

**Визуальная поддержка цикла  
разработки параллельных  
программ**

Байдалин А.Ю.

ИММ УрО РАН

[bajur@imm.uran.ru](mailto:bajur@imm.uran.ru)

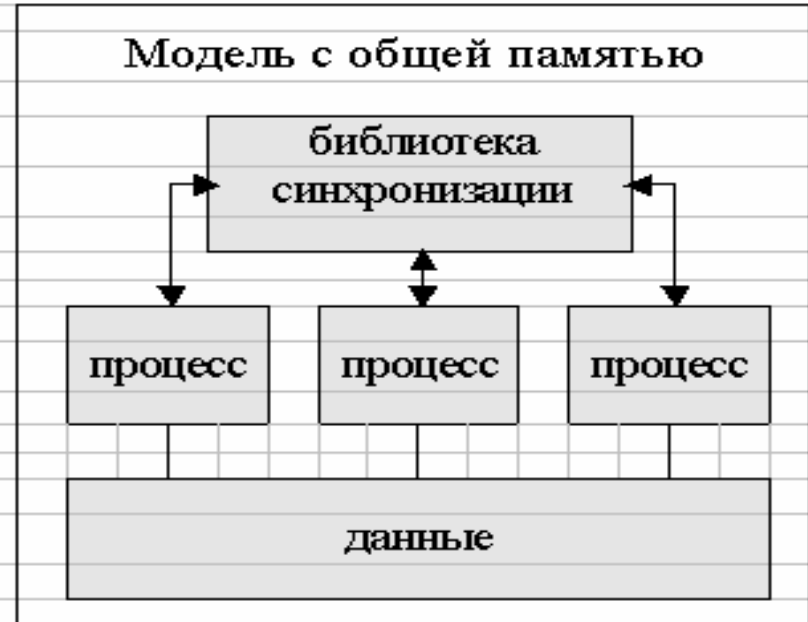
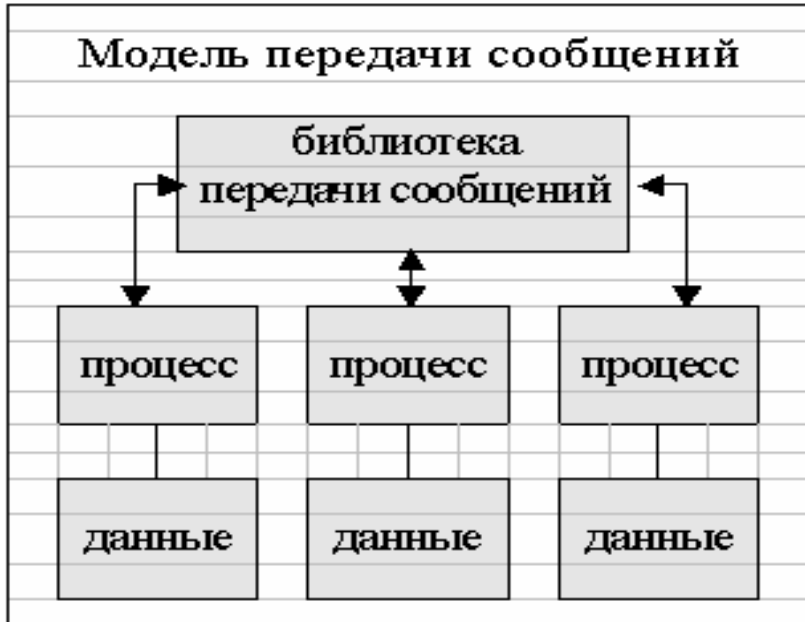
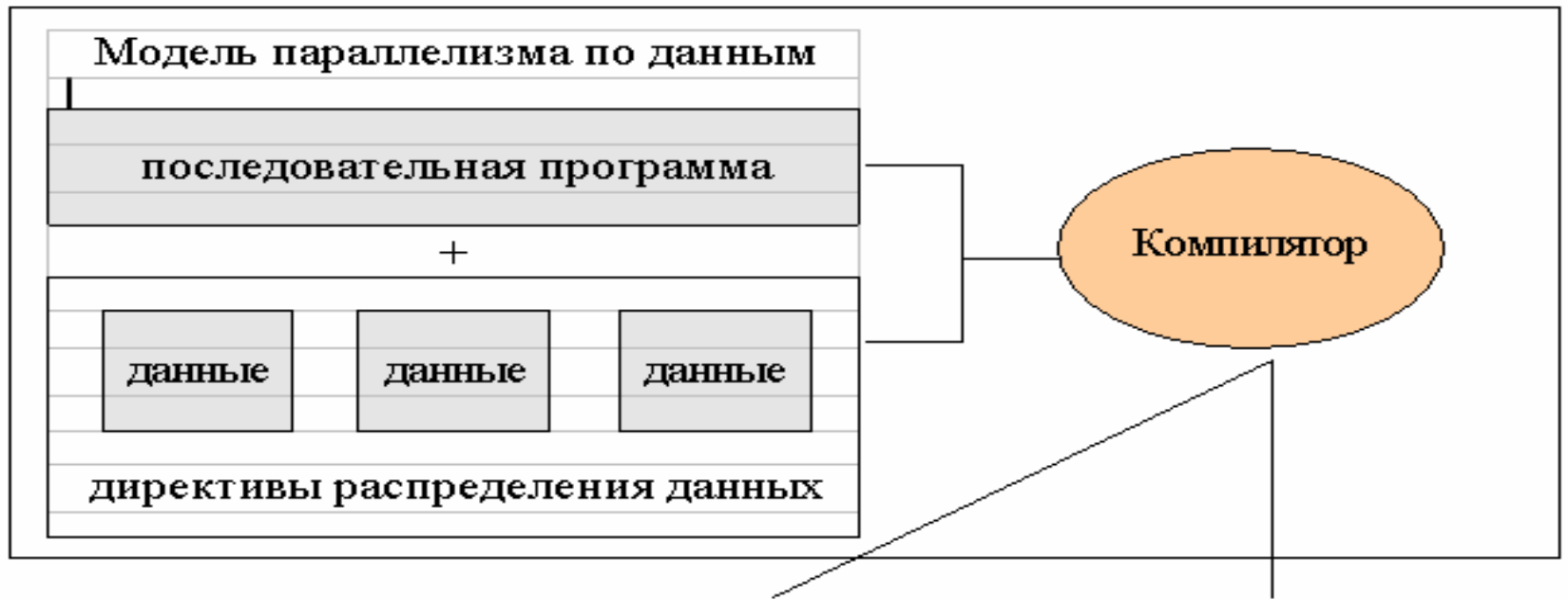
# Система DVM разработана в ИПМ РАН

