Международная конференция GRID`2008
г.Дубна, 29.06.08 - 4.07.08

ПРОГРАММИРОВАНИЕ СТРУКТУРНО СЛОЖНЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В НЕДЕТЕРМИНИРОВАННОЙ КОМПЬЮТЕРНОЙ СРЕДЕ[1]

Ю.С. Затуливетер

Институт Проблем Управления им. В.А. Трапезникова РАН

zvt@ipu.rssi.ru

[1] *Работа поддержана Российским фондом фундаментальных исследований (РФФИ), проект 08-07-00334.*

Проблемы глобализации вычислительного пространства

Системообразующие основы компьютерной индустрии:

- логический базис универсальных машинных вычислений в модели Дж.фон Неймана (общесистемное ядро для HW и SW)
- экспоненциальный прогресс технологии производства компьютерных компонентов (прежде всего СБИС, закон Мура)

Опережающее развитие микропроцессорных технологий привело к

- формированию глобальной компьютерной среды (~10^9 универсальных компьютеров)
- массовому производству многоядерных микропроцессорных кристаллов (~ 10^9 транзисторов)

На высшие приоритеты выходят задачи глобально распределенной и высокопараллельной обработки информации в *массовых* сферах жизнедеятельности: социальная самоорганизация, наука, образование, здравоохранение, бизнес, госуправление, экология и т.д.

Действие классической модели не распространяется на новые выч. среды. Прежнее общесистемное ядро уже не отвечает новым требованиям задач и возможностям технологий по их решению

Проблемы глобализации вычислительного пространства

Налицо глубинный системный кризис:

- компьютерная среда из 10^9 *универсальных* компьютеров не обладает свойством универсальной программируемости
- компьютерная индустрия почти исчерпала системообразующий потенциал классической аксиоматики
- сохранение неадекватного системного ядра ведет к обесцениванию прогресса технологий и торможению процессов освоения задач массовой обработки глобально распределенной информации
- удержание темпов компьютерного прогресса в условиях отсутствия адекватного системного ядра требует преодоления комбинаторной сложности проблем интеграции разнородных решений

Общедоступные системные возможности неадекватны опережающему спросу на новые задачи массового применения.

Обновление компьютерных основ неизбежно, но при этом должна сохраниться преемственность с классической аксиоматикой

В работе выделены и рассмотрены лишь два фундаментальных аспекта общей проблемы глобализации вычислительного пространства:

- построение универсального компьютерного базиса программирования и исполнения распределённых вычислений высокой структурно-динамической сложности, действие которого распространимо на любые совокупности связанных сетями компьютеров
- изучение потенциальных возможностей надежного исполнения распределенных вычислений в изначально ненадежной компьютерной среде

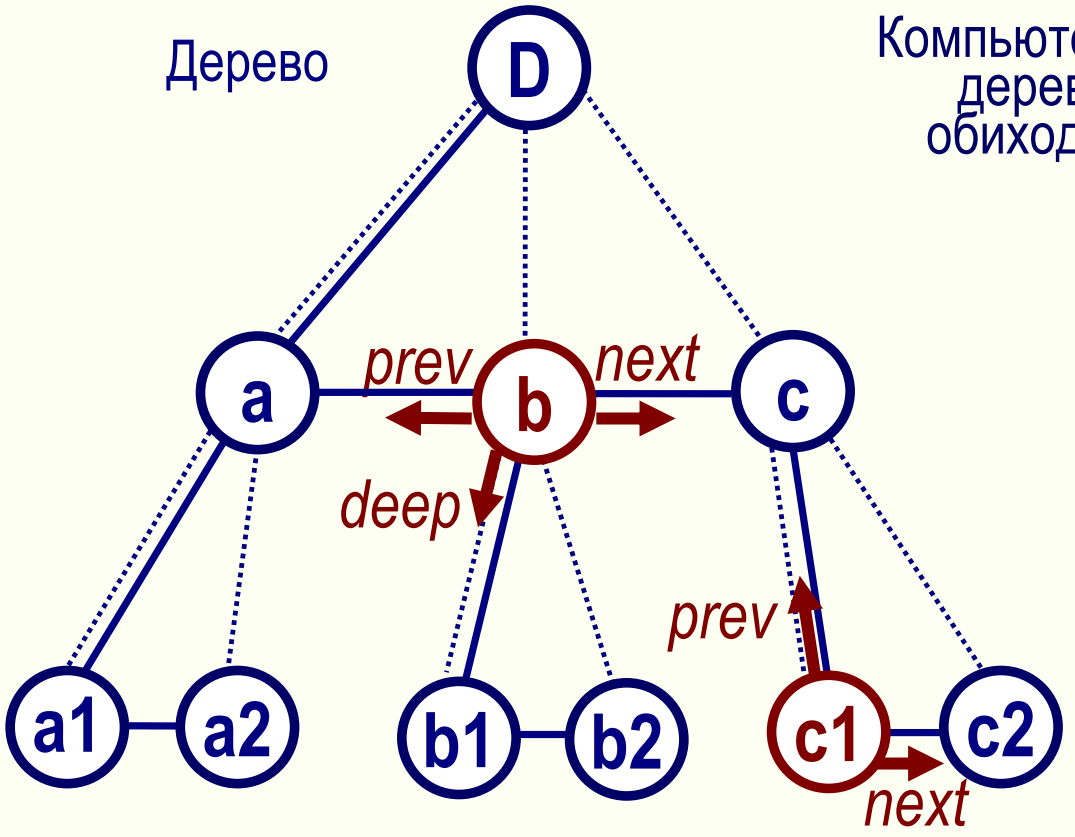
Исчисление древовидных структур(ИДС)

(Новый базис компьютерных операций для решения структурно сложных задач)

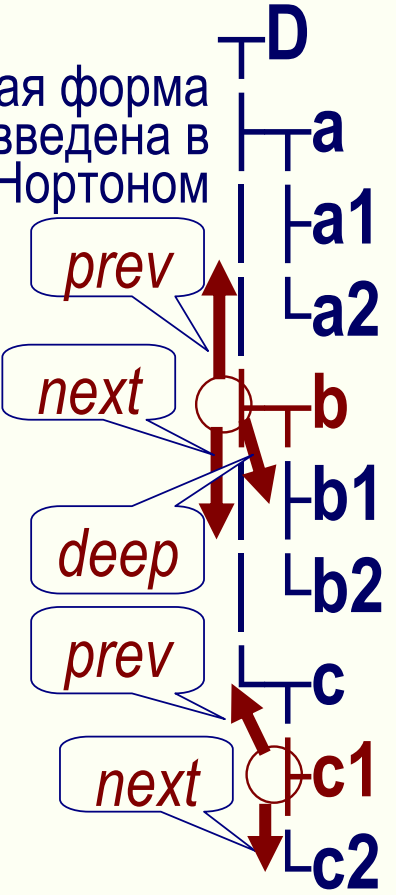
- **Двоичные деревья**
  - связи между вершинами
  - содержимое вершин
  - интерпретация деревьев
  - виды структур деревьев
- **Базисный набор действий**
- **Способы реализации ИДС**

