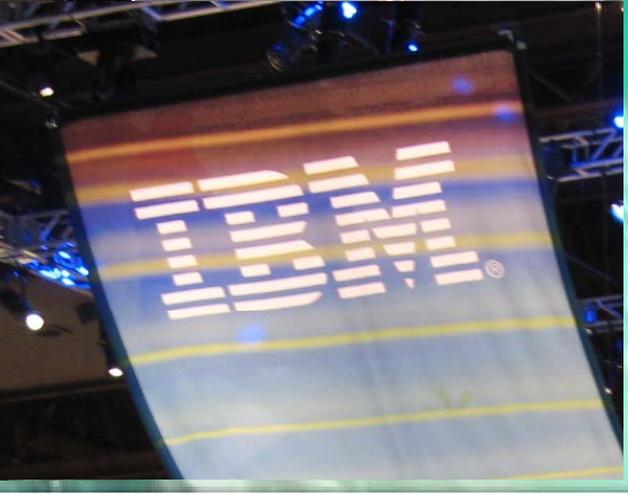
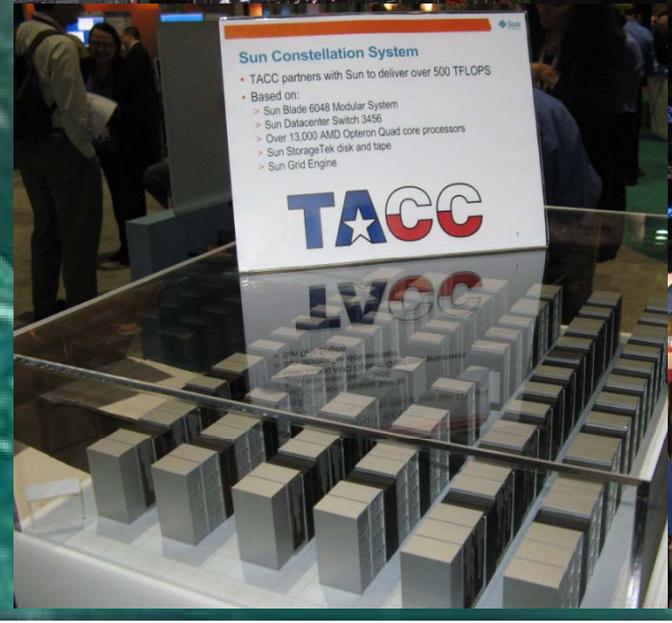
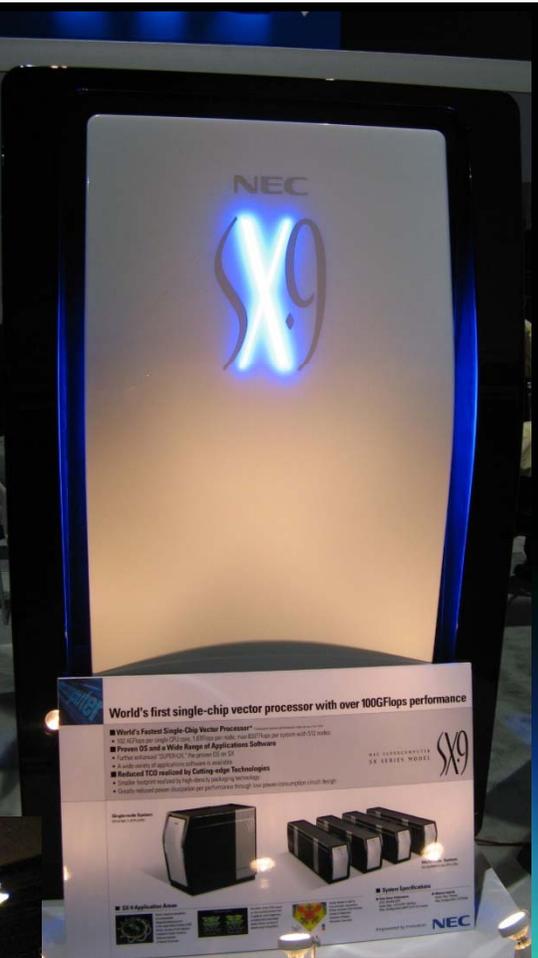


*Всероссийская молодежная школа
“Суперкомпьютерные технологии и высокопроизводительные
вычисления в образовании, науке и промышленности”*

Суперкомпьютеры и КПД паровоза

*Вл.В.Воеводин
Заместитель директора НИВЦ МГУ,
член-корреспондент РАН,
voevodin@parallel.ru*

ННГУ – 31 октября 2009 г.



Компьютер NEC “Earth Simulator”, 2002г. (#1 Top500 в 2003-2005 г.)

5120 процессоров, производительность: **35 Тфлоп/с.**

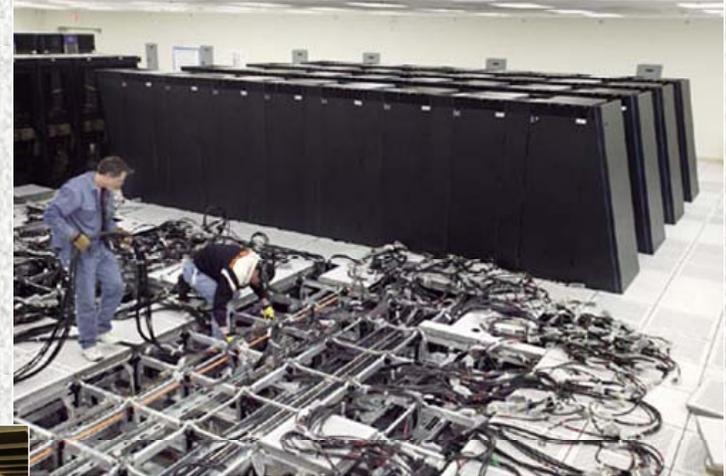
Память - 10 Тбайт, диски – 250 Тбайт, ленты – 1.5 Пбайт



The Earth Simulator Center

Компьютер IBM Blue Gene

(#1 Top500 в 2007 г.)



212 992 процессора
IBM PowerPC 440.

Производительность:
478 Тфлоп/с

Компьютер IBM "RoadRunner"

(#1 Top500 в 2008-2009 г.)



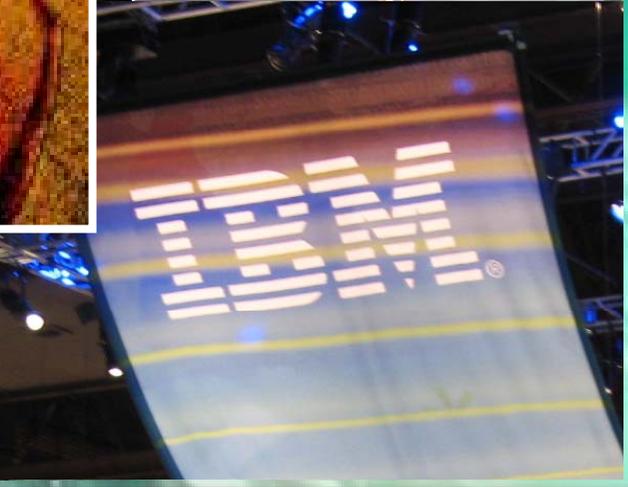
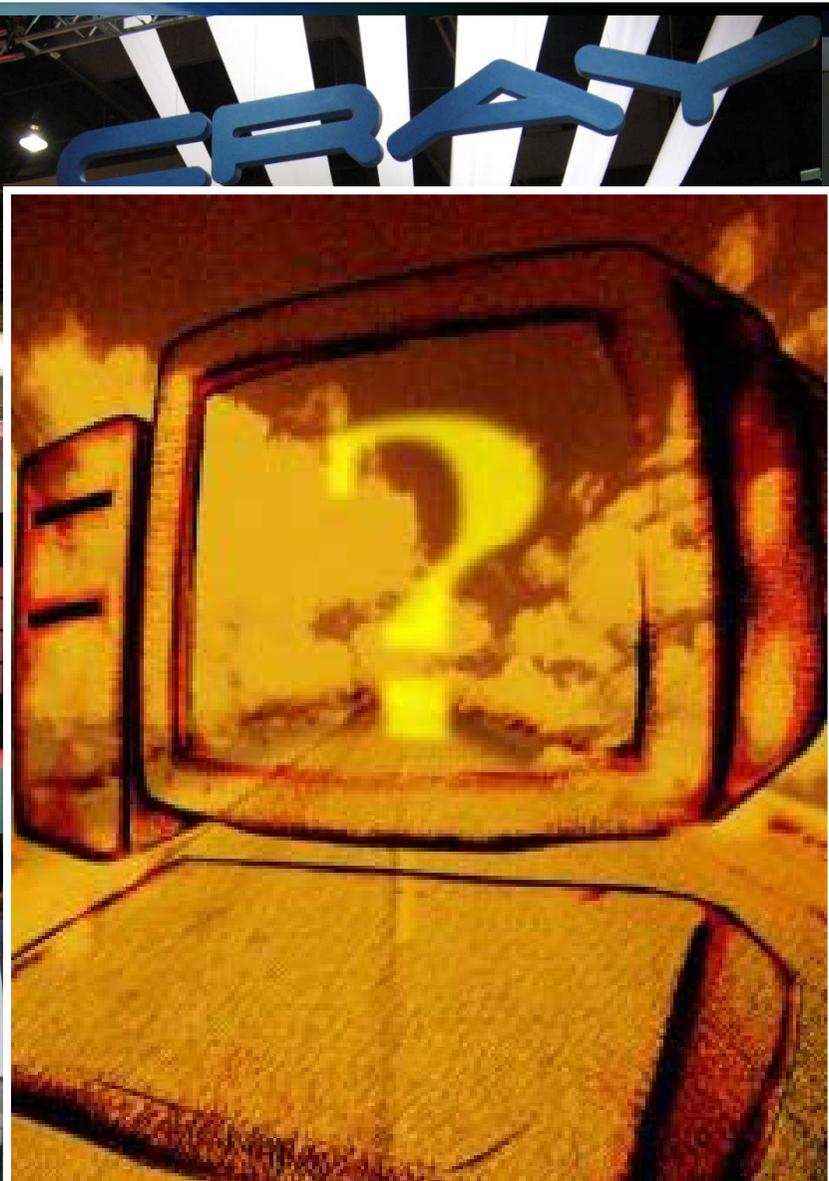
Самый мощный
компьютер мира:
122 400 процессоров
IBM Cell + AMD Opteron.