- поставляется в виде исходных текстов или в виде набора выполняемых файлов
- использует стандартные Си и Фортран77
- опирается на существующие коммуникационные библиотеки ( MPI, Router, PVM, *etc* )
- может быть развернута на большинстве современных параллельных вычислителей ( MBS, SCI-кластер, *etc* )

# Система DVM

<http://www.keldysh.ru/dvm>

- реализует параллелизм по данным
- расширяет стандартные Фортран77 и Си в эффективные языки параллельного программирования
- обладает развитой системой сбора и анализа информации об эффективности работы программ, включая анализатор и предсказатель эффективности, а также средства сбора параллельной трассы



# Состав DVM-системы

- Компилятор Fortran DVM
- Компилятор C-DVM
- Библиотека поддержки Lib-DVM
- Средства DVM-отладки
- Анализатор эффективности
- Предсказатель производительности

# Языки программирования DVM

**C-DVM = Си + специальные  
макросы**

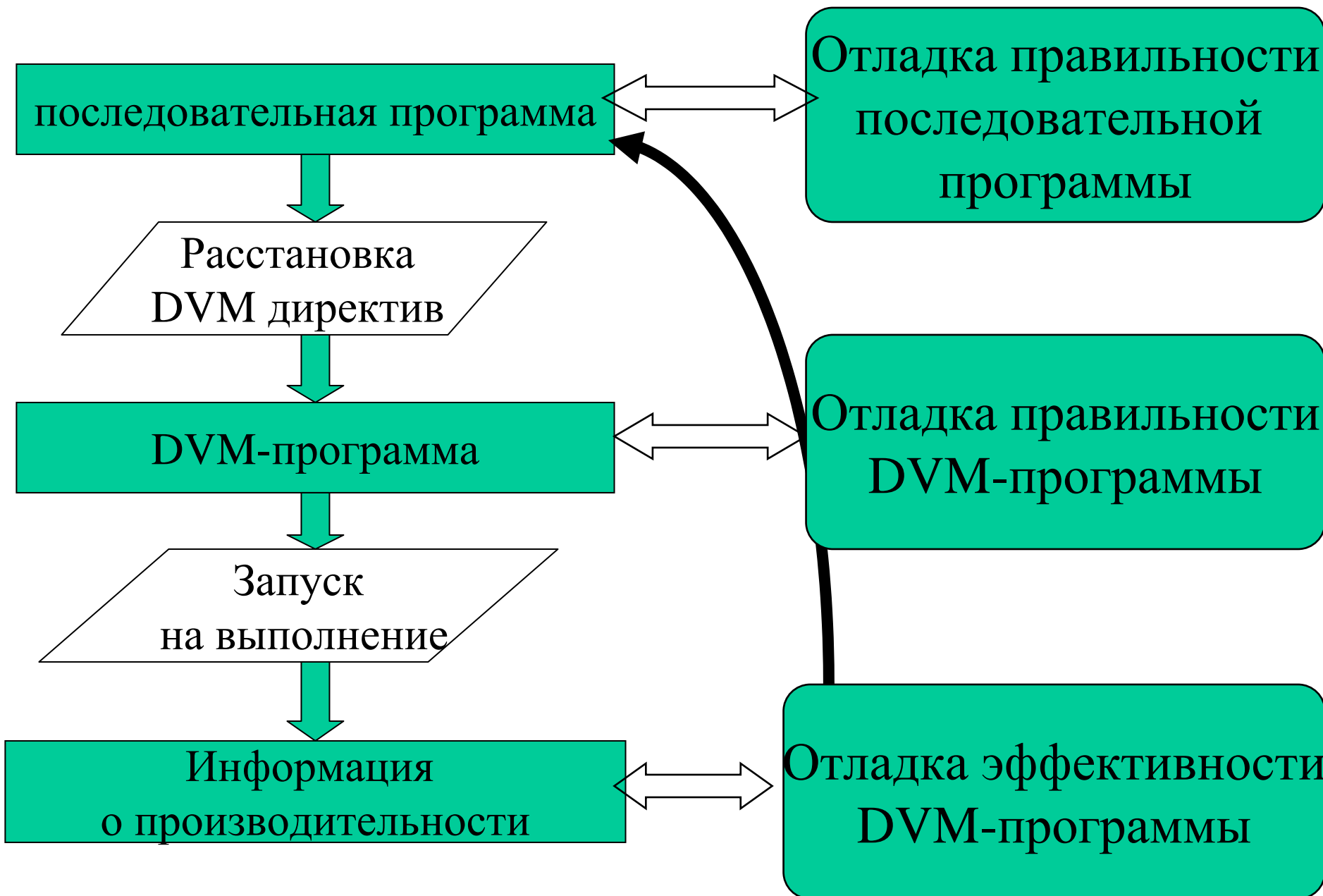
DVM(текст директивы) оператор;

**Fortran-DVM = Фортран + специальные  
комментарии**

CDVM\$ текст директивы  
оператор

Директивы распараллеливания задаются в виде специализированных макросов или комментариев, невидимых для обычных компиляторов.

# Этапы разработки DVM программы



# Фрагмент файла с информацией о производительности



jac\_4\_2.txt

Processor system=2\*2

Statistics has been accumulated on DVM-system version 382,  
platform WIN\_MPI

Analyzer is executing on DVM-system version 382, platform WIN\_MPI

-----  
INTERVAL ( NLINE=28 SOURCE=jacross.cdv ) LEVEL=0 EXE\_COUNT=1

--- The execution characteristics ---

	1	2	3	4
Lost time	4.8942	4.8375	4.8482	4.8417
User insufficient par.	0.0008	0.0023	0.0023	0.0015
Sys.insufficient par.	0.0615	0.2302	0.1980	0.2182
Idle time	0.1610	0.0260	0.0960	0.0000
Communication	4.6710	4.5790	4.5520	4.6220
Load imbalance	0.0708	0.0000	0.0110	0.0073
Execution time	4.7540	4.8890	4.8190	4.9150
User CPU time	0.0003	0.0008	0.0008	0.0005
Sys. CPU time	0.0030	0.0732	0.0622	0.0662
I/O time	0.0175	0.0035	0.0037	0.0065
Processors	4	4	4	4
Communication				
I/O	0.1250	0.0000	0.0000	0.0000
Reduction	3.0260	2.9430	2.8140	2.3220
Shadow	1.5200	1.6360	1.7380	2.3000



Сектор компьютерной визуализации ОСО ИММ УрО РАН на протяжении ряда лет занимается вопросами визуализации программного обеспечения и проводит как теоретические, так и практические исследования в данной области.

Разработка средств визуализации для системы DVM включила в себя поддержку как проектирования DVM-программ, так и анализа эффективности.

