,

2005 р

Параллелизм на уровне процедур

- Каждая процедура (функция, метод) выполняется на своем процессоре
- Используется при многопоточном программировании
- Поток – часть процесса, которая выполняется параллельно с другими такими же частями

ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

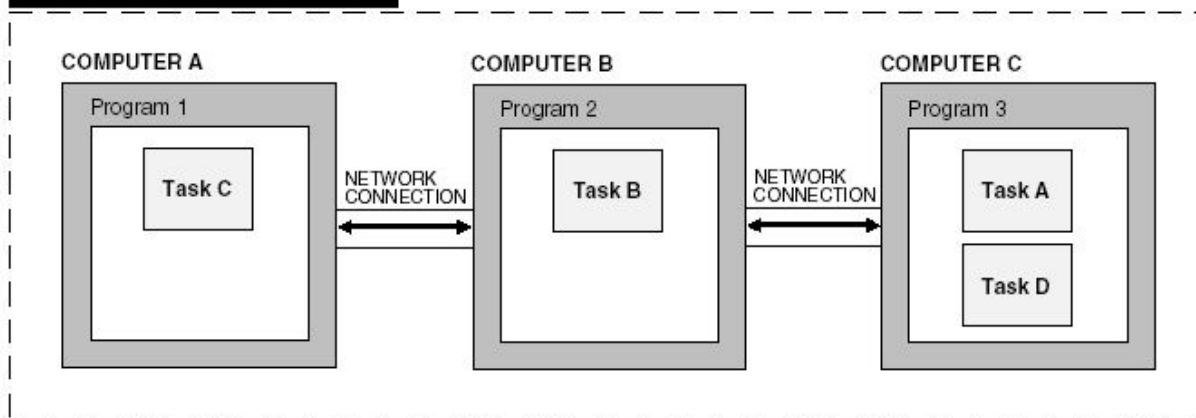
Параллелизм на уровне объектов

- Методы каждого объекта выполняются одновременно с методами других объектов
- Объект – это данные и те действия (методы, функции), которые с этими данными можно выполнять
- Используются в многопоточных программах и распределенных объектных системах (COM, CORBA)

Параллелизм на уровне приложений

- Каждое приложение выполняется на своем процессоре или на своем компьютере одновременно с другими приложениями
- Используется для кластерных вычислений и других распределенных систем

DISTRIBUTED APPLICATION



Какой уровень лучше?

- Для каждой задачи – свой
- Для повышения скорости вычислений — **повышать уровень**
- Для уменьшения задержек — **понижать уровень**
- Часто в одной и той же параллельной программе применяется сразу несколько уровней

Например, параллельная программа выполняется на 4-х узлах кластера – **уровень припожений**, на каждом узле используется многопоточная обработка – **уровень процедур**, а каждый поток выполняется на процессоре с конвейерной обработкой – **уровень инструкций**

Юрій КНУ
Імени Тараса
Шевченка,

2005 р

Сложности, связанные с параллелизмом

- Необходимость специальных параллельных алгоритмов
- Необходимость специальных параллельных программ
- Необходимость специальных аппаратных устройств для параллельных вычислений

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Параллельные алгоритмы

- Классическое определение: Алгоритм – последовательность операций, которую необходимо выполнить для решения задачи
- Параллельный алгоритм – последовательность нужно разбить на одновременно выполняемые последовательности - распараллелить
- Очень часто задача распараллеливания чрезвычайно сложна
- Иногда применяются свои уникальные «параллельные» подходы
- Не все алгоритмы можно эффективно распараллелить

ЮЦКНУ
имени Тараса
Шевченка,

2005 р

Декомпозиция, связь и синхронизация

- Каждый параллельный алгоритм имеет три составляющие:
 - Декомпозиция
 - Связь
 - Синхронизация

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Декомпозиция

- Декомпозиция – разбиение задачи на части, которые выполняются параллельно
 - Декомпозиция данных – данные, с которыми работает программа разбиваются на меньшие части и с каждой частью выполняются свои операции
 - Декомпозиция функций – последовательность действий разбивается на участки, которые выполняются параллельно