Производительность:
1 026 Тфлоп/с.

Характеристики суперкомпьютеров

- *IBM RoadRunner, 6562 AMD Opteron DC + 12240 IBM Cell, 1105 Tflop/s, ОП = 98 TB*
- *SGI Altix Ice 8200, 51200 CPUs, Intel Xeon 2.66 GHz QC, 487 Tflop/s, ОП = 51 TB, диски = 900 TB*
- *IBM Blue Gene, 212992 CPUs, PowerPC 440, 478 Tflop/s, ОП = 74 TB*
- *Cray XT4, 38642 CPUs, AMD Opteron 2.3 GHz QC, 266 Tflop/s, ОП = 77 TB, диски = 340 TB*





Эффективность работы компьютера. Что это?

Пиковая производительность компьютера, R_{peak}

Реальная производительность, R_{max}

$$R_{max} = \frac{\text{число операций в программе}}{\text{время работы компьютера}}$$

$$\text{Эффективность} = \frac{R_{max}}{R_{peak}}$$

Производительность компьютеров. Тест Linpack

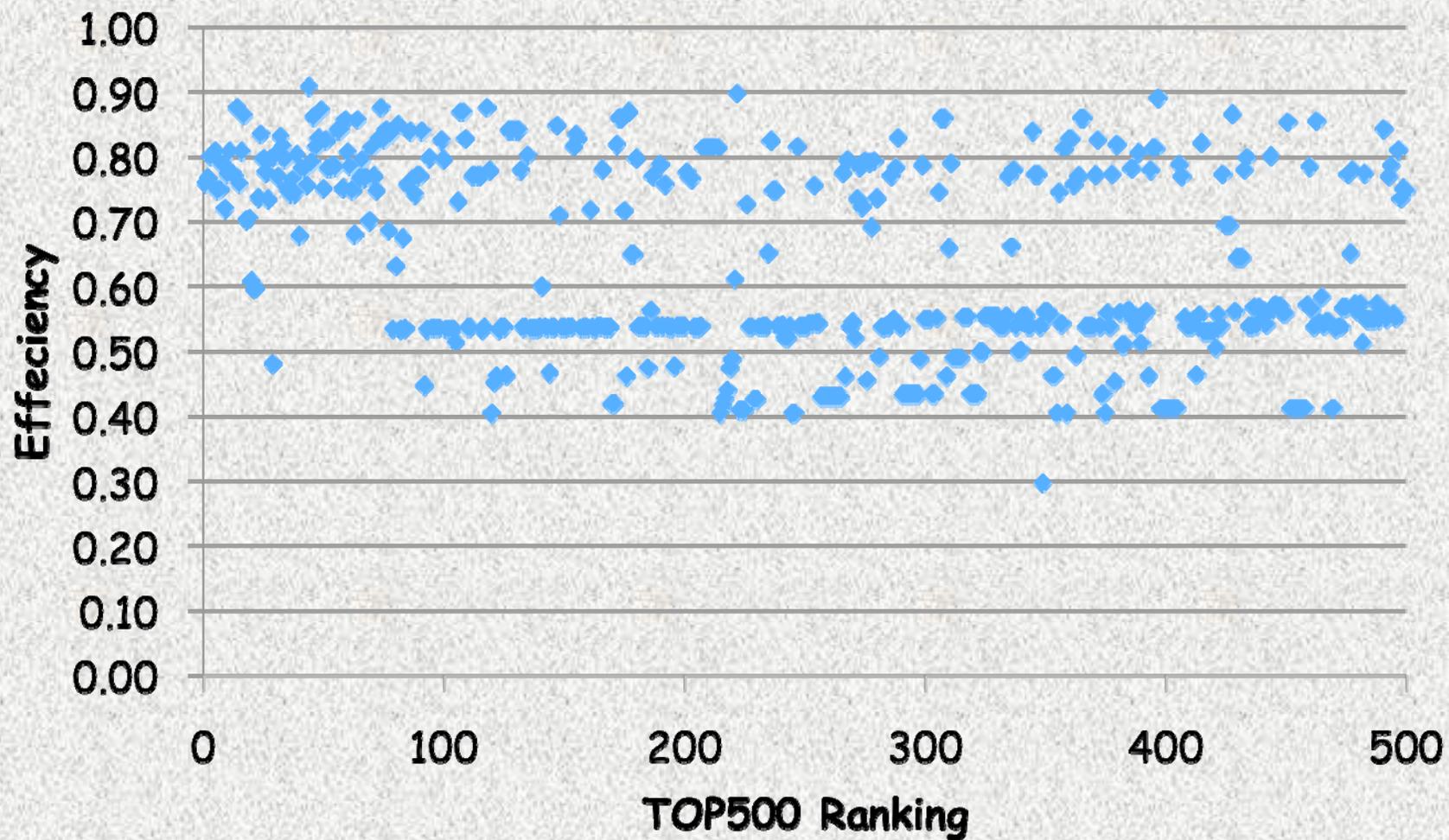
Тест Linpack – решение системы линейных алгебраических уравнений с плотной матрицей.

- 1. Матрица 100*100, фиксированный текст программы.*
- 2. Linpack TPR: матрица 1000*1000, можно менять метод и текст программы. Сложность : $2n^3/3+2n^2$.*
- 3. High Performance Linpack: матрица любого размера, множество дополнительных параметров.*

<http://www.top500.org>

Rank	Site	System/Vendor	Cores	R _{max}	R _{peak}
1	DOE/NNSA/LANL United States	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband IBM	129600	1105	1456.7
2	Oak Ridge National Laboratory United States	Cray XT5 QC 2.3 GHz Cray Inc.	150152	1059	1381.4
3	Forschungszentrum Juelich (FZJ) Germany	Blue Gene/P Solution IBM	294912	825.5	1002.7
4	NASA/Ames Research Center/NAS United States	SGI Altix ICE 8200EX, Xeon QC 3.0/2.66 GHz SGI	51200	487.01	608.83
5	DOE/NNSA/LLNL United States	eServer Blue Gene Solution IBM	212992	478.2	596.38
6	National Institute for Computational Sciences/University of Tennessee United States	Cray XT5 QC 2.3 GHz Cray Inc.	66000	463.3	607.2
7	Argonne National Laboratory United States	Blue Gene/P Solution IBM	163840	458.61	557.06
8	Texas Advanced Computing Center/Univ. of Texas United States	SunBlade x6420, Opteron QC 2.3 Ghz, Infiniband Sun Microsystems	62976	433.2	579.38
9	DOE/NNSA/LLNL United States	Blue Gene/P Solution IBM	147456	415.7	501.35
10	Forschungszentrum Juelich (FZJ) Germany	Sun Constellation, NovaScale R422-E2, Intel Xeon X5570, 2.93 GHz, Sun M9/Mellanox QDR Infiniband/Partec Parastation Bull SA	26304	274.8	308.28
11	NERSC/LBNL United States	Cray XT4 QuadCore 2.3 GHz Cray Inc.	38642	266.3	355.51
12	Oak Ridge National Laboratory United States	Cray XT4 QuadCore 2.1 GHz Cray Inc.	30976	205	260.2
13	NNSA/Sandia National Laboratories United States	Sandia/ Cray Red Storm, XT3/4, 2.4/2.2 GHz dual/quad core Cray Inc.	38208	204.2	284
14	King Abdullah University of Science and Technology Saudia Arabia	Blue Gene/P Solution IBM	65536	185.17	222.82
15	Shanghai Supercomputer Center China	Dawning 5000A, QC Opteron 1.9 Ghz, Infiniband, Windows HPC 2008 Dawning	30720	180.6	233.47