Двоичные деревья

Связи между вершинами дерева



Компьютерная форма дерева, введена в обиход П.Нортоном



Базовый постулат:

Двоичные деревья, и только они являются математически замкнутой формой представления компьютерной информации (как данных, так и программ)

Содержимое вершин

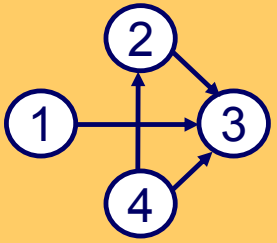
- **Строки произвольной длины**
  - двоичные (машинные) строки
  - символьные строки
- **Числовые форматы**
  - целые числа
  - числа с плавающей точкой
- **Адресные указатели вершин**

Примеры интерпретации в качестве данных

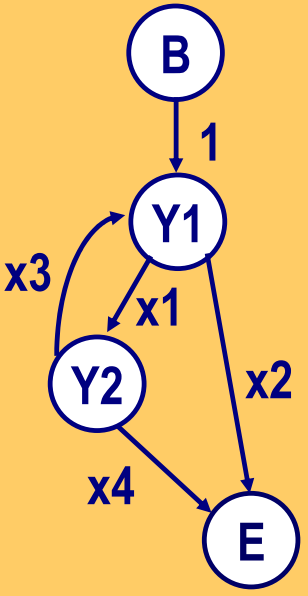
Матрицы	Графы	Автоматы

$$A=\begin{pmatrix} a11,a12,a13 \\ a21,a22,a23 \end{pmatrix}$$

```
A
  A1
    a11
    a12
    a13
  A2
    a21
    a22
    a23
```



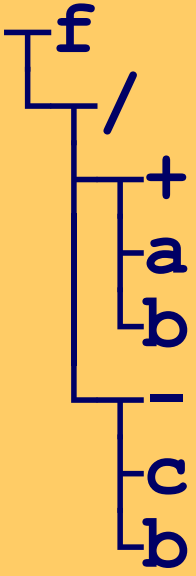

```
Граф
  1
    3
  2
    3
  4
    2
    3
```

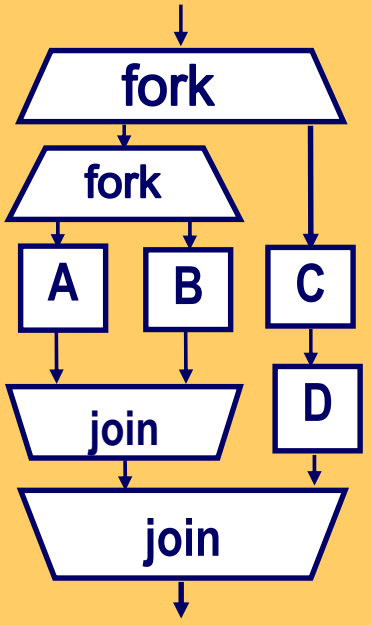


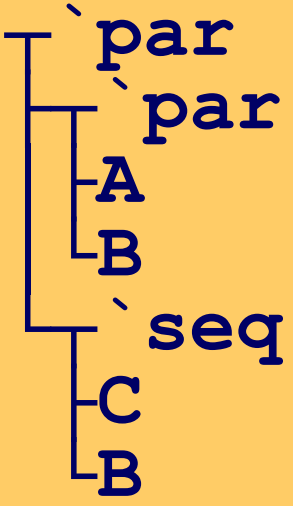

```
Q
  B
    1
    Y1
  Y1
    x1
    Y2
    x2
    E
  Y2
    x3
    Y1
    x4
    E
```

Примеры интерпретации в качестве программ

Формулы	Параллельные схемы
f=(a+b)/(c-d)	







Формулы, параллельные схемы

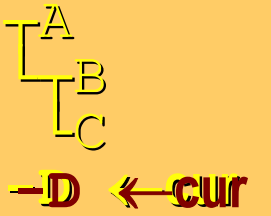
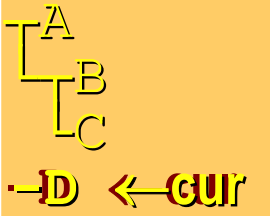
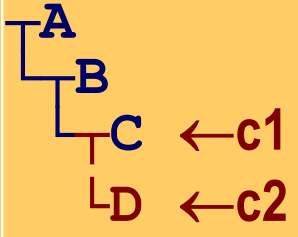
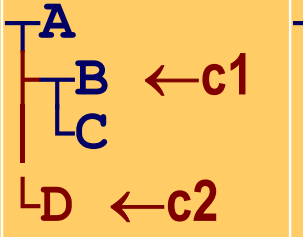
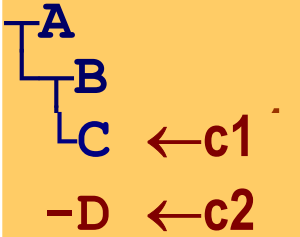
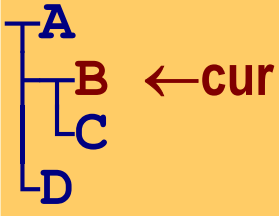
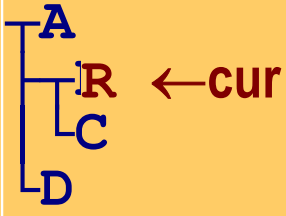
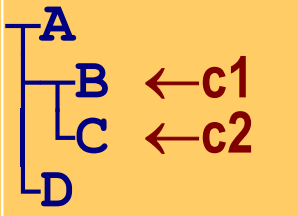
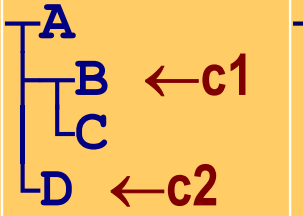
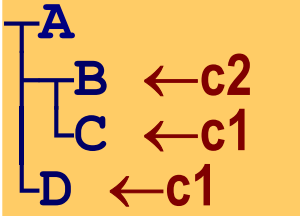
Виды структур деревьев