ЮИЦ КНУ
імені Тараса
Шевченка,

2005 р

Пример декомпозиции

- Расчет прогноза погоды для Украины
- Территория разбивается на более мелкие области и для каждой области выполняется расчет на своем процессоре, параллельно с остальными



2005 р

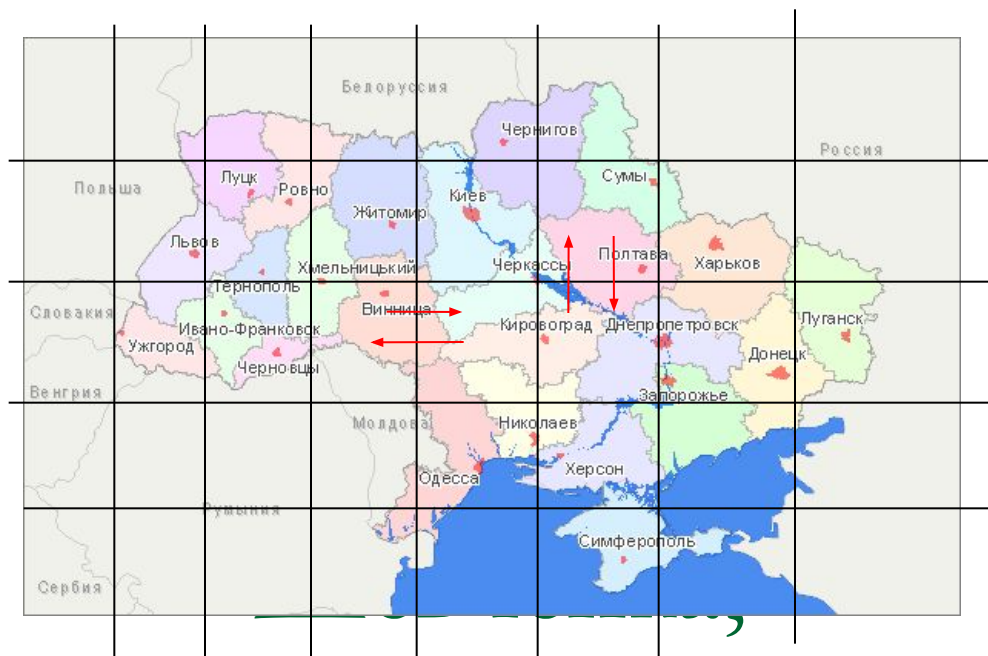
СВЯЗЬ

- Разные процессоры должны обмениваться между собой информацией

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Пример связи

- Расчет прогноза погоды для Украины
- Между соседними областями должен выполняться обмен информацией о состоянии погоды на границе областей



2005 р

Синхронизация

- Обеспечение того, что все параллельно выполняющиеся части в определенные моменты времени находятся в нужном состоянии
 - Например, задача решена, только когда все параллельно выполняющиеся части завершают свою работу
 - Чтобы данные, считанные из переменной корректными, нужно гарантировать, что их в эту переменную записали

Использование специальных параллельных алгоритмов

- Пример: найти сумму $S = a_1 + a_2 + \dots + a_N$
- `for (i=0; i<N; i++) S+=a[i];`
- В таком виде задача существенно последовательная
- Для распараллеливания воспользуемся ассоциативность сложения



Пример - конвейер

$$y = \sqrt{\ln(\sin(x))}$$

На входе 5 значений (X_1, X_2, \dots, X_5)

В системе 3 процессора, каждый выполняет свои дей

$$A_1 = \cos(X_1)$$

$$A_2 = \ln(A_1)$$

$$Y_i =$$

Пр1 (cos)

Пр2 (ln)