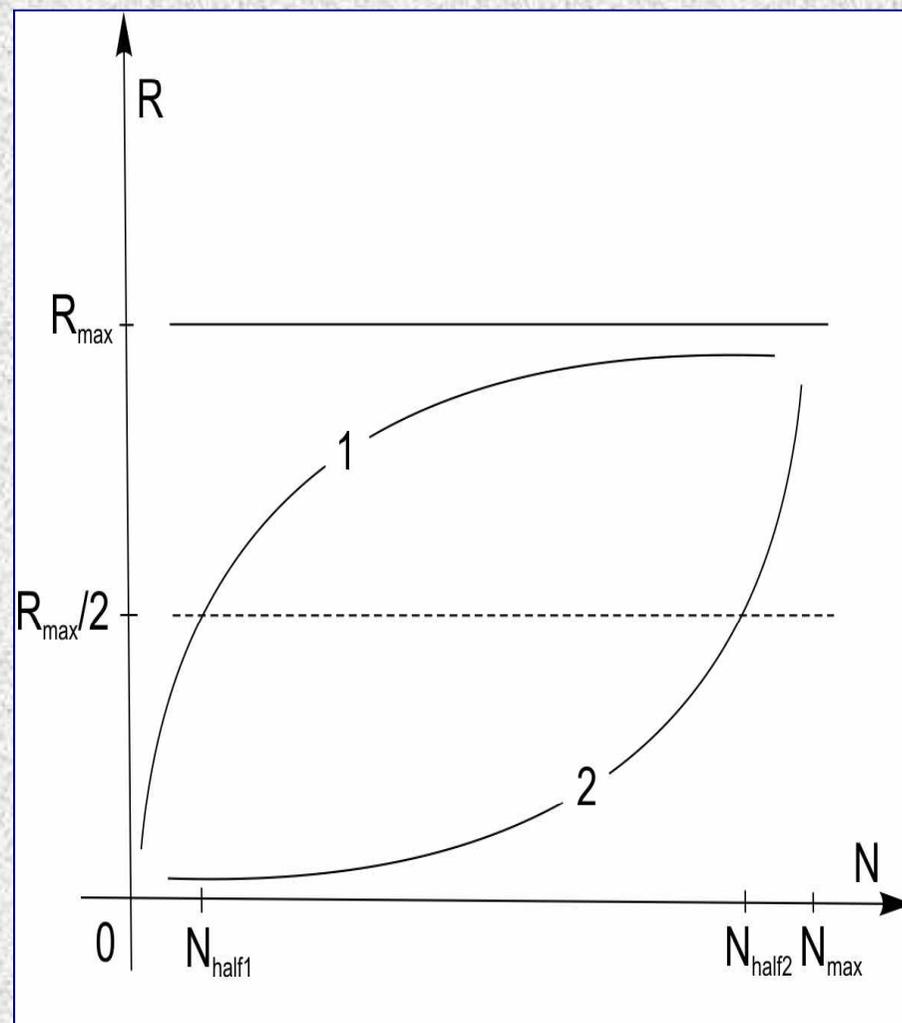
Top500 , Linpack, эффективность



Эффективность работы компьютеров

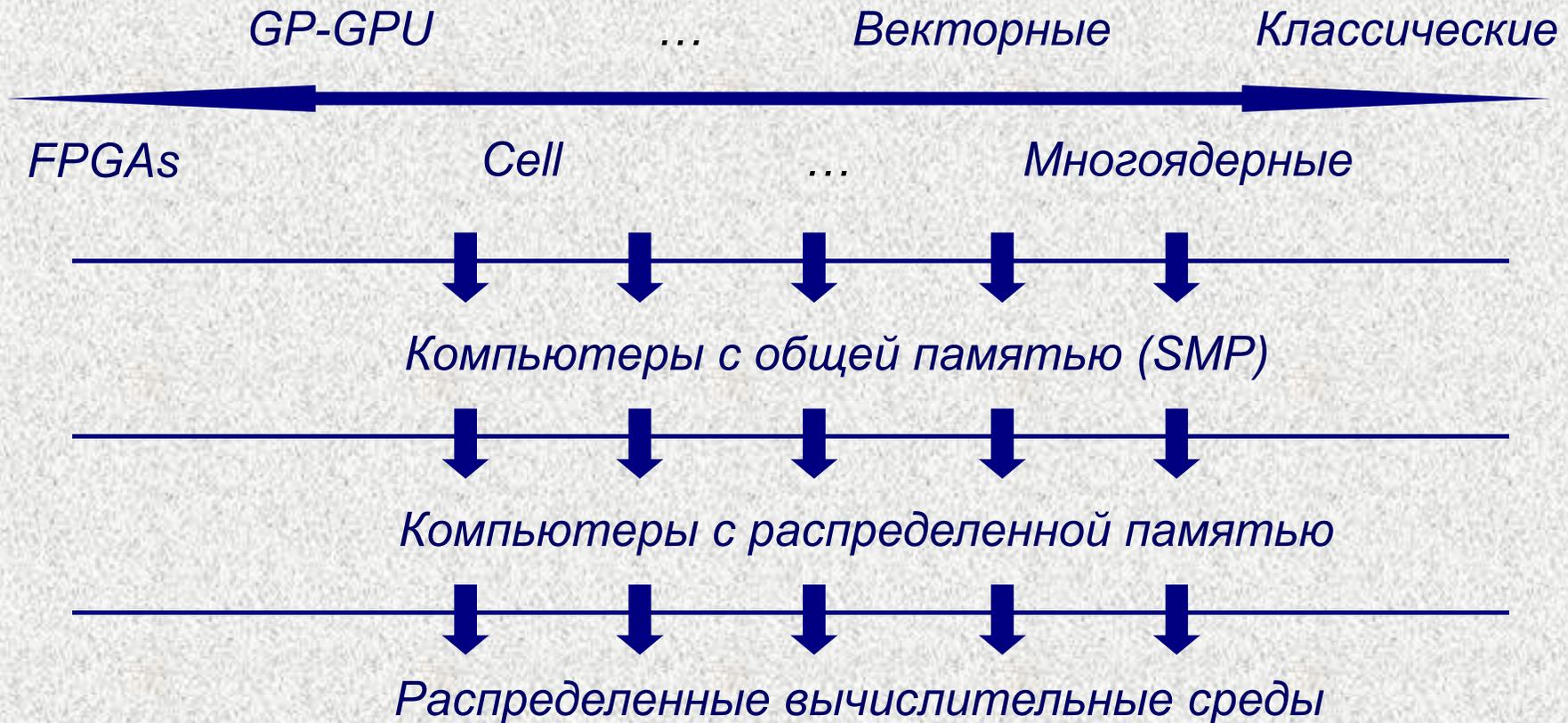
N_{max} — размер задачи, на которой достигается максимальная реальная производительность R_{max}

N_{half} — размер задачи, на которой достигается половина максимальной производительности $R_{max}/2$



*Что снижает эффективность работы
реальных компьютеров?*

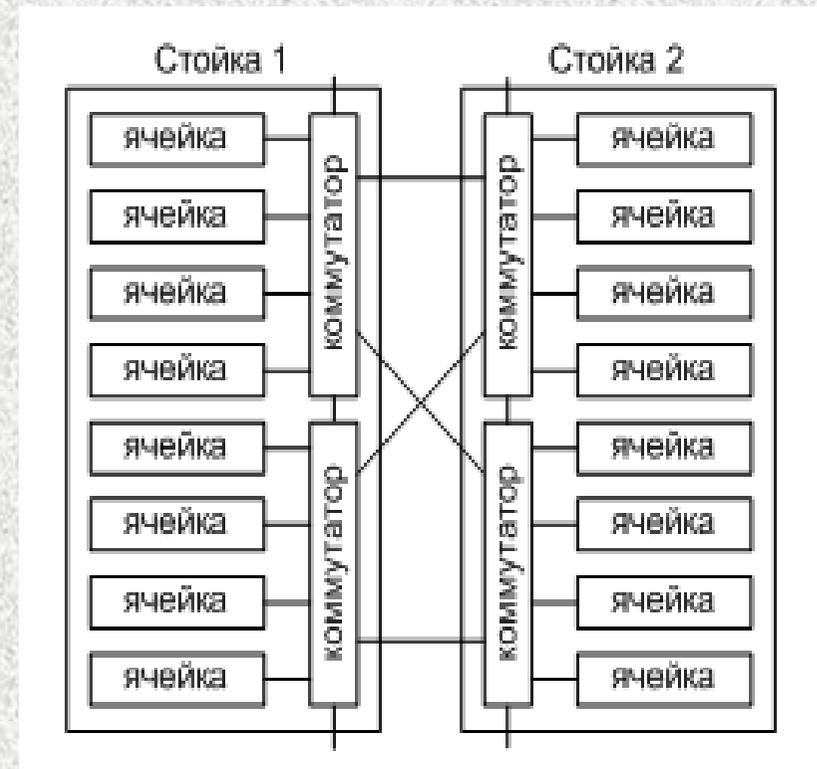
Высокопроизводительные компьютерные системы



Компьютеры с общей памятью (Hewlett-Packard Superdome)

Выпускается с 2000 г.

- Используемые процессоры:
- PA-8600, PA-8700, PA-8900
 - Intel IA/64: Itanium, Itanium 2



Причины снижения производительности компьютеров с общей памятью

1. Закон Амдала.
2. *сс*NUMA.
3. *сс*NUMA.
4. Сбалансированность вычислительной нагрузки процессоров.
5. Производительность отдельных процессоров.
6. ...

Закон Амдала

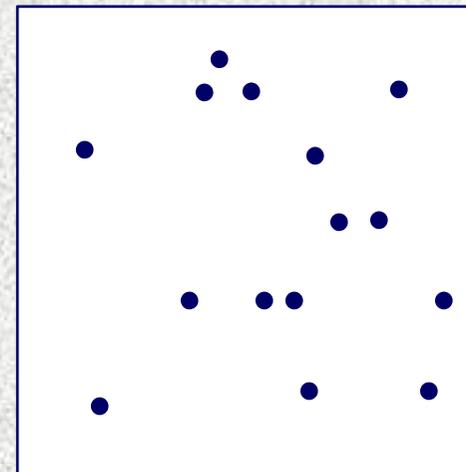
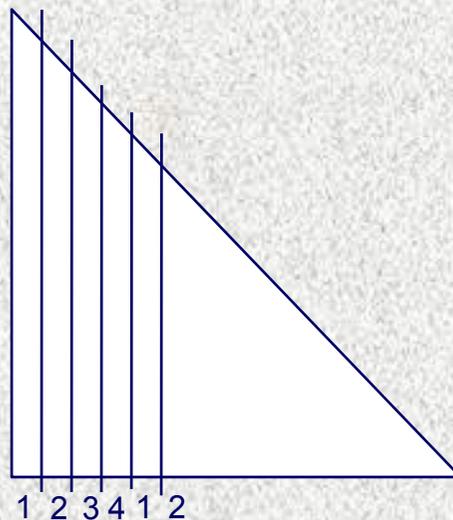
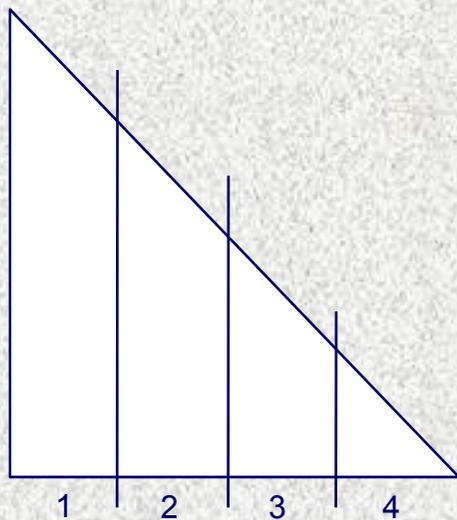
$$S \leq \frac{1}{\alpha + \frac{(1 - \alpha)}{p}} \qquad S \approx \frac{1}{\alpha}$$

где:

α – доля последовательных операций,
 p – число процессоров в системе.