- **Регулярные деревья**
  - **преобладание структурной симметрии**
  - **однородность типов содержимого вершин**
  - **примеры:**
    - *вектор*
    - *матрица*
    - *многомерный массив*
- **Нерегулярные деревья**
  - **отсутствие или слабая симметрия структур**
  - **разнородность типов содержимого вершин**
  - **примеры:**
    - *текстовые структуры*
    - *организационные структуры*
    - *структуры векторной графики*
    - *большие хранилища данных (СУБД)*
    - *динамические структуры вычислительных процессов*

Состав базиса

- **Переменные и их значения**
  - **курсорные переменные** (*адресные указатели на вершины*)
- **Действия**
  - **с курсорными переменными**
    - *создание/удаление вершин*
    - *создание/удаление связей между вершинами*
    - *перемещение между вершинами*
    - *доступ к содержимому вершин*
  - **обработки содержимого вершин**

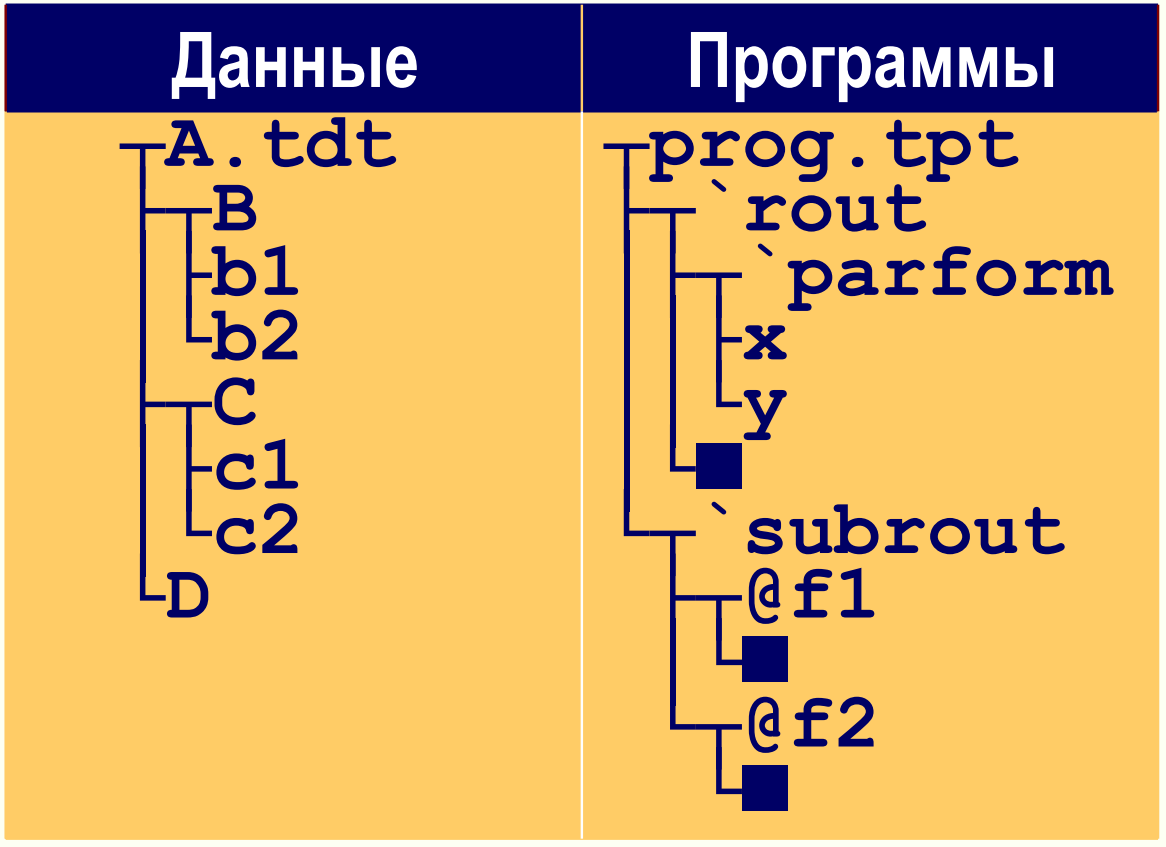
Действия с курсорными переменными

Создание/Удаление вершин		Создание/Удаление связей вершин		
cur=node_create("D")	cur=node_del(cur)	link_deep(c1,c2)	link_next(c1,c2)	link_del(c1,c2)
				
Доступ к значениям вершин		**Перемещение курсора**		
B=cont(cur)	re_cont(cur,"R")	c2=deep(c1)	c2=next(c1)	c2=prev(c1)
				

Обработка содержимого вершин

- **Значения содержимого вершин – константы**
- **Тип значения – тип константы**
- **Узнавание типа значения**
- **Чтение значений из вершин по адресным указателям**
- **Вычисление значений**
  - арифметические действия
  - логические действия
  - действия со строками
  - сравнение
  - преобразование значений при изменении типа
- **Результат**
  - занесение в вершину по адресному указателю
  - суперпозиция значения в качестве операнда или аргумента

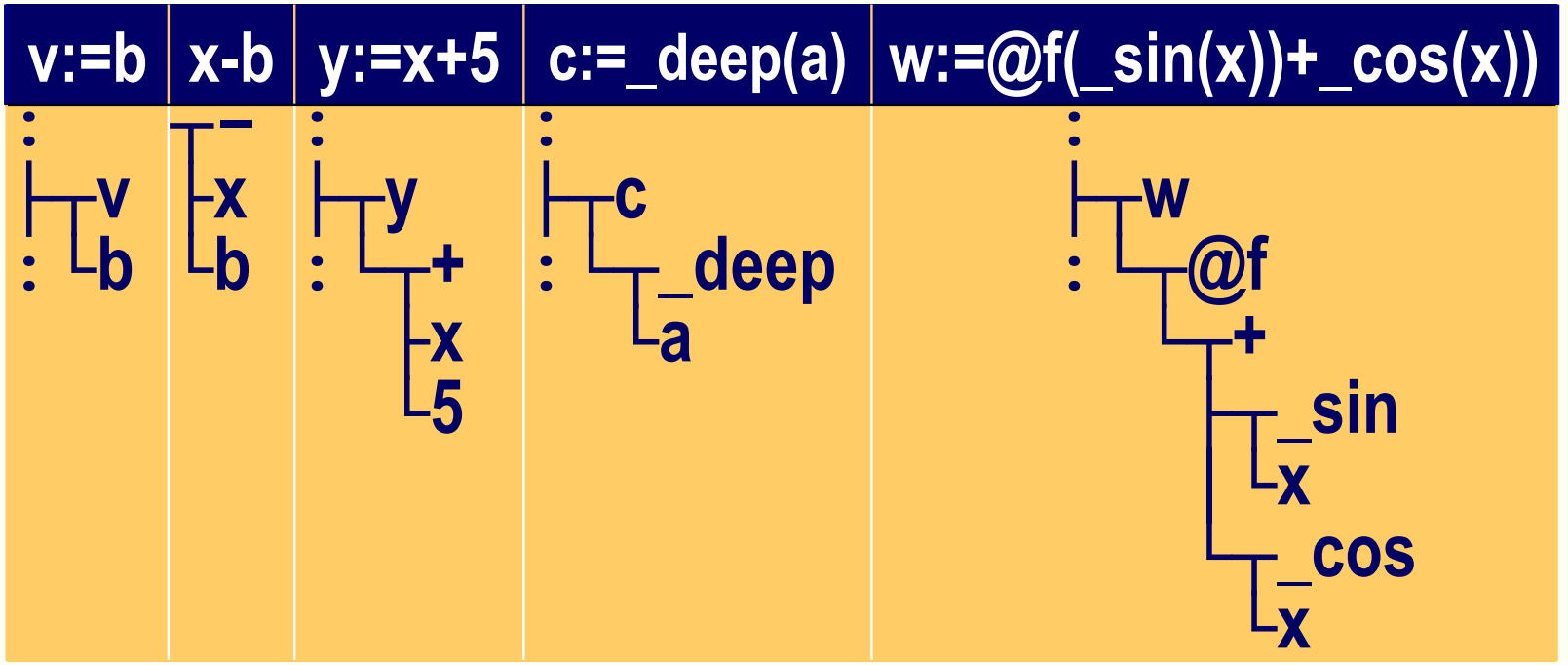
Язык Парсек - процедурная реализация исчисления древовидных структур