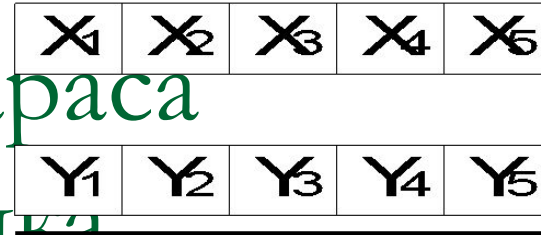
Пр3

ІОЦ КНУ

імені Тараса

Шевченка,

2005 р



Состояния конвейера

	Процесор 1 sin	Процесор 2 ln	Процесор 3 sqr	Y1	Y2
1	sin(X1)	-	-	-	-
2	sin(X2)	ln(sin(X1))	-	-	-
3	sin(X3)	ln(sin(X2))	sqr(ln(sin(X1)))	Y1	-
4	sin(X4)	ln(sin(X3))	sqr(ln(sin(X2)))	Y2	Y1
5	sin(X5)	ln(sin(X4))	sqr(ln(sin(X3)))	Y3	Y2

ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

Сложность

- Параллельный алгоритм получается значительно сложнее последовательного
- При небольшом количестве слагаемых или при большом количестве процессоров можно получить не выигрыш а проигрыш в скорости
- При очень большом количестве слагаемых и не очень большом количестве процессоров выигрыш в скорости будет существенным по сравнению с последовательным случаем
- Эффективность распараллеливания зависит от задачи

ЮЦКНУ
імені Тараса
Шевченка,

2005 р

Специальные параллельные программы

- Последовательная программа выполняется на одном процессоре, потому не получает никакого преимущества от параллельного выполнения
- Для параллельных программ кроме самих вычислений необходимо реализовать связь и синхронизацию
- Необходимо реализовать декомпозицию
- Для упрощения существуют специальные компиляторы и библиотеки
- Сложность отладки и профилирования

Аппаратные средства параллельных вычислений

- Для параллельных вычислений нужно несколько процессоров или компьютеров
- Несколько процессоров/компьютеров всегда в сумме дороже, чем один процессор/компьютер
- Необходимо обеспечение высокоскоростных каналов связи между процессорами/компьютерами
- С увеличением количества и сложности оборудования часто уменьшается его надежность

ЮЦ КНУ
імені Тараса
Шевченка,

2005 р

Примеры параллельных систем



ЦОЦ КН

имені Тараса

Кластер Киевского национального
Университета имени Тараса Шевченка

Кластеры Института кибернетики
НАН Украины имени Глушкова

2005 р

Законы Гроша (Grosch) и Мура (Moore)

- Закон Гроша: Производительность компьютера пропорциональна квадрату стоимости компьютера (сейчас уже не работает)
- Закон Мура: Производительность последовательных процессоров возрастает в два раза каждые 18-24 месяца

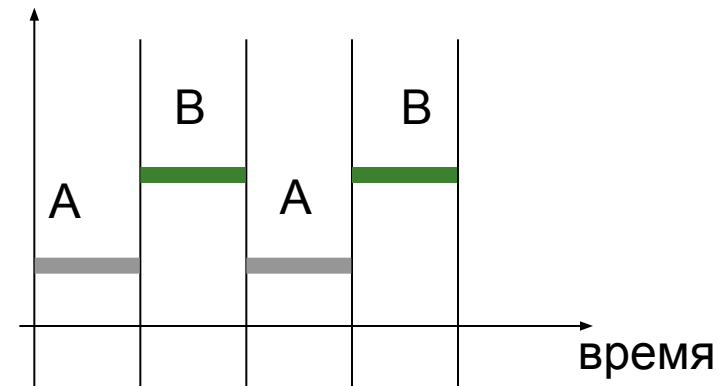
ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

Истинный и псевдопараллелизм

- Для многозадачных операционных систем с одним процессором одновременного выполнения получить нельзя, но кажется, что задачи выполняются одновременно



- В такой ситуации проблемы параллелизма остаются, а повышения производительности нет - псевдопараллелизм

Параллелизм и конкуренция

- Concurrent и Parallel - синонимы
- Параллелизм – одновременность выполнения задачи
- Конкуренция – одновременность использования ресурсов

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Выводы относительно параллелизма