Сбалансированность вычислительной нагрузки процессоров



Компьютеры с распределенной памятью (семейство Cray XT)

Семейство: T3D, T3E, T3X, XT3, XT4, XT5

Cray T3D – начало 90-х годов

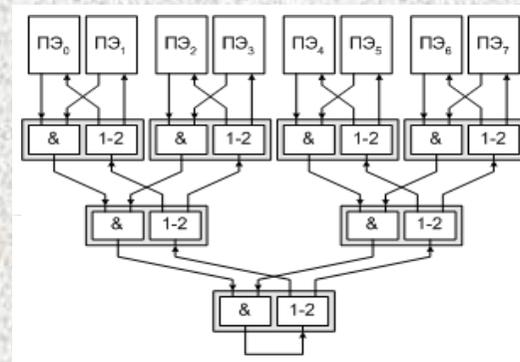
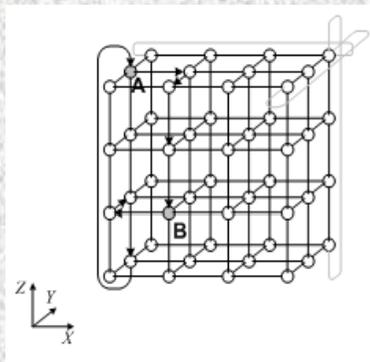
Cray XT5 – на октябрь 2009 г. второе место в Top500

Используемые процессоры:

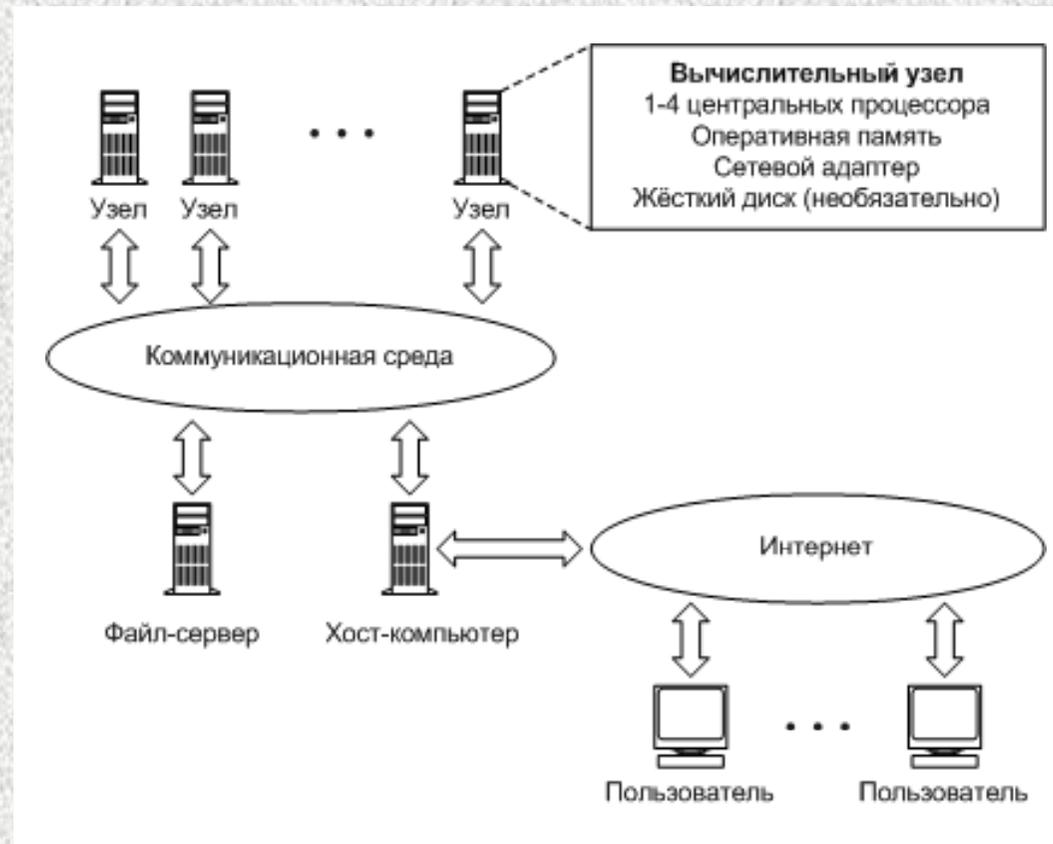
T3D/T3E – DEC ALPHA,

XT3/XT4/XT5 – AMD Opteron.

Топология коммуникационной сети: трехмерный тор.

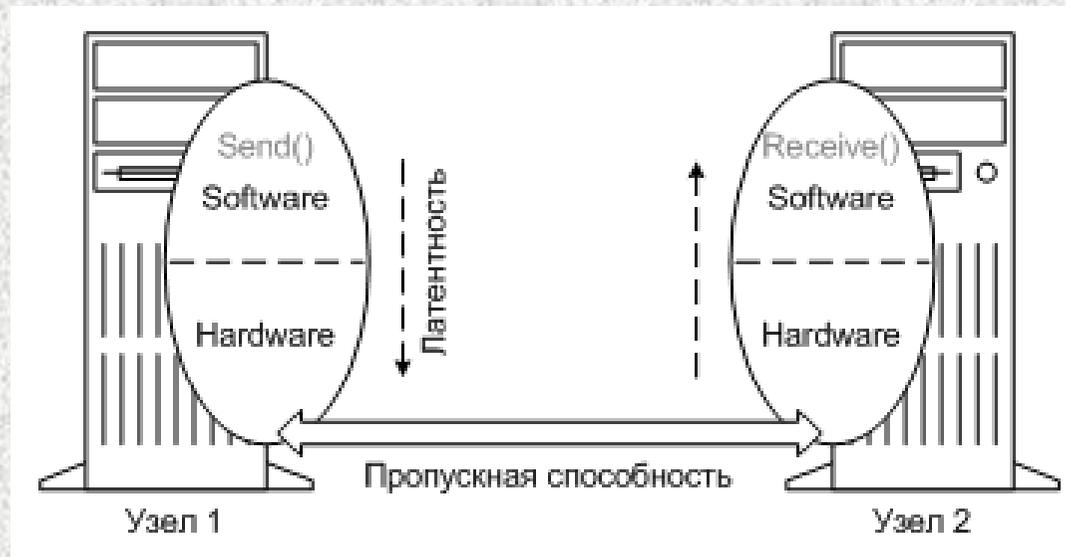


Компьютеры с распределенной памятью (вычислительные кластеры)



Возможность формирования архитектуры кластера с учетом характерных особенностей решаемых задач

Компьютеры с распределенной памятью (вычислительные кластеры)

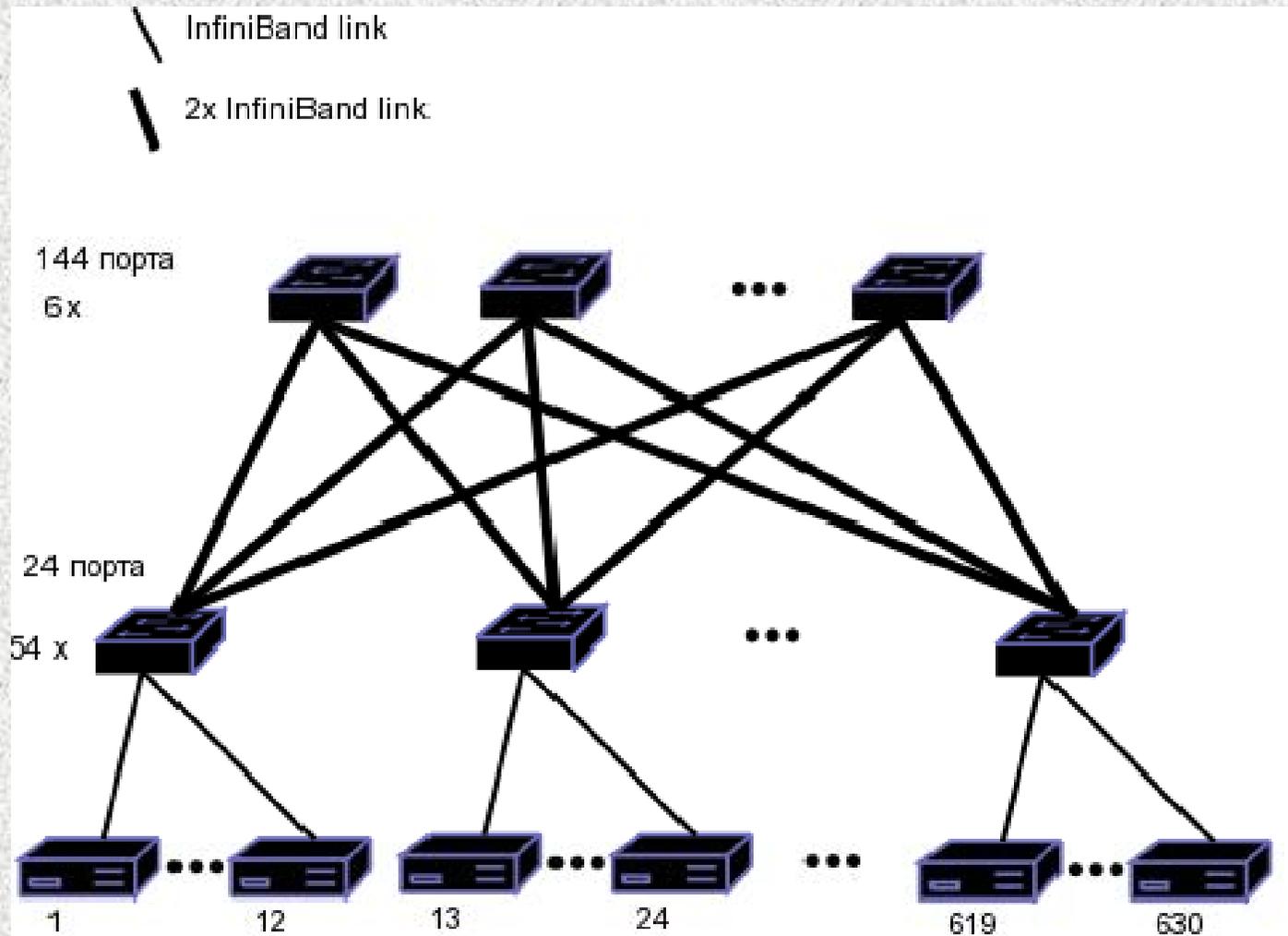


Латентность и пропускная способность сети – основные параметры коммуникационной сети кластеров

Что снижает производительность компьютеров с распределенной памятью?