Элементы языка

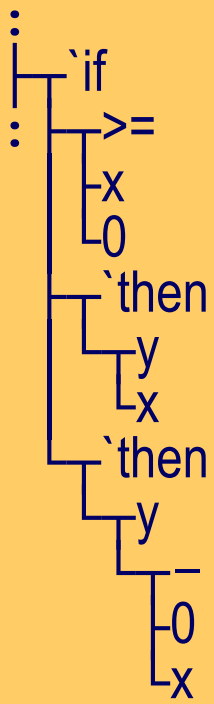
- **Константы**
  - числа (#целые, вещественные)
  - строки ($двоичные, \"символьные")
  - адресные указатели
- **Переменные и их значения**
  - курсорные (адресные указатели)
  - вычисляемые (числа, строки)
- **Операции**
  - арифметические: +, - , *, /, %
  - логические: &, |, ^
  - строковые (конкатенация, сдвиги, логические поразрядные)
  - предикатные: ==, <>, <, >, =<, >=
- **Функции**
  - встроенные (расширяемая библиотека: _sin(x), _sqrt(x), …)
  - определяемые (подпрограммы: @f(x,y,z), …)

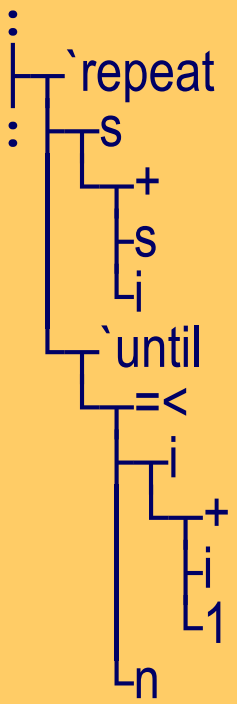
Выражения

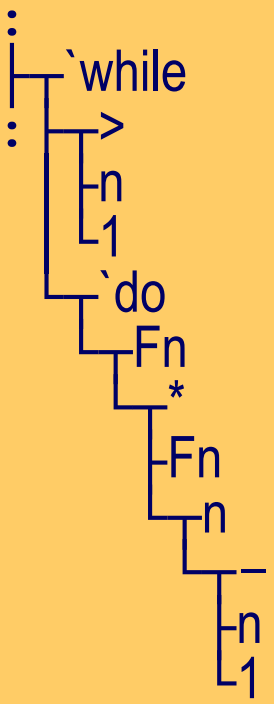


Управляющие конструкции

`if	`repeat	`while	`case
y=abs(x)	s=1+2+…+n	Fn=fact(n)	Почтовый фильтр

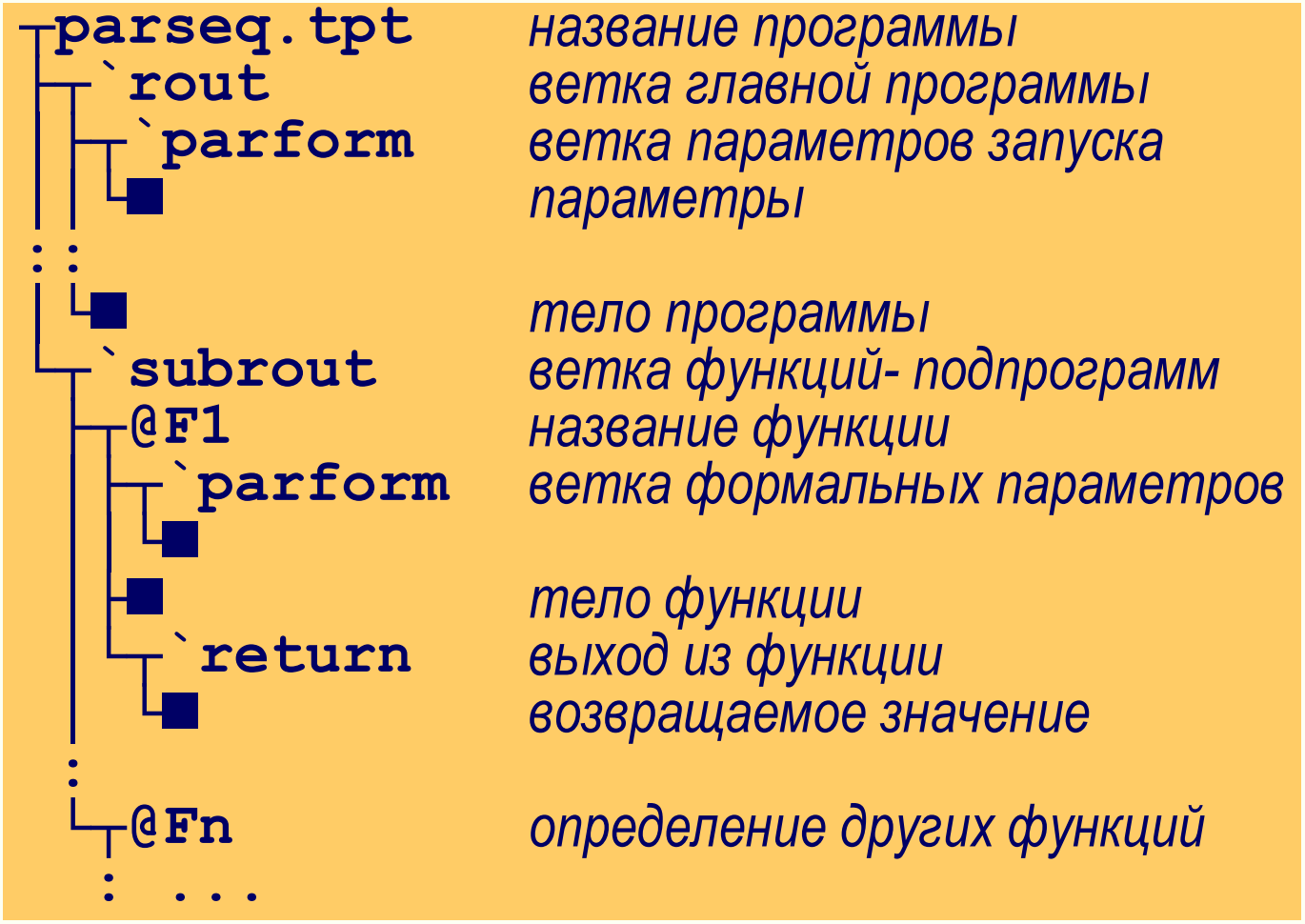






`continue, `break

Структура программы



Встроенные функции

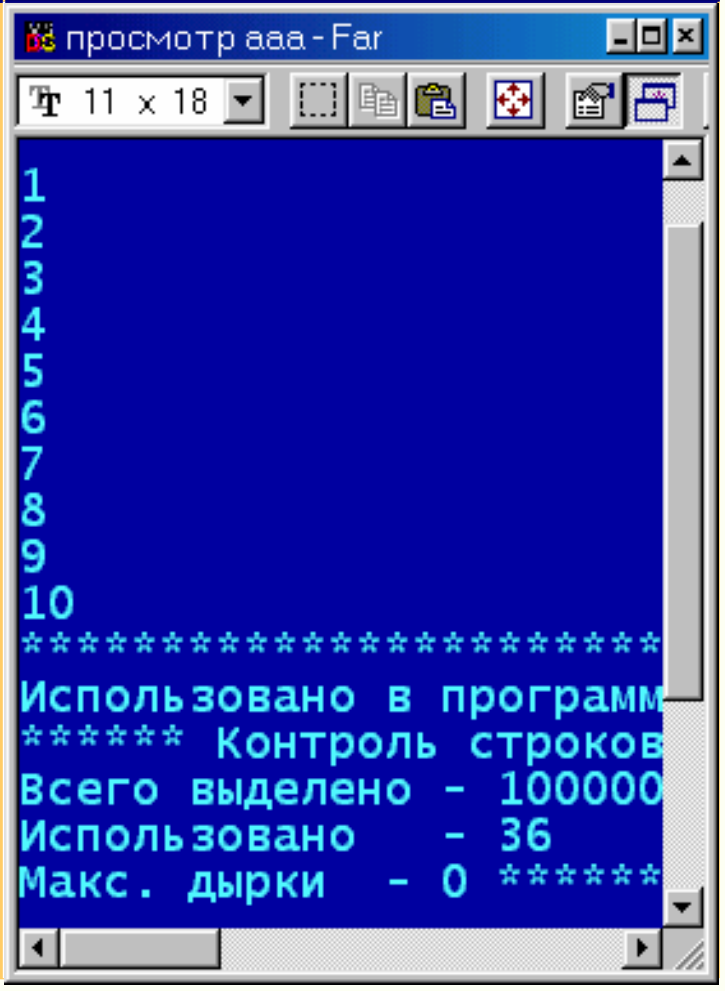