В настоящее время создан комплекс VProjDVM, который обеспечивает:

- визуальную поддержку разработки в виде мастеров директив, генерирующих директивы распараллеливания
- визуальный анализ сведений об эффективности DVM-программы, как одного, так и группы пусков (в направлении поиска наилучшей конфигурации процессоров)

Для эффективного проектирования, отладки правильности и отладки эффективности параллельных программ необходима визуальная поддержка всех этапов разработки.

Распараллеливание DVM-директивами требует очень хорошего представления распараллеливаемого алгоритма и обрабатываемых данных

Отладка параллельной правильности – та же работа с директивами распараллеливания. Кодирование и отладка правильности должны осуществляться на одних средствах представления.

Успешный анализ эффективности требует сбора больших объемов данных по одному, а чаще несколькими пусками программы на разных конфигурациях. Причем собираемые данные в случае DVM являются не одной циферкой, а комплексными наборами показателей

## Мастера директив – помощь в распараллеливании

Мастера применяются для генерирования DVM-команд, для создания DVM-директив с большим набором параметров.

Мастер --- специализированная экранная форма, наподобие анкеты или формы тестирования. В зависимости от значений главных параметров появляются или исчезают средства задания значений второстепенных параметров.

Пользователь взаимодействует с мастером путем выбора элемента из списка или указания конкретного имени и получает текст программы, сгенерированный на основе указанной им информации.

Обратно, пользователь может получить по исходному тексту директивы подробное описание ее назначения и действия.

### Мастер команд DVM

Файл запуска DVM:

Имя файла без расширения:

Размеры матрицы процессоров:

<input checked="" type="radio"/> 1	<input type="text" value="1"/>
<input type="radio"/> 2	<input type="text" value=""/>
<input type="radio"/> 3	<input type="text" value=""/>

**Язык**

C

Fortran77

Отладочные опции компиляции:

опций нет

динамический отладчик

анализатор эффективности

Динамический отладчик:

трассирования нет

модификация распред. массивов

модиф. и чтение распред. массивов

модификация. всех данных

модиф. и чтение всех данных

Анализатор производительности:

0. сбора информации нет

1. паралл. и объемл. послед. циклы

2. опред. пользователем INTERVAL

3. режим 1 + режим 2

4. режим 3 + послед. циклы

Цель:

- Выполняемый файл
- DVM-Конвертация в .c, .f
- Компиляция и связывание
- Послед. отладочный вариант
- Паралл. отладочный вариант
- Контроль DVM-директив
- Сбор эталонной трассировки
- 1-Сравнение с эталоном трассы
- N-Сравнение с эталоном трассы
- Множественная трассировка
- Комплексная проверка
- Оценка размеров трассировки
- Запуск на выполнение
- Анализ эффективности
- Подготовка прогноза
- Прогноз эффективности

Создать Копировать Очистить

DVM Команда [Опции] ИмяФайла

### Распределение массивов

Язык:  DVM-C  DVM-Fortran

Массивов:  1 (один)  N (много)

Имя/Список имен:

Тип распределения:

Distribution Первичное

Redistribution Вторичное

Размерность:

Распределение размерностей:

\*  
список распределений:  
распределение каждой  
размерности

Тип распределения:

\* (Локально)

BLOCK (Блок)

GEN\_BLOCK (Блок с весом)

Массив весов:

На массив процессоров

Окно результата – текст DVM директивы

Создать Очистить  
Копировать Отмена

ИЛИ

ИЛИ

# Анализатор производительности

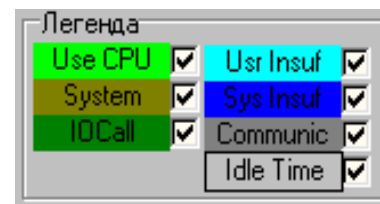
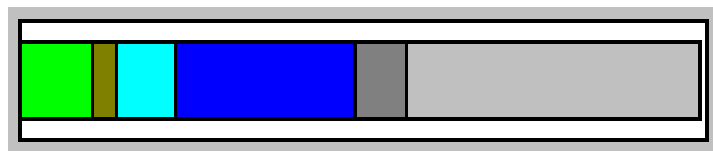
## Основные характеристики интервала

- Total Time -- общее время выполнения
- Execution time -- время выполнения на процессоре
- Productive time -- время выполнения на одном процессоре:
  - CPU -- время полезного счета программы
  - Sys -- время, потраченное на сист. вызовы
  - I/O -- время на операции ввода/вывода
- Lost time -- Временные потери:
  - Insufficient parallelism -- недостаточный параллелизм
    - User Insuff. -- пользовательский
    - System Insuff. -- системный
  - Communications -- потери на коммуникации
  - Idle Time -- простой процессоров в ожидании

Основной элемент сбора информации о производительности – *интервал* – фрагмент кода, выделенный системой или пользователем, и выполняющийся на одном или нескольких процессорах. Для него замеряется как полное время работы, так и его составляющие.