- 1. Закон Амдала*
- 2. Латентность передачи по сети*
- 3. Пропускная способность каналов передачи данных*
- 4. Особенности использования SMP-узлов*
- 5. Балансировка вычислительной нагрузки*
- 6. Возможность асинхронного счета и передачи данных*
- 7. Особенности топологии коммуникационной сети*
- 8. Производительность отдельных процессоров*
- 9. ...*

Топология FatTree



*Комплексная программа
приемо-сдаточных испытаний
высокопроизводительных компьютеров!*

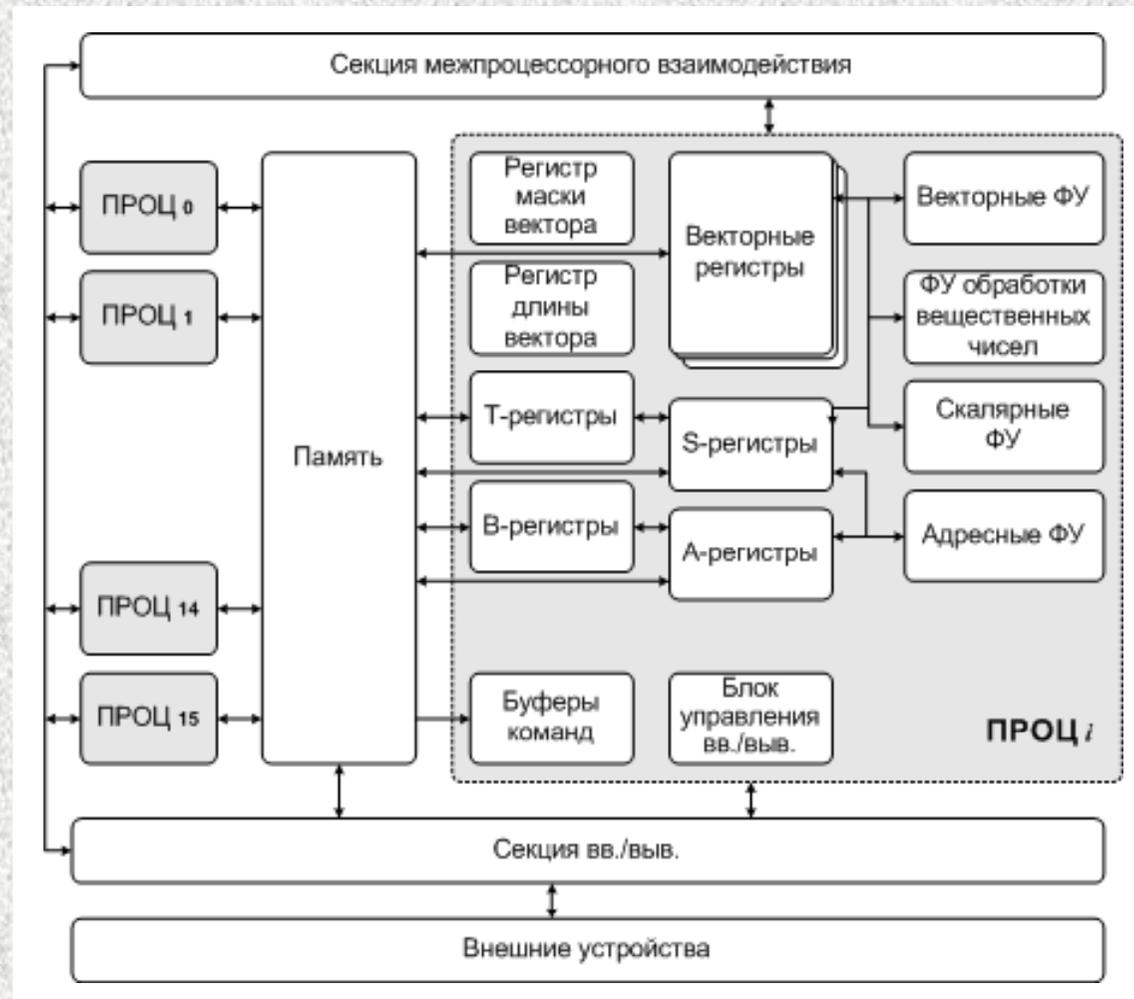
*Эффективность неоднородных
системы ?*

Графические процессоры и HPC

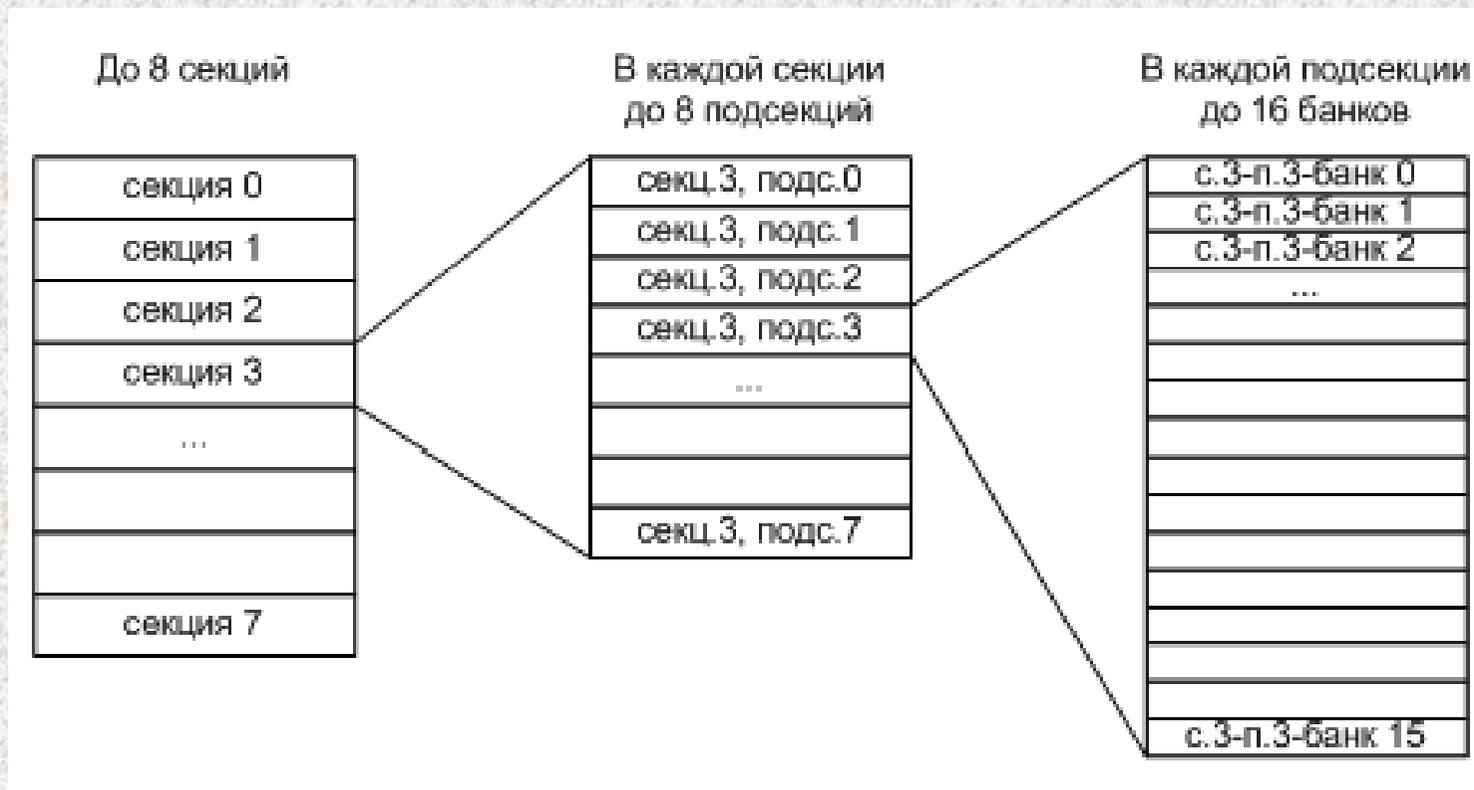
(<http://gpu.parallel.ru>)

Видеокарта	<u>NVidia GeForce 8800 GTX</u>	<u>Nvidia Tesla C1060</u>	<u>AMD Radeon HD3870x2</u>	<u>AMD Radeon HD4850</u>
Кол-во видеокарт	2	1	1	1
Кол-во ПУ на видеокарте	1	1	2	1
Разрядность вещественных вычислений	32	32/64	32 / 64	32 / 64
Типы памяти	видеоОЗУ константная разделяемая статическая регистровая	видеоОЗУ константная разделяемая статическая регистровая	видеоОЗУ константная кэш регистровая	видеоОЗУ константная кэш регистровая
Число и тип потоковых процессоров	128, скалярные	240, скалярные	64, векторные (float4)	160, векторные (float4)
Пиковая производительность, ГФлопс (32 / 64 бит)	345 / --	936/93	ГПУ: 421 / 84 Видеокарта: 842 / 168	1000 / 160
Техпроцесс	90 нм	55 нм	55 нм	55 нм
Мощность	145 Вт	210 Вт	2 x 150 Вт	150 Вт
Энергоэффективность (32 / 64 бит), ГФлопс/Вт	3.57 / --	4.45/0.44	2.80 / 0.56	5.33 / 1.07
Объем видеоОЗУ, МБ	768	4096	2 x 512	512
Пропускная способность видеоОЗУ	86.4	102	2 x 57.6	108

Векторно-конвейерные компьютеры (Cray C90, NEX SX)



Векторно-конвейерные компьютеры (Cray C90, NEX SX)



Что снижает производительность векторно-конвейерных компьютеров?

- 1. Закон Амдала*
- 2. Время разгона конвейера*
- 3. Секционирование векторных команд*
- 4. Конфликты в памяти*
- 5. Каналы процессор-память*
- 6. Операции чтения/записи в векторные регистры*
- 7. Ограниченное число векторных регистров*
- 8. Несбалансированное использование устройств*
- 9. Отсутствие операции деления*
- 10. Перезагрузка буферов команд*
- 11. ...*

Что снижает производительность векторно-конвейерных компьютеров?

```
for( i = 0; i < n*k; i +=k)  
  a[i] = b[i]+c[i]*d;
```

Шаг по памяти	Производительность (Mflops)
1	705.2
2	444.6
4	274.6
8	142.8
16	84.5
32	44.3
64	22.7
128	22.6

Производительность CRAY C90 на операции $a_i = b_i + c_i * d$, $n=1000$.
Влияние конфликтов в памяти

Что снижает производительность векторно-конвейерных компьютеров?