- **Функции на курсорных переменных**
- **Числовые функции**
- **Функции со строками**
  - определение длины
  - селекторы подстрок (чтение, замена)
- **Работа с типами значений**
  - определение типов
  - преобразование типов
- **Ввод/вывод** (файлы, клавиатура, монитор)
- **Управление процессами**
  - создание процесса
  - ожидание завершения
  - защита разделяемых ресурсов
- **Управление сетевыми взаимодействиями**

Обегание вершин деревьев

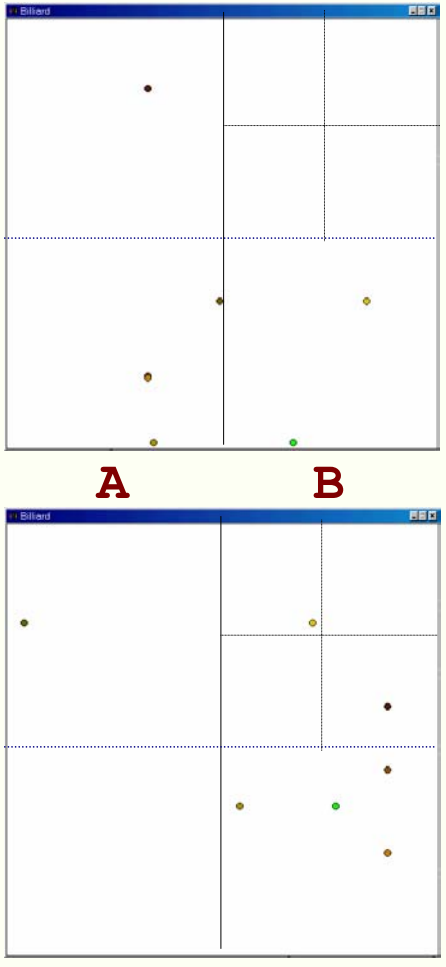
Программа	Данные	Результат

```
treetour.tpt
  `rout
    `parform
    filename
    @do_subtree
      _read_tree
      filename
  `subrout
    @do_subtree
      `parform
      subroot
      `if
      subroot
        `then
          _print
          {subroot}
          @do_subtree
            _deep
            subroot
          @do_subtree
            _next
            subroot
```

```
1
2
3
4
5
6
7
8
9
10
```