- Производительность последовательных ЭВМ не может возрасти до бесконечности
- Единственный способ получить чрезвычайно высокую производительность на существующем техническом уровне – это использовать параллельные вычисления на уровне инструкций, процедур, объектов, приложений
- Современные (даже последовательные компьютеры) используют параллелизм
- Использование параллельных вычислений ведет к удорожанию оборудования
- Параллельные вычисления требуют разработки специальных алгоритмов и использования специальных средств программирования
- Не все задачи можно эффективно распараллелить

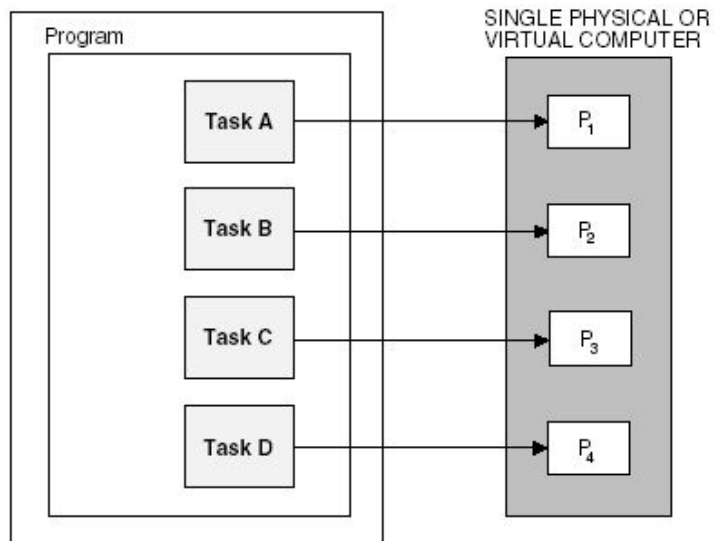
Распределенные вычисления

- Вычисление выполняется в **нескольких адресных пространствах** (с помощью **нескольких процессов**)
- Процесс (task) – единица выполнения задания, которая включает выполняющийся код и ресурсы, которые это код использует и которые защищены от доступа других процессов
- Адресное пространство – это то, как память и другие ресурсы представляются процессу

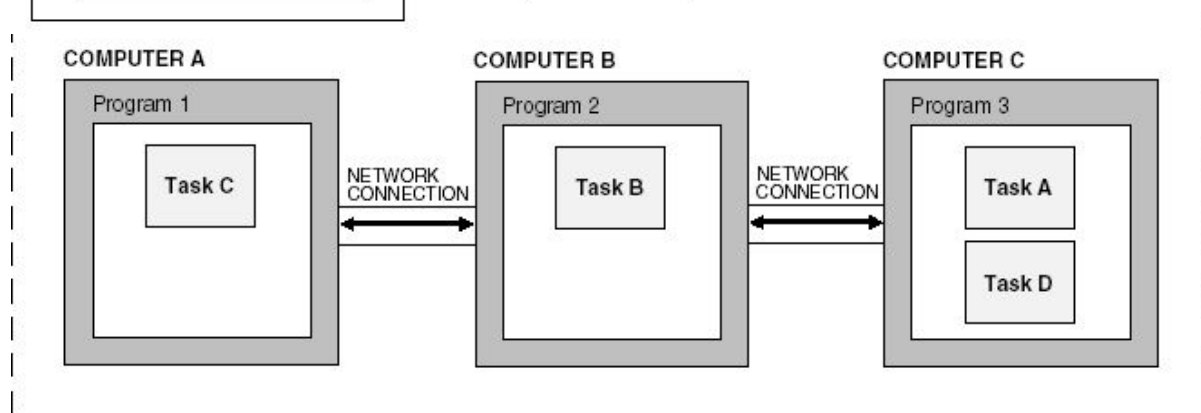
ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Распределенные программы



Несколько процессов выполняются на одной машине и служат для решения общей задачи



Программы выполняются на разных машинах и служат для решения общей задачи

Шевченко,
2005 г

Преимущества распределенных систем

- Возможность использования ресурсов, которые находятся на разных аппаратных платформах или принадлежат разным программам
- Возможность специализации ресурсов
- Возможность децентрализации
- Возможность создания избыточности ресурсов для повышения надежности