(Влияние конфликтов в памяти)

Do $i = 1, n$

Do $j = 1, n$

Do $k = 1, n$

$$X(i,j,k) = X(i,j,k) + P(k,i) * Y(k,j)$$

$X(N1, N2, N3)$

Фортран: хранение массивов “по столбцам”:

$X(i,j,k) \approx X(i+1,j,k)$ расстояние 1,

$X(i,j,k) \approx X(i,j+1,k)$ расстояние $N1$,

$X(i,j,k) \approx X(i,j,k+1)$ расстояние $N1 * N2$.

Реальное описание массива: $X(40, 40, N)$

$X(i,j,k)$ $X(i,j,k+1)$ расстояние $40 * 40 = 1600 = 64 * 25$.

Решение? Описание массива $X(41, 41, N)$

Пользователь: почему?

$$A_{ijk} = A_{i-1jk} + B_{jk} + B_{jk}, \quad i=1,40; \quad j=1,40; \quad k=1,1000$$

Cray C90, пиковая производительность **960** Mflop/s

```
do k = 1, 1000
```

```
  do j = 1, 40
```

```
    do i = 1, 40
```

```
      A(i,j,k) = A(i-1,j,k)+B(j,k)+B(j,k)
```

Производительность: 20 Mflop/s на Cray C90

Пользователь: почему?

$$A_{ijk} = A_{i-1jk} + B_{jk} + B_{jk}, \quad i=1,40; j=1,40; k=1,1000$$

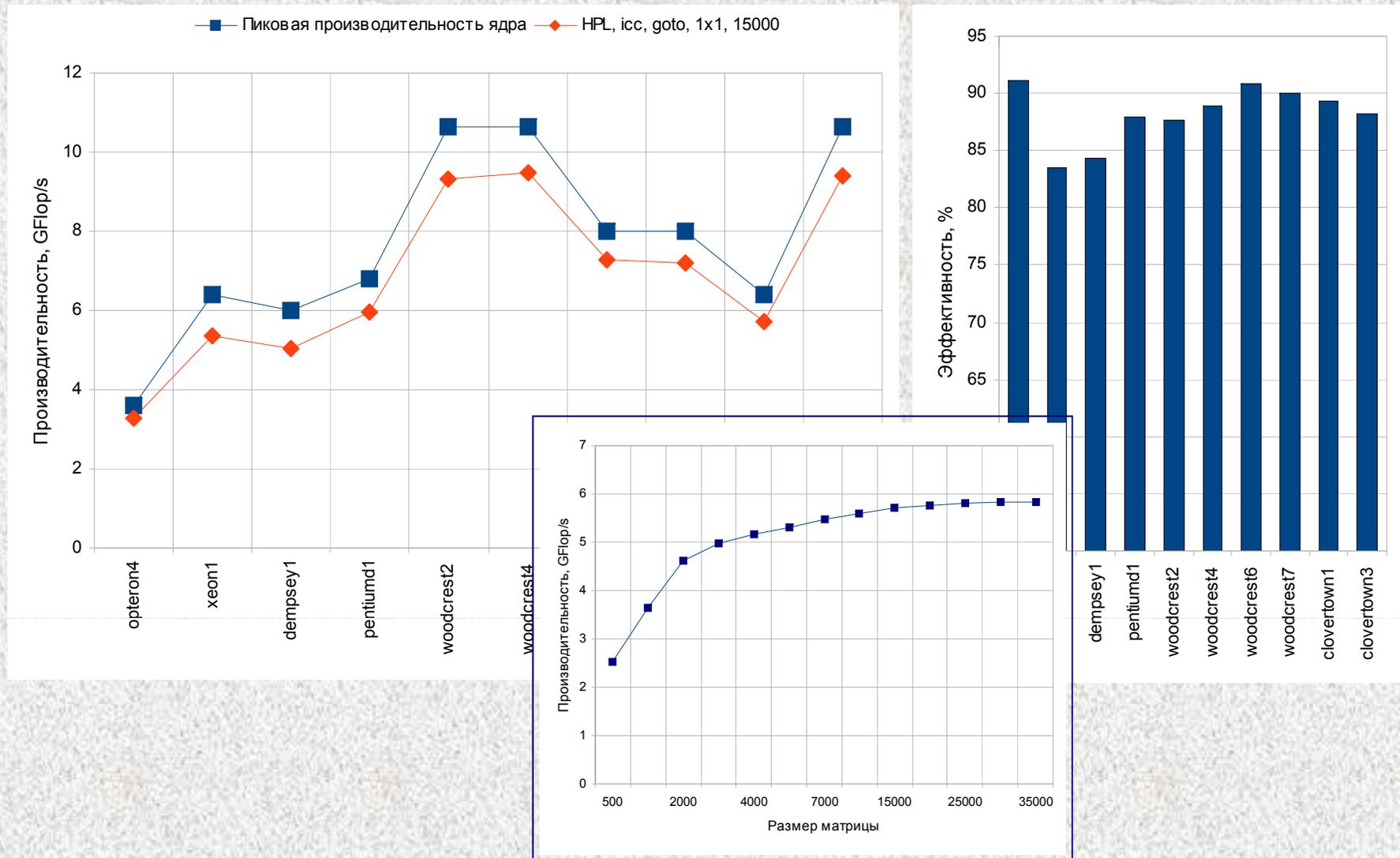
Cray C90, пиковая производительность **960** Mflop/s

```
do i = 1, 40, 2
  do j = 1, 40
    do k = 1, 1000
      A(i,j,k) = A(i-1,j,k)+2*B(j,k)
      A(i+1,j,k) = A(i,j,k)+2*B(j,k)
    do k = 1, 1000
      do j = 1, 40
        do i = 1, 40
          A(i,j,k) = A(i-1,j,k)+B(j,k)+B(j,k)
        end do
      end do
    end do
  end do
end do
```

Производительность: **700** Mflop/s на Cray C90

Что снижает производительность процессоров?

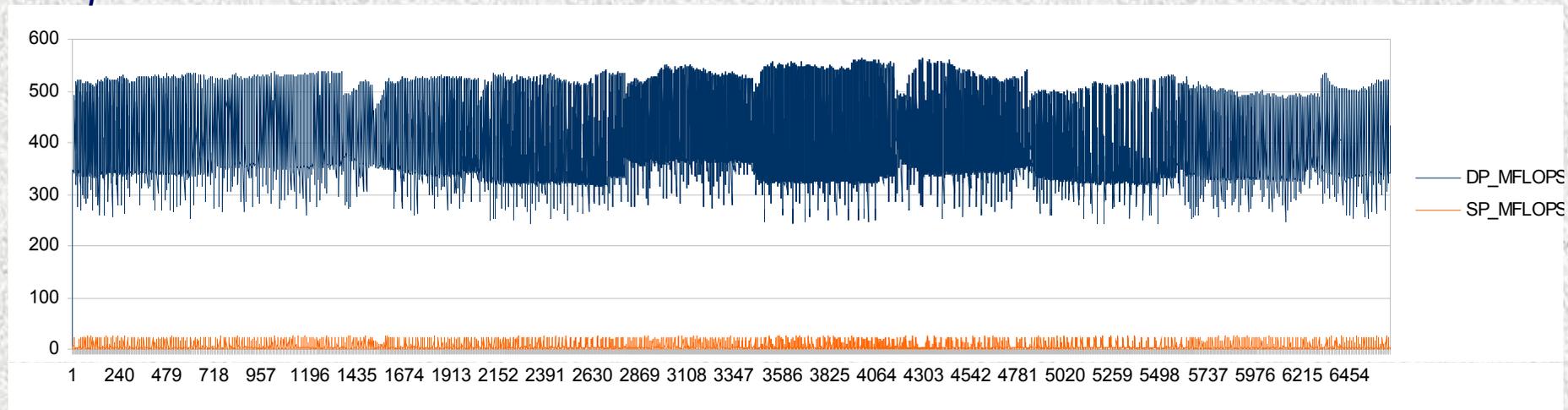
Высокая эффективность приложений (Linpack, 2GB, 16GB)



Компьютерный дизайн лекарств

(Intel -fast, исследование эффективности, Clovertown 2.66GHz)

Реальная производительность,
Mflops



КПД процессора на задаче: 4% !!!

Процессорный полигон НИВЦ МГУ

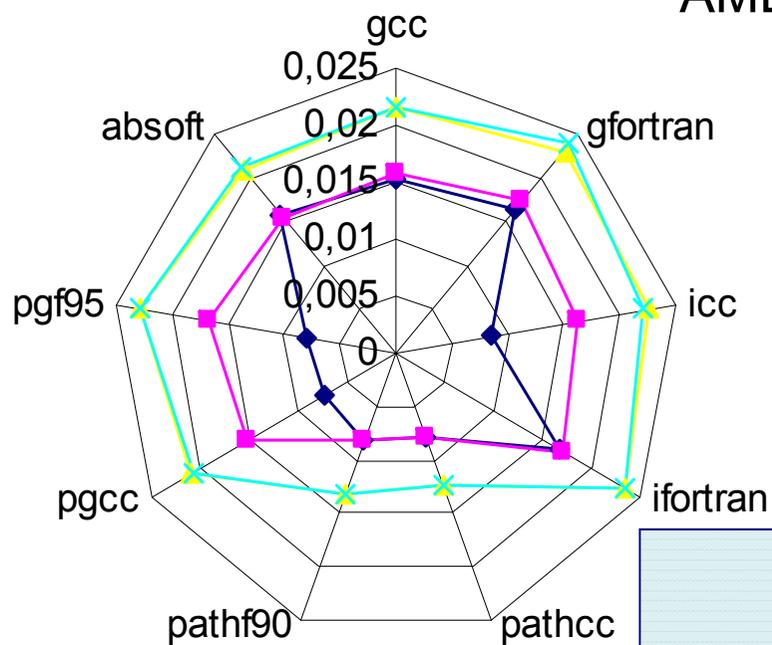


15 серверов **Intel**
(Pentium D, Xeon EMT,
Dempsey, Woodcrest,
Clowertown...),
6 серверов **AMD**
(Opteron single/dual/quad),
1 сервер **IBM**
(Power5),