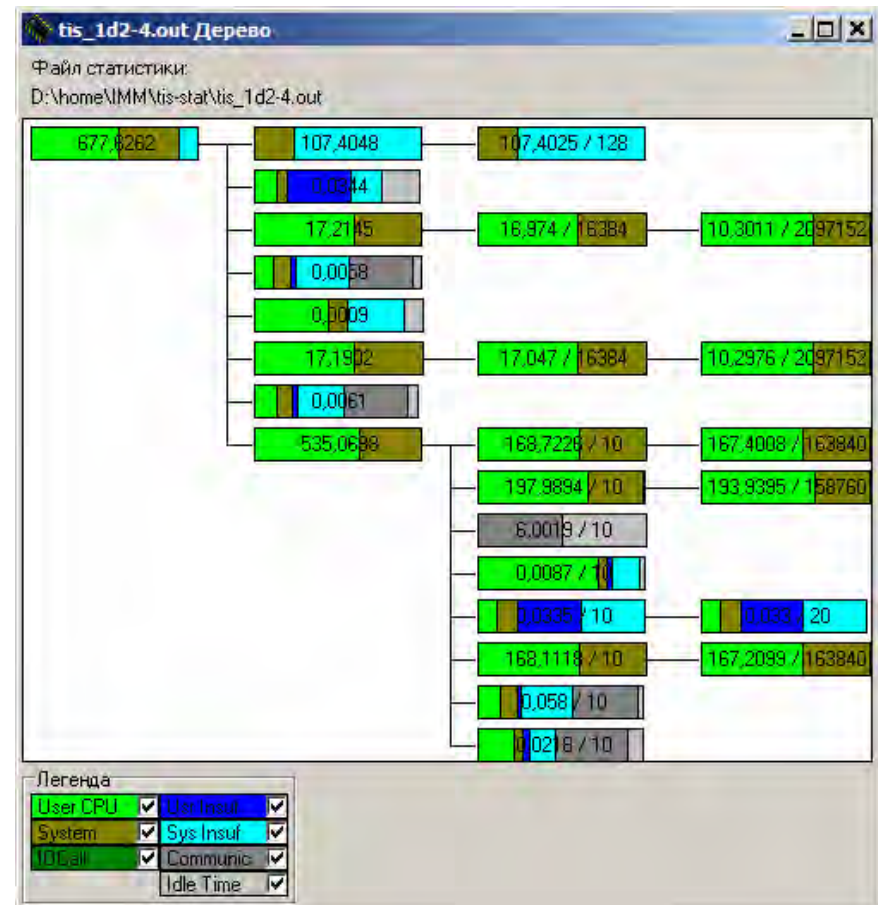
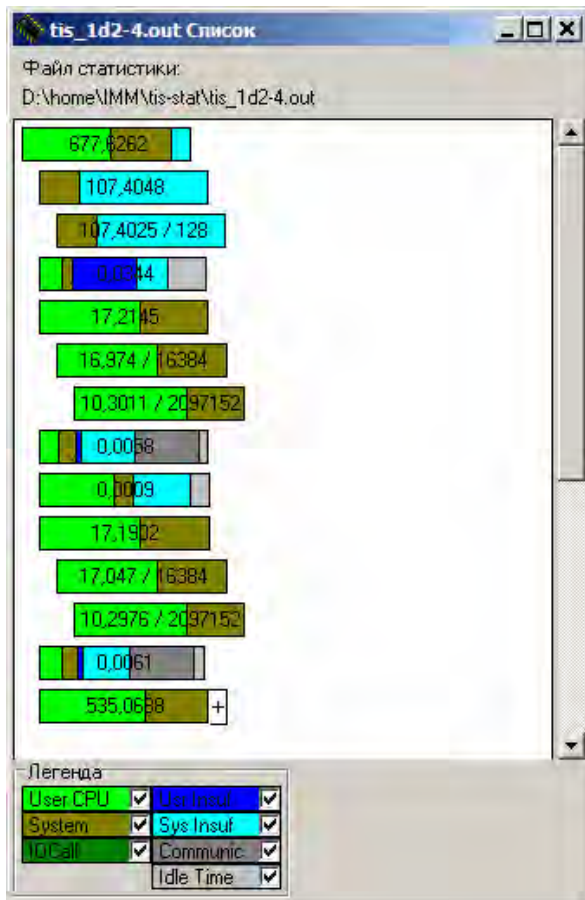
Интервалы вложены друг в друга и образуют дерево – дерево интервалов программы, где корнем является «вся программа»

Соответственно и визуальные средства проектировались чтобы обеспечить правильную интерпретацию визуального образа.



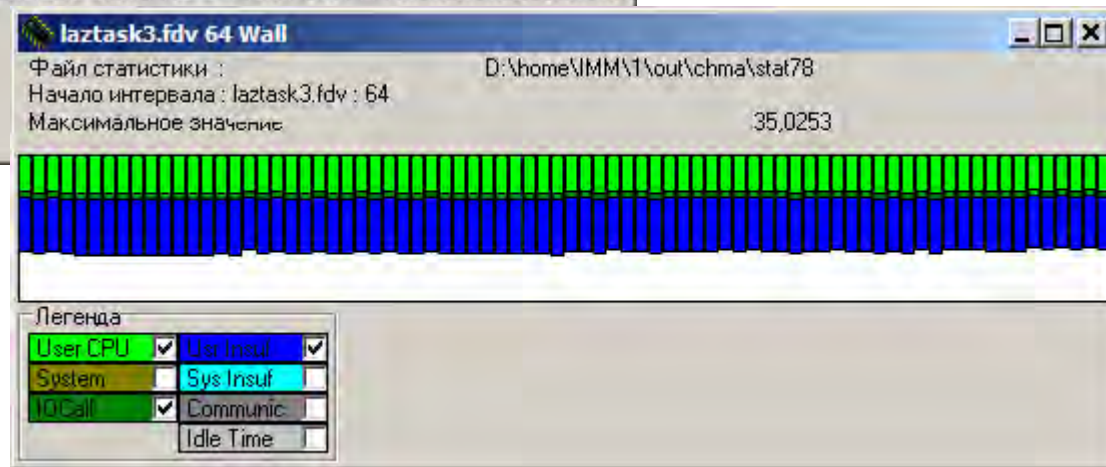
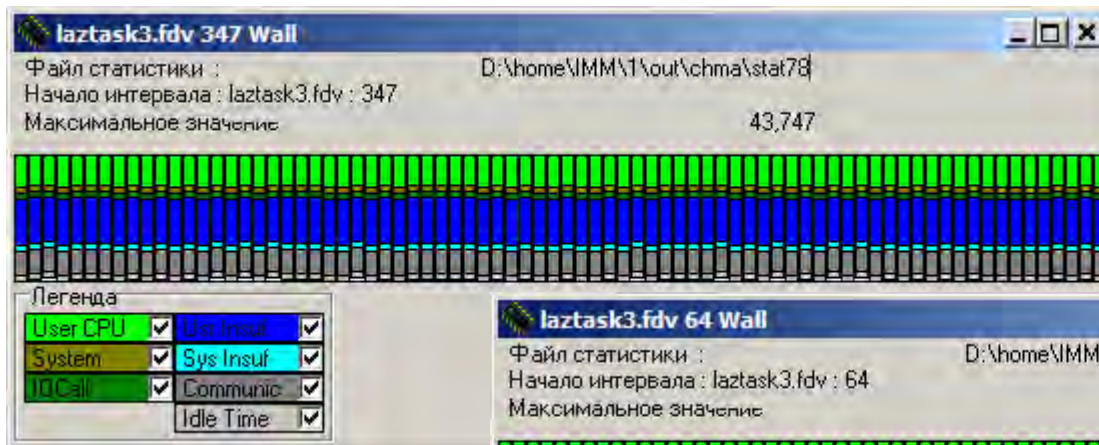
Основным элементом была взята диаграмма интервала, отражающая как общее время выполнения интервала, так и его слагаемые

- На диаграмме интервала базируются основные комплексные виды отображения информации о производительности программы
- дерево интервалов (справа)
  - иерархический список интервалов (слева)