Движение упругих частиц на плоскости



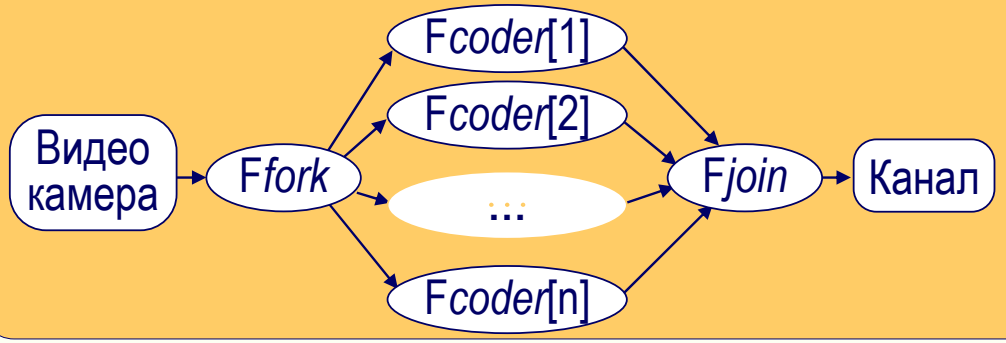

```
Q7.tdt
|Все шары
 #1
  0    X
  2   dX
  399  Y
  2   dY
 :
 #7
  399  X
  2   dX
  0    Y
  2   dY
```

```
Q5.tdt
 A
  #1
   X1
   dX1
   Y1
   dY1
  #3
   X3
   dX3
   Y3
   dY3
  #4
   X4
   dX4
   Y4
   dY4
 B
  #2
   X2
   dX2
   Y2
   dY2
```

Сжатие видео в реальном времени в ЛВС

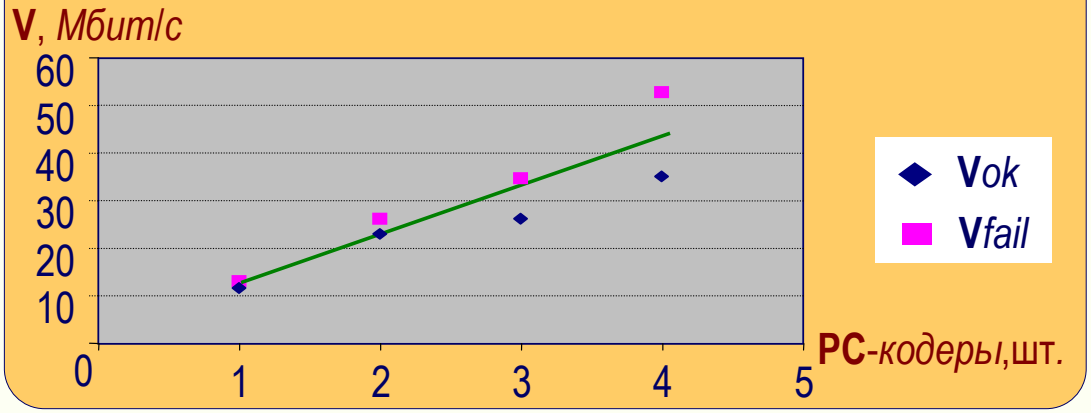
(Совместно с ООО ИДМ, г.Зеленоград)

Распределенный кодер сжатия цифрового видео



ЛВС	100Мбит/с
PC {Ггц}	3 – 6 шт. {0.8, 2.4(3),1.7(2)}
Входной поток	10-50 Мбит/с
Сжатый поток	≈0.3 – 4. Мбит/с

Зависимость вх.потока данных V от числа кодеров



Видеокамера	
Характеристики кадров	**Значение**
Размер *(в пикселях)*	176×144 320×240 352×288 640×480
Частота, (*кадр/с*)	5,10,15,20,30

Приборы и системы №1, 2006 — **Разработка и испытание системы распределенных вычислений для сжатия цифрового видео в реальном времени**