22 сервера, 1-2 U
35 процессоров, 79 ядер
597 Гфлопс, 150 Гбайт ОП

Процессорный полигон НИВЦ МГУ – набор тестов “Stream” + компиляторы

AMD Opteron 244



◆ Copy
■ Scale
▲ Add
× Triad

$$a_i = b_i$$

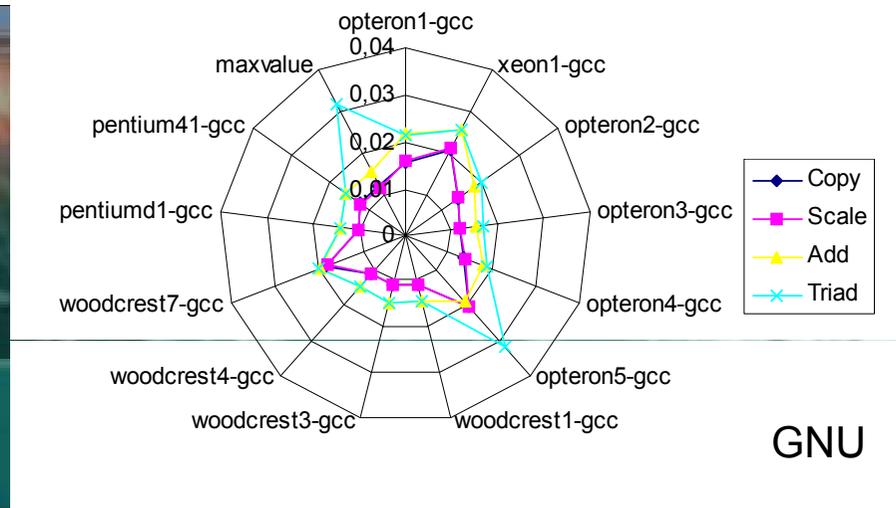
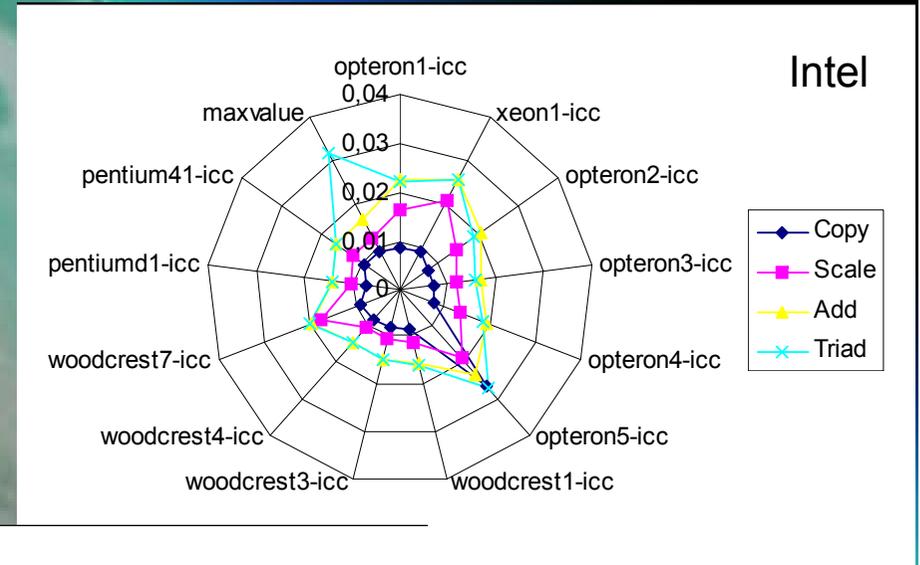
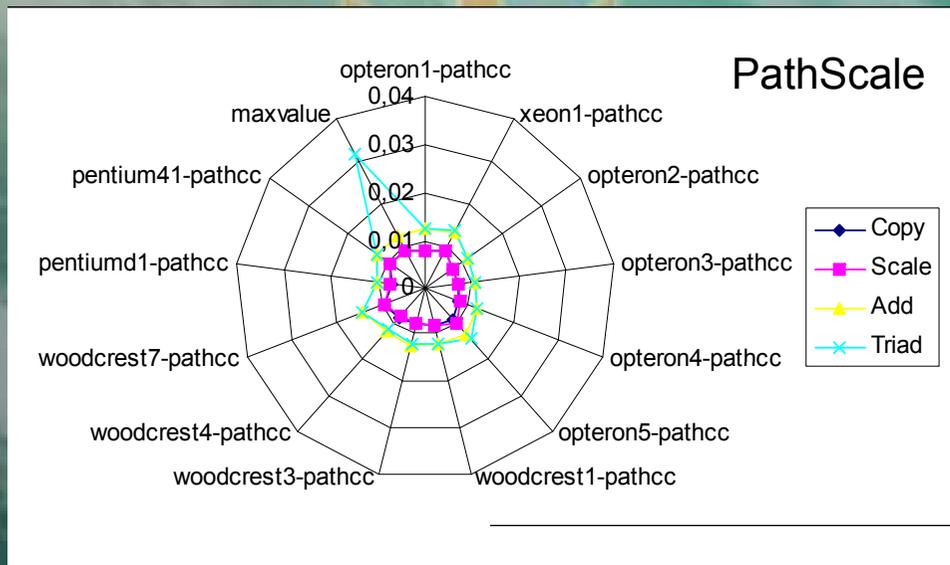
$$a_i = q * b_i$$

$$a_i = b_i + c_i$$

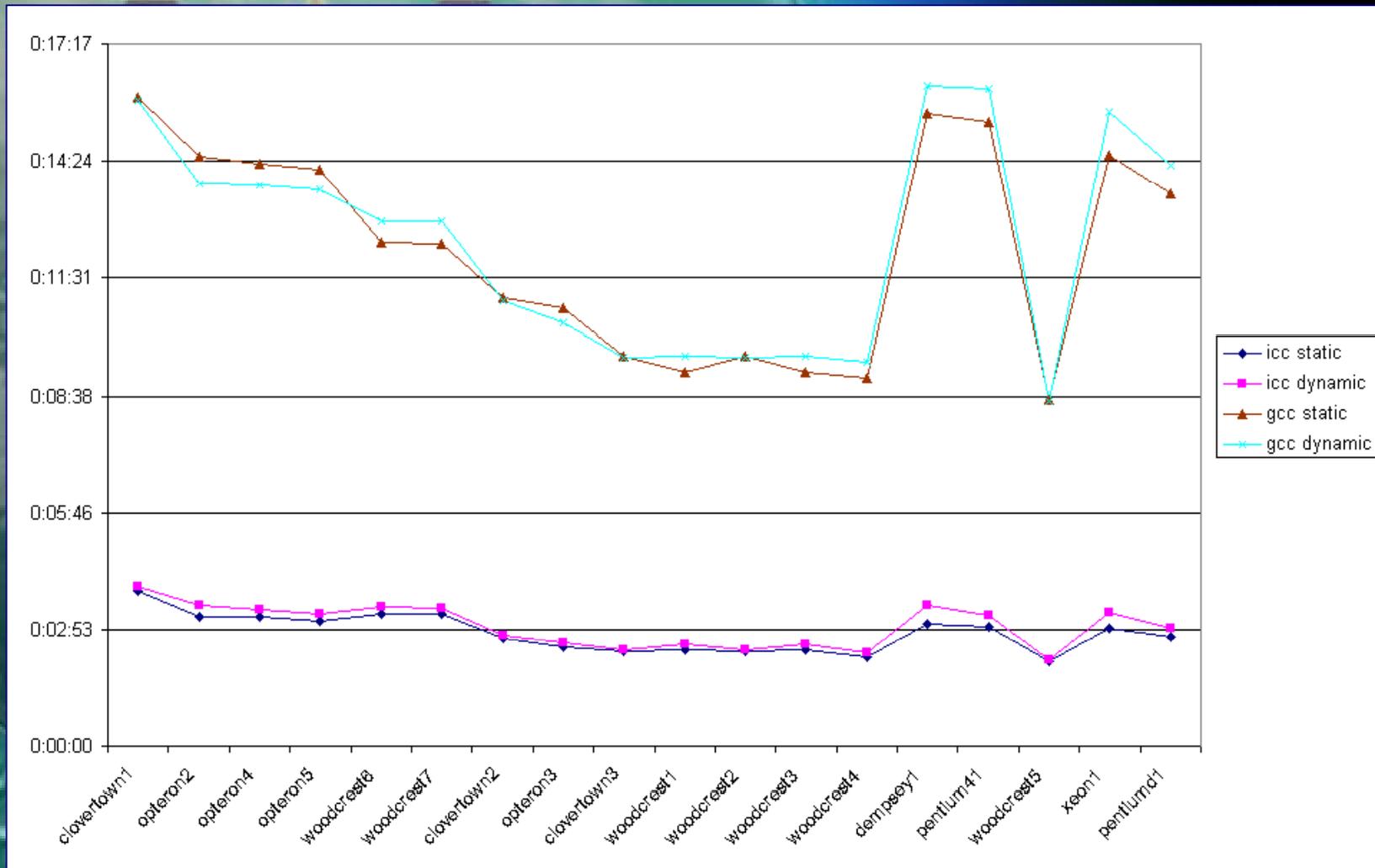
$$a_i = b_i + q * c_i$$

GNU Compilers,
Intel Compilers,
Portland Group Inc. Compilers,
PathScale EKOPath Compiles Suite,
Absoft Fortran.