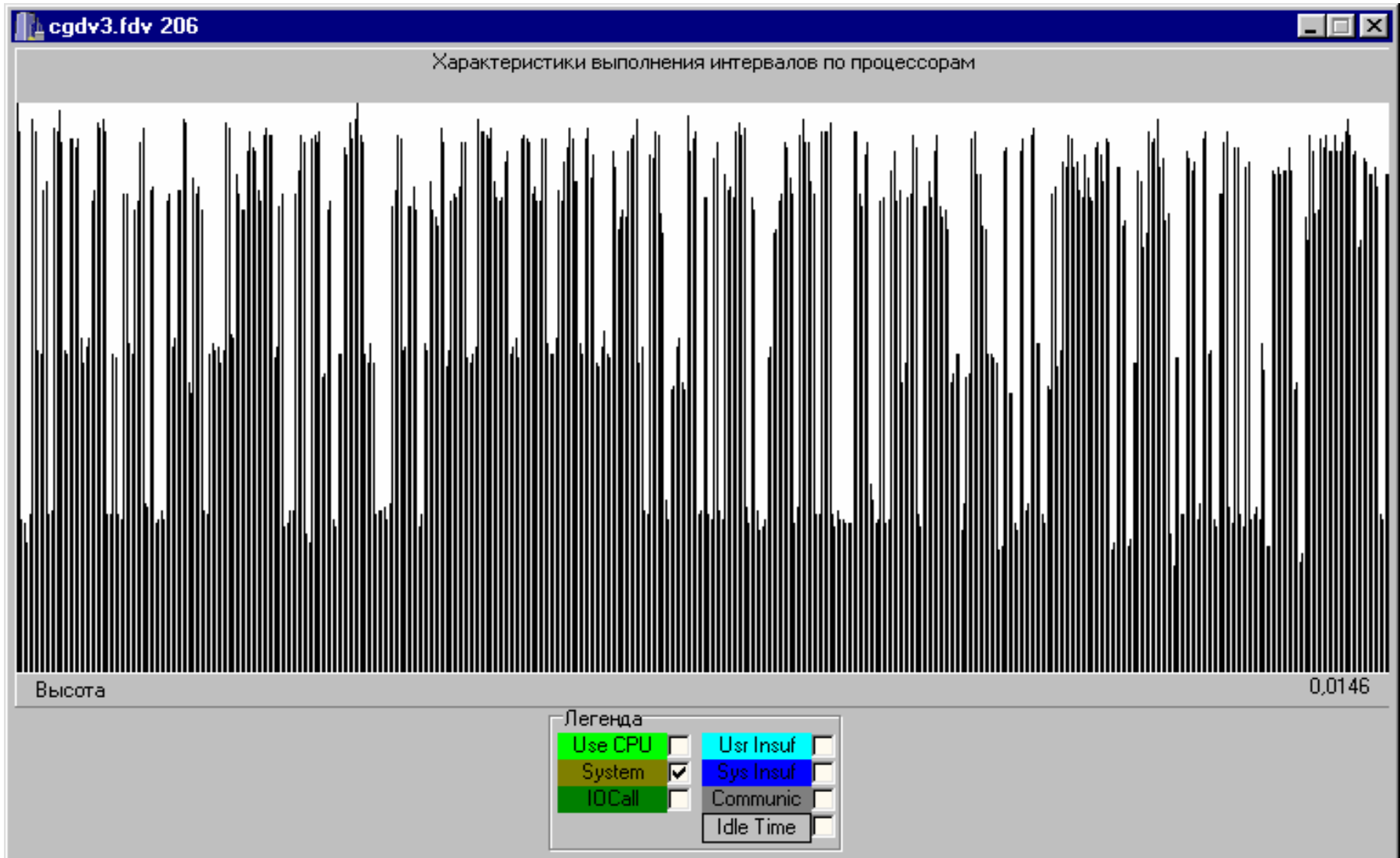


Для анализа баланса загруженности процессоров при выполнении параллельных циклов необходимо анализировать не общее дерево интервалов, а наборы показателей производительности, собранные для одного интервала, выполнявшегося на разных процессорах.

Общую картину выполнения интервала на группе процессоров отражает т.н. *стена интервалов*

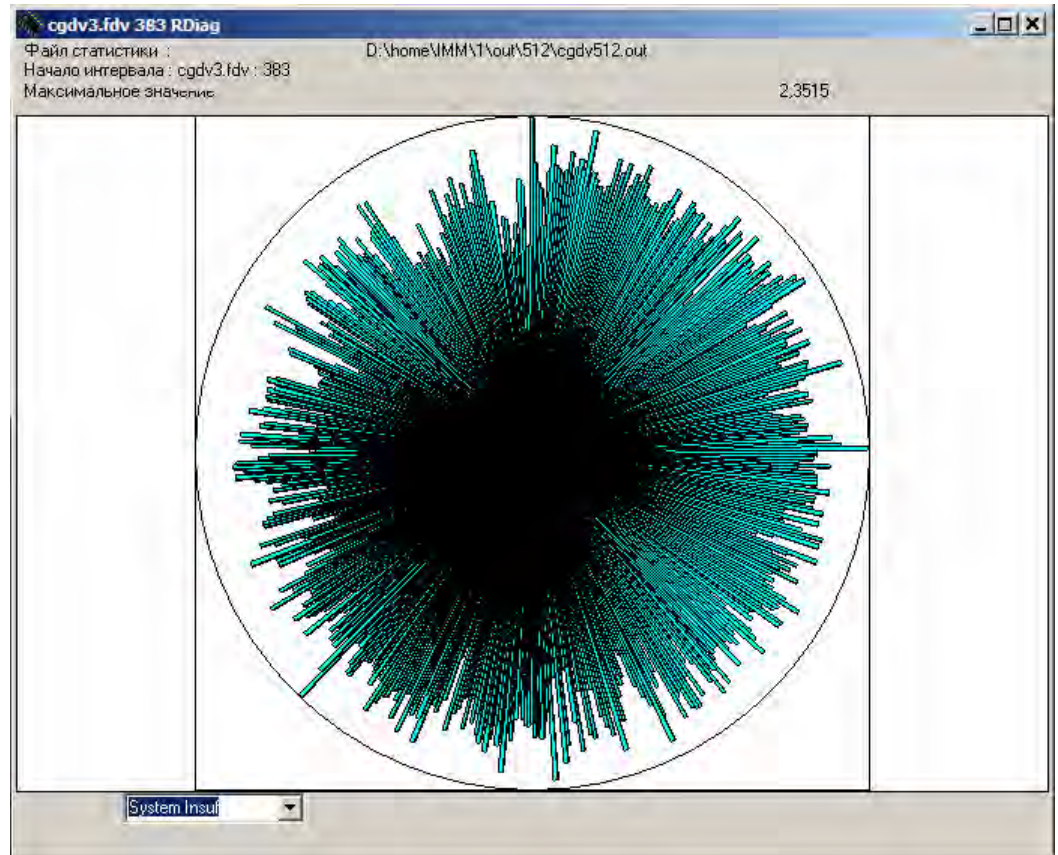
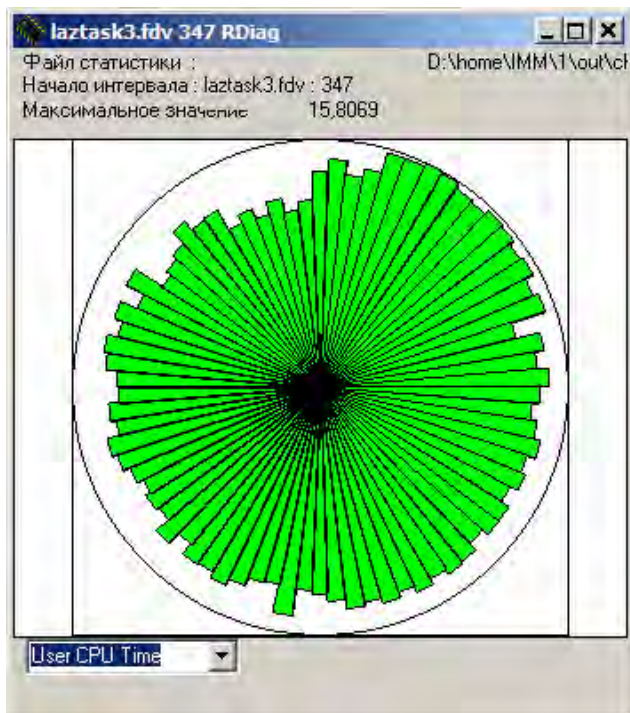