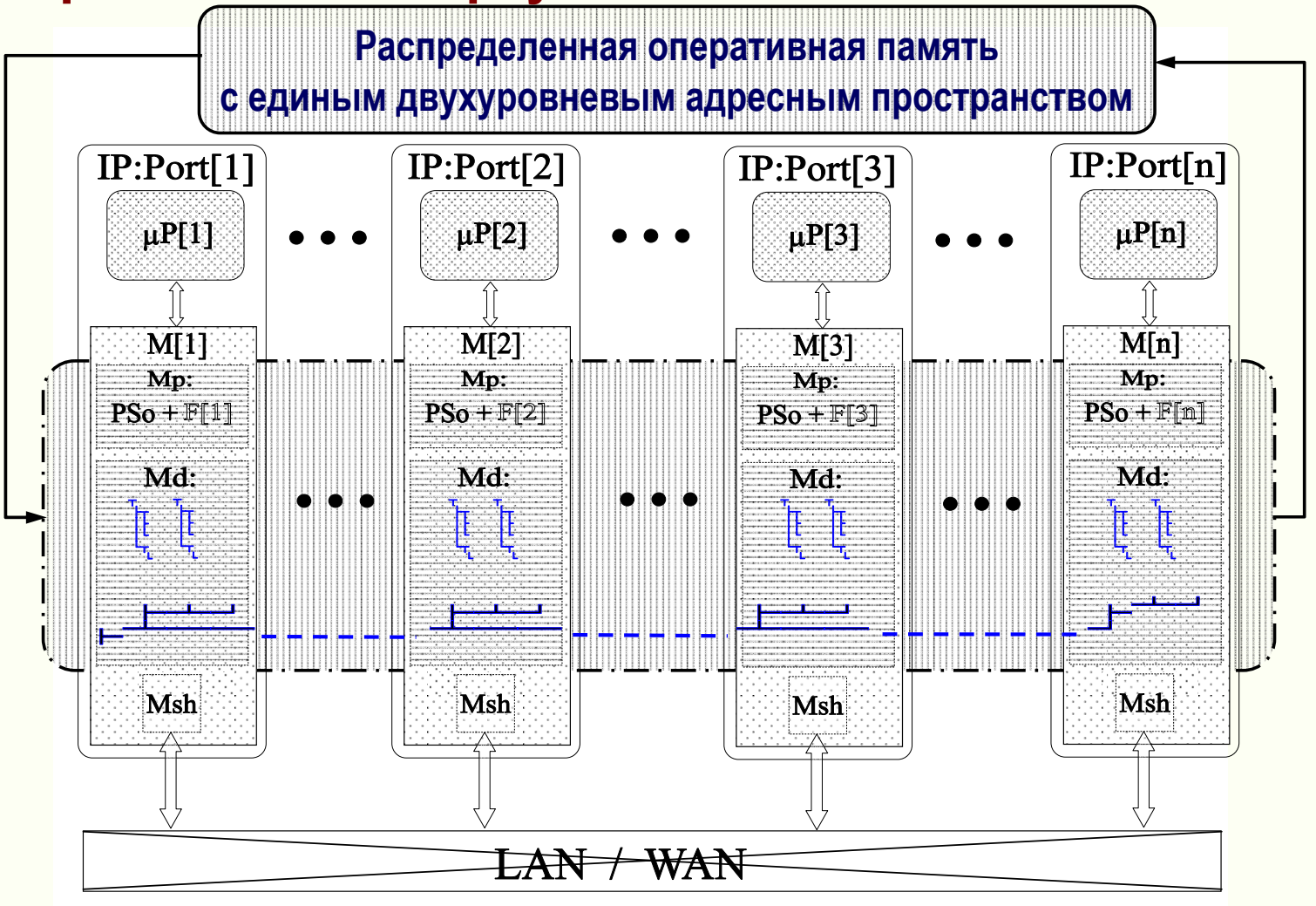
Классы задач

- **Задачи с нерегулярными структурами**
  - обработка текстовой информации
  - сортировка слиянием
  - моделирование
  - СУБД
  - индуктивный синтез программ по частным примерам
  - векторная графика
  - распределенная система сжатия видео в реальном времени
- **Сферы широкого и конкурентоспособного применения**
  - кардинальное снижение комбинаторной компоненты сложности задач интеграции глобально распределенных данных, программ, процессов и систем
  - глобально распределенные вычисления
  - глобально распределенные СУБД
  - глобально распределенные системы с управлением

Особенности языка

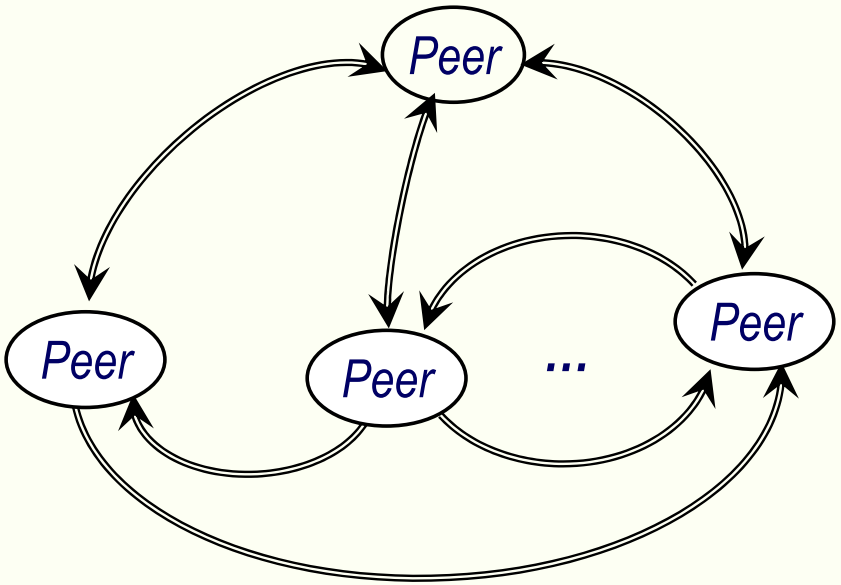
- **Структуры данных и программ: деревья и только деревья** (*и программы, и данные без go_to*)
- **Простой процедурный язык для решения структурно сложных задач** (*деревья обеспечивают структурную целостность посредством минимальной структурной сложности*)
- **В отличие от других процедурных языков построен по универсальной математически замкнутой модели ИДС**
  - структурная и функциональная целостность на математическом уровне базиса компьютерных операций
  - математическая однородность поля компьютерной информации
  - автоматическое и эффективное управление ресурсами отдельного компьютера и всех компьютеров сети
  - инвариантность логики программирования относительно конфигураций машинной среды (манипуляции не с ресурсами, а с деревьями - математическими объектами)

Распределенная виртуальная ПАРСЕК-машина

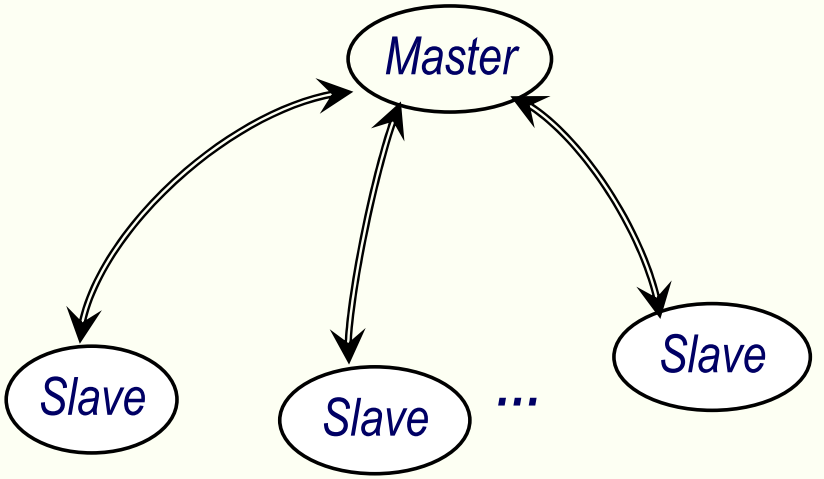


RPC – отдаленный запуск процедур в сквозном адресном пространстве распределенной оперативной памяти

Обмены данными между распределенными процедурами



a) Общий случай: произвольные прямые обмены данными (сетевая архитектура P2P)