ЮЦ КНУ
імені Тараса
Шевченка,

2005 р

Использование ресурсов, которые находятся на разных компьютерах

- Доступ к удаленным сетевым ресурсам
- Предоставление доступа пользователям других машин к своим ресурсам
- Распределенные вычисления сейчас стали синонимом слова Интернет

ІОЦ КНУ
імені Тараса
Шевченка,

2005 р

Специализация

- Если есть ресурс, который необходим большому количеству людей, то его необязательно размещать на компьютерах всех пользователей, которым он нужна
- Можно создать один ресурс, к которому будет выполняться доступ многих пользователей

ІОЦ КНУ
імені Тараса

Шевченка,

2005 р

Децентрализация

- Данные очень большого объема можно разнести по нескольким физическим системам
- Пример: большие поисковые системы

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Обеспечение надежности

- Создается несколько копий одного ресурса и в случае выхода из строя одной копии, все остальные будут доступны
- Пример: несколько копий базы данных, или несколько задач, которые выполняют одни и те же действия

ІОЦ КНУ

імені Тараса

Шевченка,

2005 р

Сложности

- Декомпозиция, связь, синхронизация
- Усложнение программирования

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

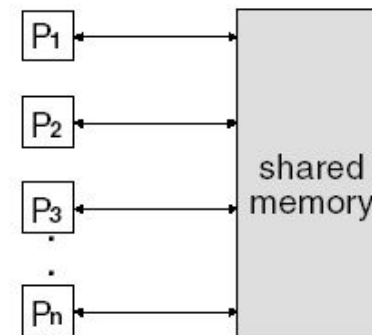
Параллельные и распределенные вычисления

- Много общего в целях и подходах
- Не все параллельные вычисления являются распределенными
- Не все распределенные вычисления являются параллельными

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Система с совместно используемой памятью

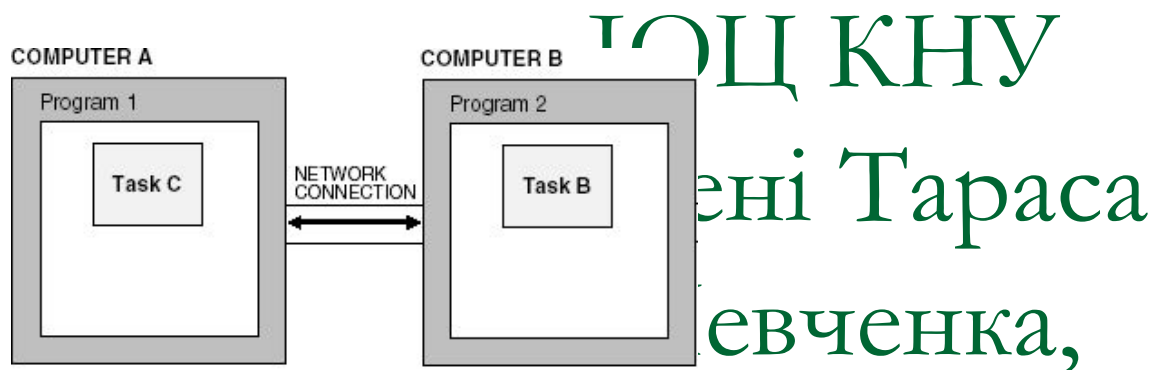
- Параллельная, но не распределенная
- Все программы могут совместно использовать одни и те же данные



ІОЦ КНУ
імені Тараса
Шевченка,
2005 р

Система с зеркалированием

- Компьютер А выполняет работу, а компьютер В является резервным на случай выхода компьютера А из строя
- Компьютеры не работают одновременно



2005 р

Выводы

- Параллельные и распределенные вычисления позволяют решать проблемы производительности, надежности и обеспечения доступа к ресурсам
- Тем не менее использование параллельных и распределенных вычислений требует усложнения алгоритмов, программирования и аппаратных средств
- У параллельных и распределенных вычислений много общего, но есть некоторые отличия

Вопросы

ІОЦ КНУ
імені Тараса
Шевченка,
2005 р