Стена интервалов для 512 процессоров

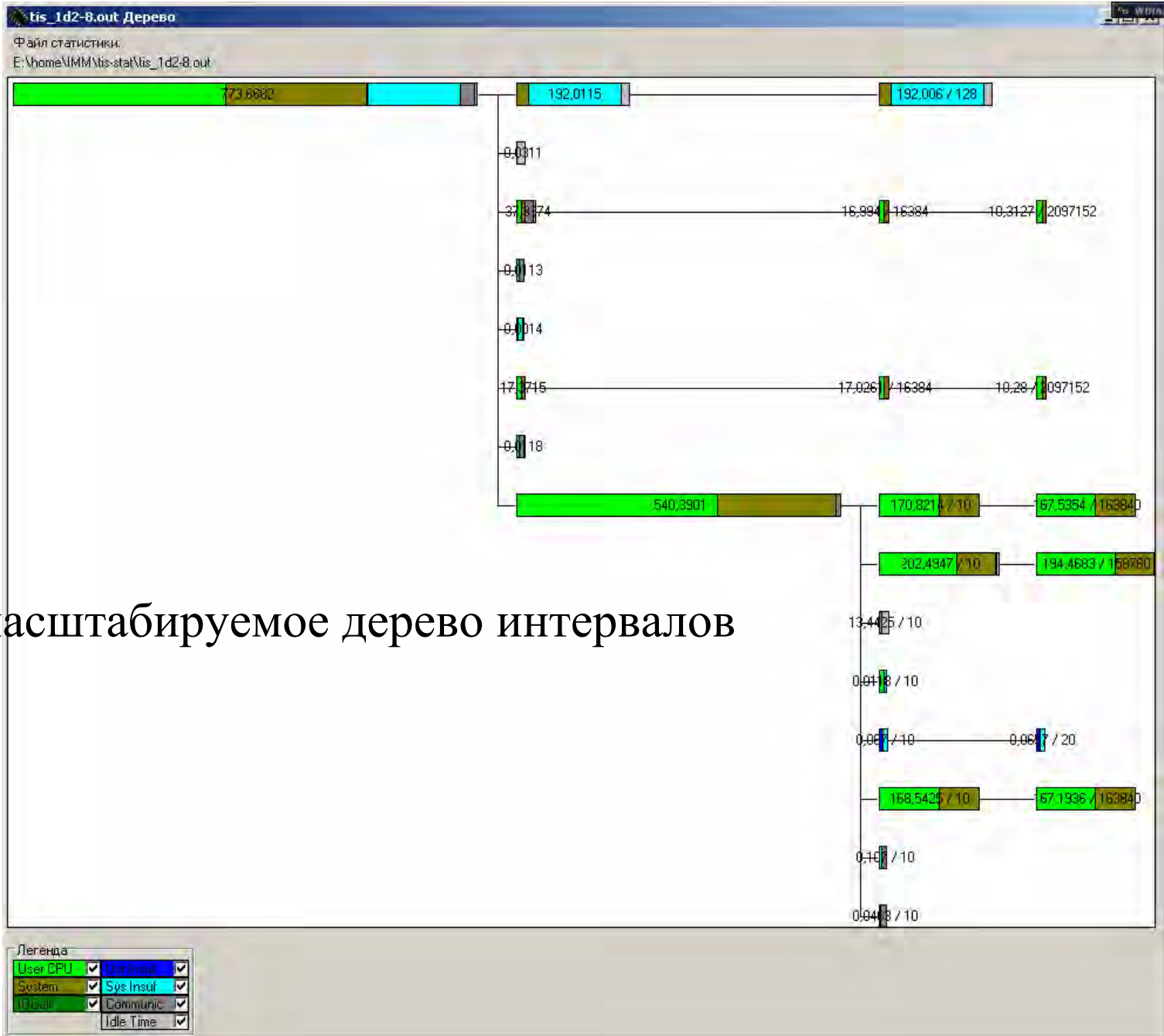
Для анализа баланса загрузки в рамках одного слагаемого  
полного времени выполнения была разработана  
*радиальная диаграмма*



Для повышения информативности визуальных образов в настоящее время разрабатываются новые комплексные средства представления, передающие общий масштаб временных величин, а не только доли слагаемых в общем времени выполнения.