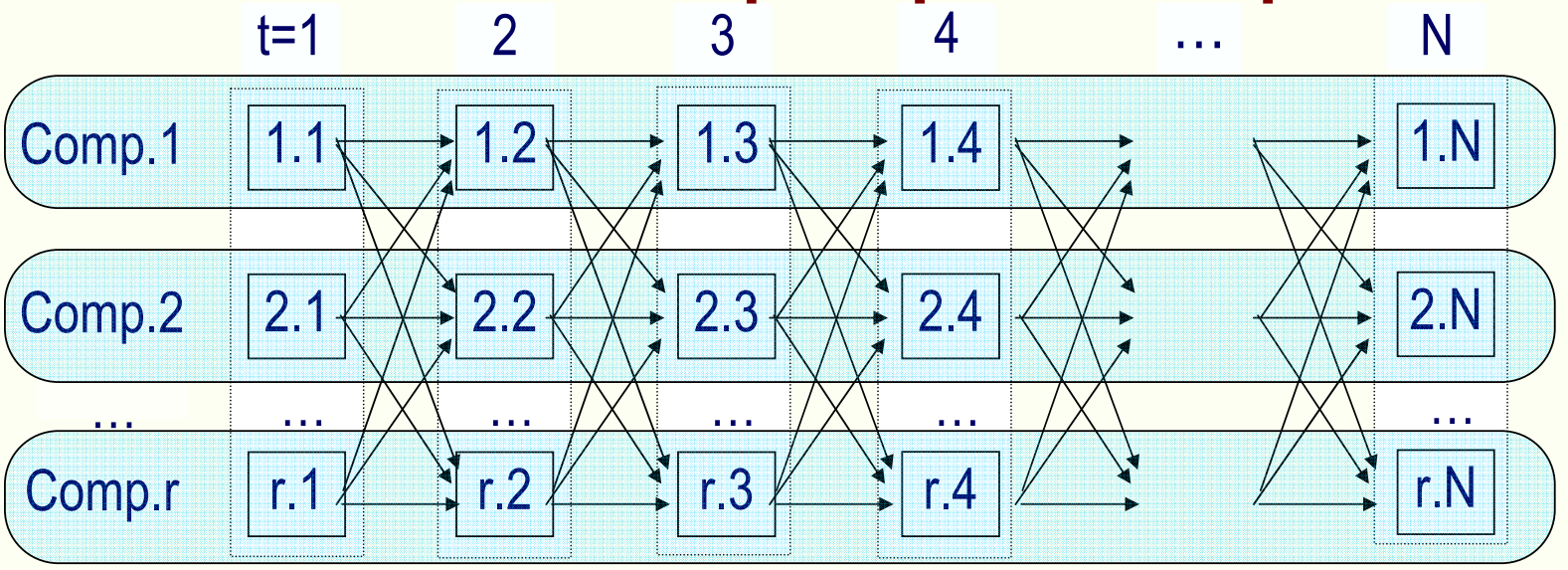
b) Частный случай: обмены данными через головной компьютер (сетевая архитектура "клиент-сервер")

Программно реализуемые методы повышения надежности распределенных процессов вычисления и управления в ненадежной компьютерной среде

Недетерминированность среды, избыточность вычислений, модели резервирования

- **Принципиально неустранимая недетерминированность глобальной компьютерной среды**
  - внутренние причины (техническая ненадёжность)
  - внешние причины (привносимые)
- **Повышение надежности за счет избыточных вычислений**
  - идентичные взаимопроверяющие процессы (резервирование)
  - пороговый метод определения достоверных значений
- **Модели**
  - резервирование с заменой выбывающих (верхние оценки)
  - резервирование без замены выбывающих (нижние оценки)
  - ситуационные модели (противодействие деструктивным воздействиям)

Избыточные вычисления: идентичные взаимопроверяющие процессы



Пороговый метод определения достоверных значений

- на каждом вычислительном шаге t=1,2, …, N производится обмен вычисляемыми значениями контрольных m разрядных слов (m>1)
- определяется максимальное количество совпавших значений
- критерий достоверности: максимальное число совпавших слов должно быть не меньше задаваемого порога d (1< d < r)
- компьютеры от которых пришли недостоверные контрольные слова считаются отказавшими

Модели

- **Резервирование с заменой выбывающих**
  - отказавшие компьютеры заменяются на работоспособные (с передачей через сеть текущего состояния процесса)
- **Резервирование без замены выбывающих**
  - процесс продолжается без выбывающих до тех пор, пока число работоспособных компьютеров не меньше заданного порога
- **Ситуационные модели (надежность обеспечивается также и противодействием деструктивным воздействиям**
  - переносом текущего состояния процессов через сеть на компьютеры отдаленные от деструктивных воздействий
  - пространственное перемещением компьютеров в безопасные места

Связываемые параметры процессов и моделей

- **N** – количество вычислительных шагов t=1,2,…,N
- **r** – кратность резервирования
- **n** – разрядность контрольного слова (n>1)
- **d** – порог достоверности (1<d<r)
- **p** – вероятность безотказной работы компьютера (отказ q=1-p)

Предварительные оценки: простейшая модель резервирования с заменой выбывающих (d=1, n=0)

- **p=1-q – вероят. безотказной работы компьютера на каждом выч. шаге**
- **q=1-p – вероятность отказа компьютера**
- **N – количество вычислительных шагов**
- **r – число компьютеров зарезервированного вычислительного процесса**
- **1-ε - требуемая вероят. корректного завершения процесса из N выч. шагов**

Вероятность успешного завершения процесса: $P_r(N) = (1-q^r)^N = 1-\varepsilon$

Оценка избыточности r для требуемой надежности процесса 1-ε:

$$r \approx (lg(\varepsilon)/lg(\delta)) - (lg(\delta))^{-1} lg(N)$$

Пусть $q=10^{-i}$, $\varepsilon=10^{-j}$, $N=10^n$, тогда $\boldsymbol{r \approx (j+n)/i}$

В случае ненадежности компьютеров $q=10^{-4}$ вероятность ошибки величиной $\varepsilon=10^{-10}$ для процесса с числом шагов $N=10^{20}$ может быть обеспечена за счет восьмикратного резервирования $r \approx 8$

Источники (http://zvt.hotbox.ru/)

Программирование структурно сложных распределенных вычислений в математически однородном поле компьютерной информации

- Затуливетер Ю. На пути к глобальному программированию // Открытые системы. 2003. № 3. http://www.osp.ru/os/2003/03/182704/
- Затуливетер Ю.С., Халатян Т.Г. ПАРСЕК - язык компьютерного исчисления древовидных структур с открытой интерпретацией. Стендовый вариант системы программирования. М.: Институт проблем управления РАН, 1997. 71с.
- Затуливетер Ю.С., Топорищев А.В. Язык Парсек: программирование глобально распределенных вычислений в модели исчисления древовидных структур // Проблемы управления. 2005. №4. С. 12-20.
- Затуливетер Ю.С. Проблемы глобализации парадигмы управления в математически однородном поле компьютерной информации // Проблемы управления. 2005. №1. Ч.I. С.1-12. №2. Ч.II. С.13-23. http://zvt.hotbox.ru.

Источники

Программные методы повышения надежности распределенных вычислений в недетерминированной компьютерной среде

- Затуливетер Ю.С., Карибский В.В., Лубков Н.В. Проблема организации надежных процессов управления с применением ненадежных вычислительных сред // Тезисы докладов. Четвертая международная конференция "Проблемы управления в чрезвычайных ситуациях", Москва, 11 января, 1997, С. 160-163. (http://zvt.hotbox.ru/bezop_96.htm)
- Ю.С. Затуливетер, И.А. Ходаковский. Пороговый метод повышения надежности распределенных вычислений в ненадежной компьютерной среде. (I.Резервирование с заменой выбывающих) (II.Резервирование без замены выбывающих). Труды IV международной конференции "Параллельные вычисления и проблемы управления" (PACO`08), 27 - 29 октября, 2008 (в печати).