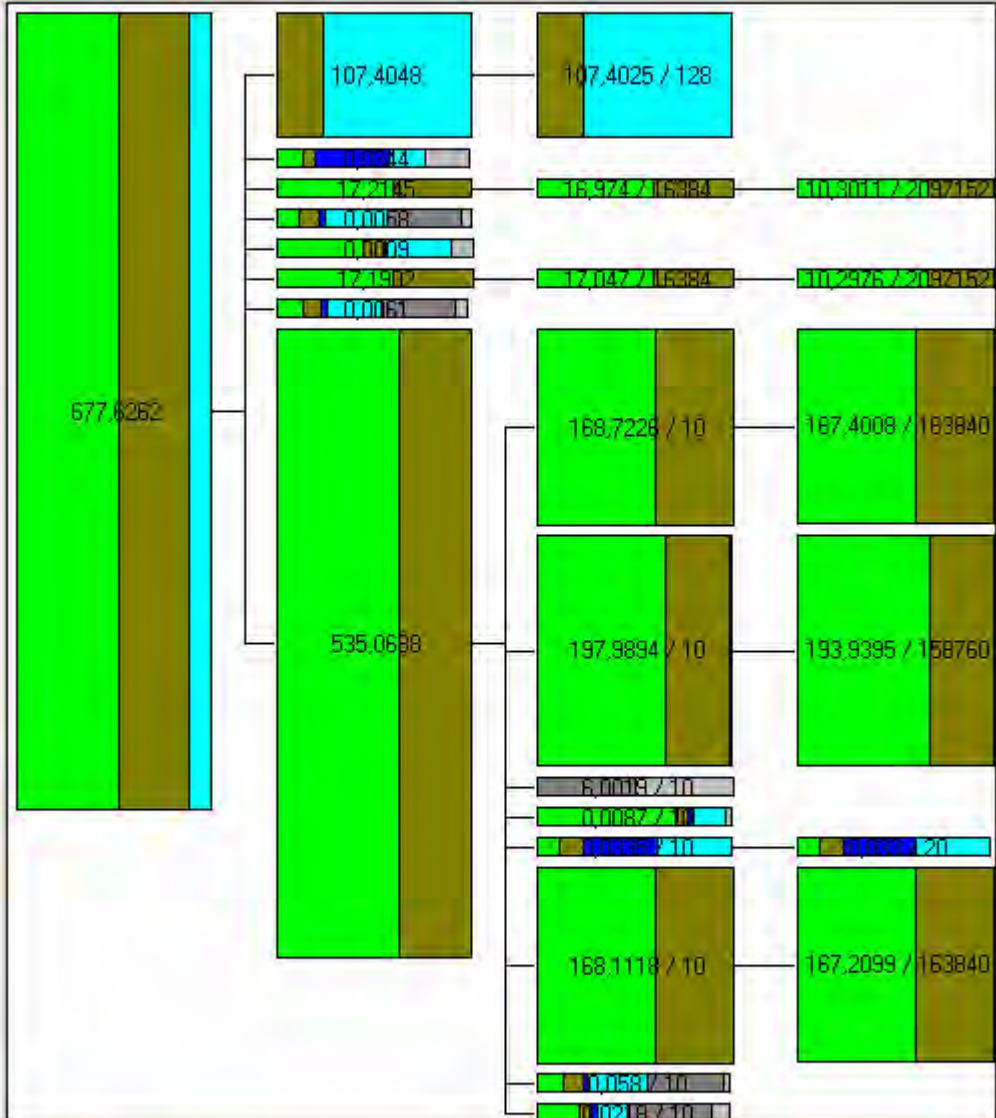
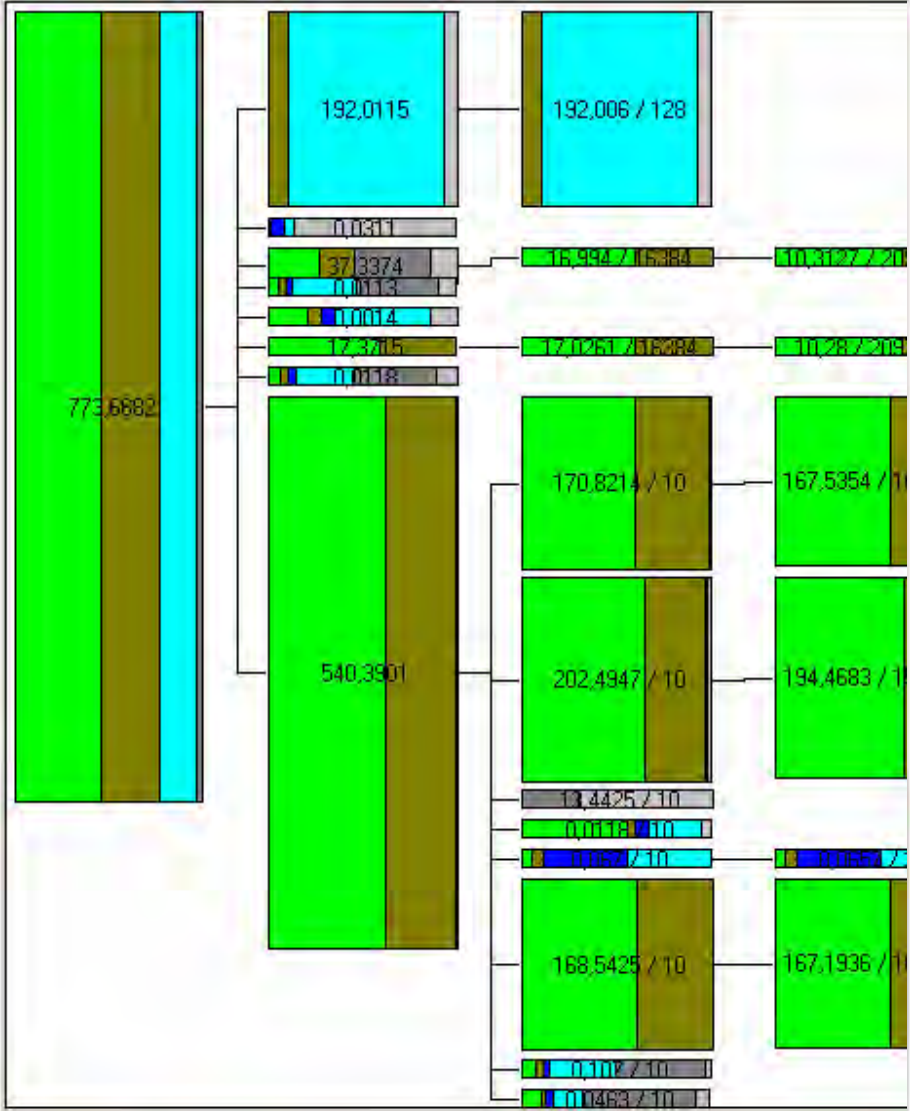
Так, дальнейшим развитием обычных списков и деревьев интервалов являются масштабируемые.

Также предложен вариант объединения дерева интервалов и стен интервалов – т.н. *дерево стен*



## X-масштабируемое дерево интервалов

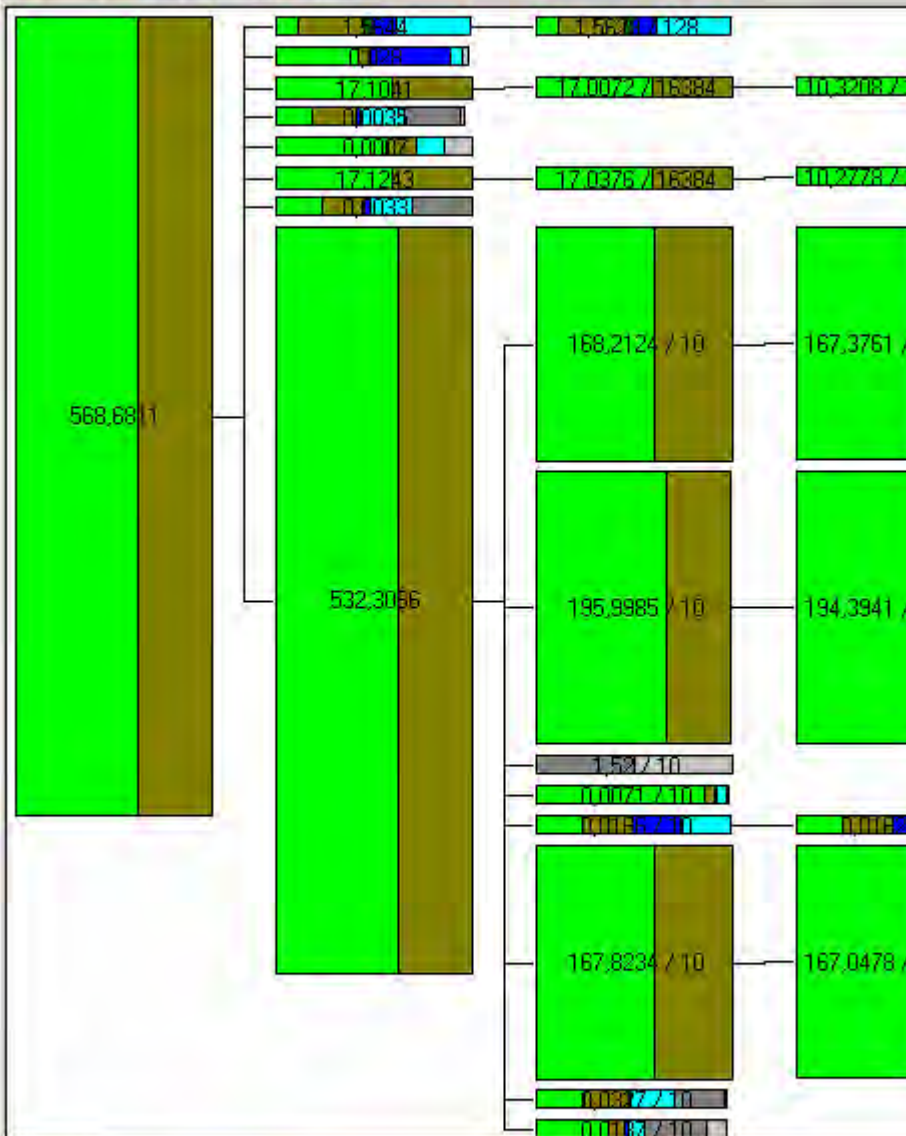
# Y-масштабируемые деревья интервалов



## Y-масштабируемые деревья интервалов

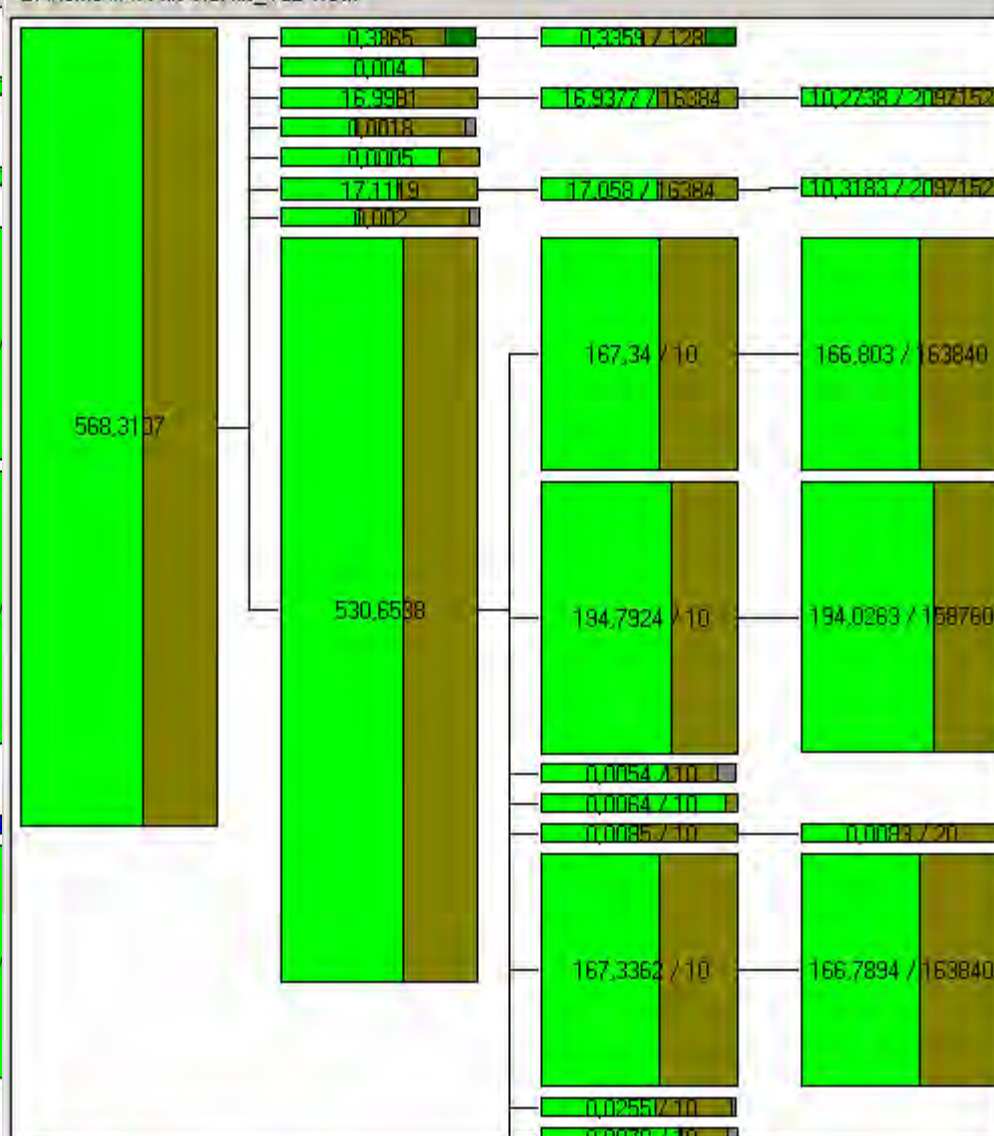
Файл статистики:

E:\home\MM\tis-stat\tis\_1d2-2.out



Легенда

User CPU	<input checked="" type="checkbox"/>	User Insuf	<input checked="" type="checkbox"/>
System	<input checked="" type="checkbox"/>	Sys Insuf	<input checked="" type="checkbox"/>
IOCall	<input checked="" type="checkbox"/>	Communic	<input checked="" type="checkbox"/>
Idle Time	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>



Легенда

User CPU	<input checked="" type="checkbox"/>	User Insuf	<input checked="" type="checkbox"/>
System	<input checked="" type="checkbox"/>	Sys Insuf	<input checked="" type="checkbox"/>
IOCall	<input checked="" type="checkbox"/>	Communic	<input checked="" type="checkbox"/>
Idle Time	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>

# Дерево стен и Список стен