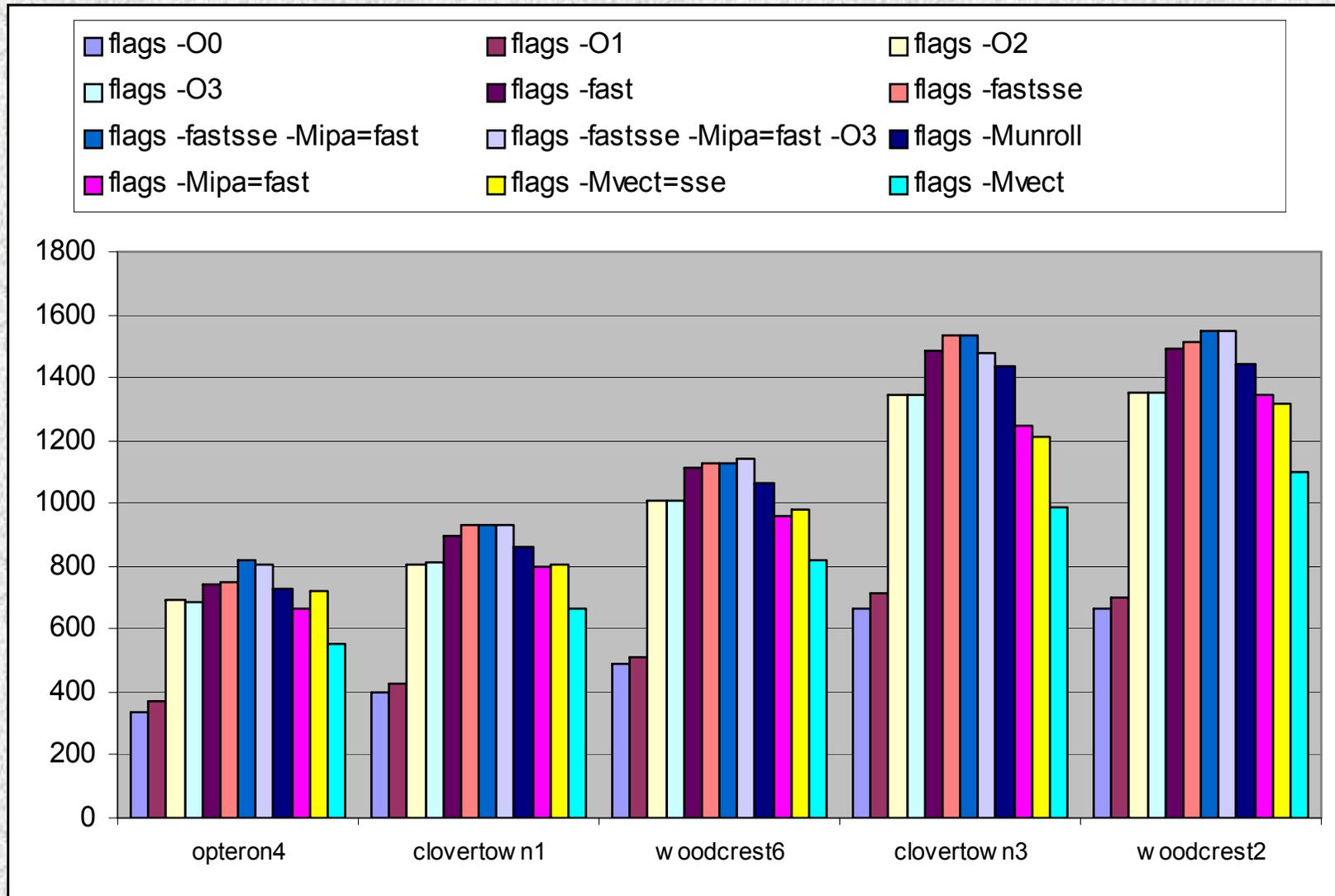
Процессорный полигон НИВЦ МГУ – набор тестов “Stream” + компиляторы



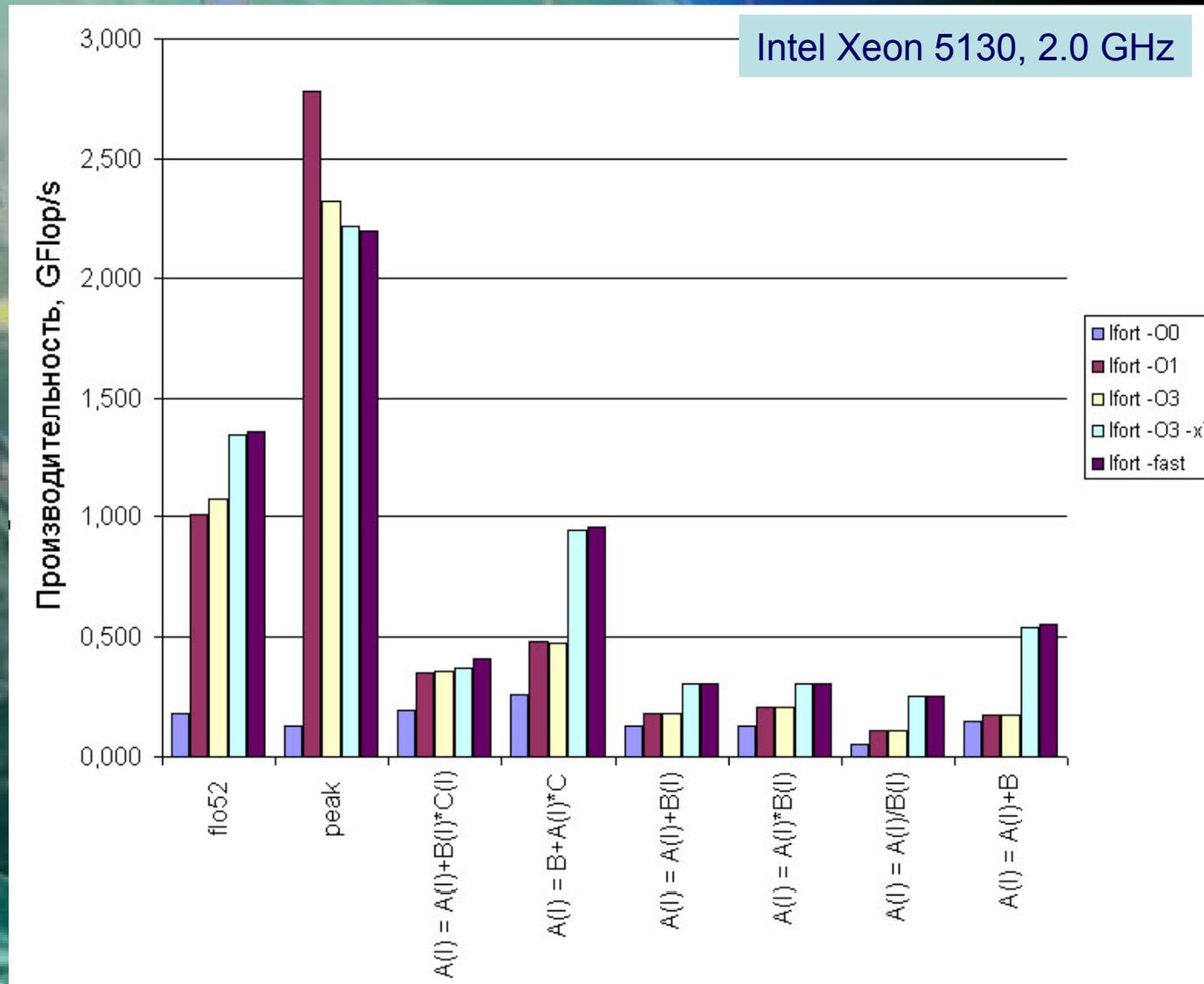
Процессорный полигон НИВЦ МГУ – анализ работы компиляторов



Опции компилятора PGI (FLO52 benchmark, Mflops)



Искусство компиляции ...



Анализ эффективности программ

АНАЛИЗ АЛГОРИТМИЧЕСКОГО ПОДХОДА



АНАЛИЗ СТРУКТУРЫ ПРИКЛАДНОЙ ПРОГРАММЫ



АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМ РАЗРАБОТКИ ПО



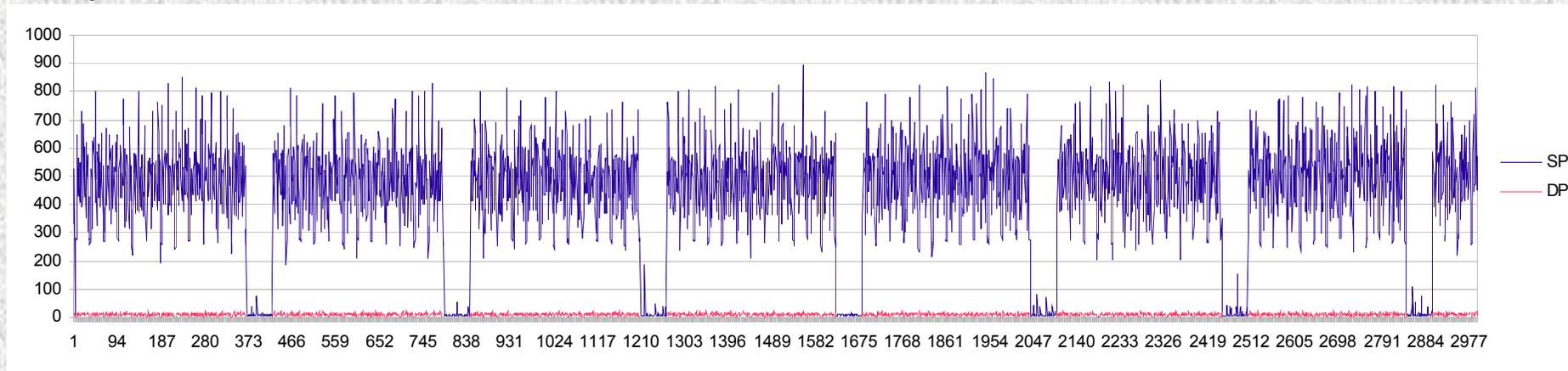
АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМНОГО ПО



АНАЛИЗ КОНФИГУРАЦИИ КОМПЬЮТЕРА

Анализ эффективности программ

Реальная производительность,
Mflops



Анализ эффективности программ

АНАЛИЗ АЛГОРИТМИЧЕСКОГО ПОДХОДА



АНАЛИЗ СТРУКТУРЫ ПРИКЛАДНОЙ ПРОГРАММЫ



АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМ РАЗРАБОТКИ ПО



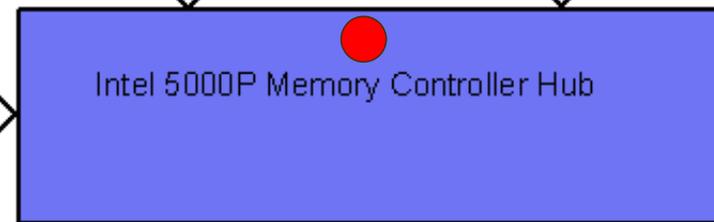
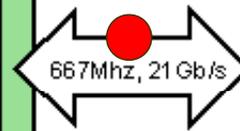
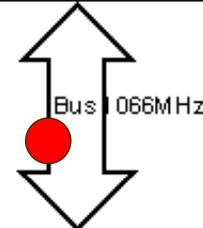
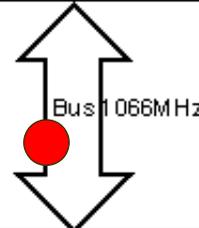
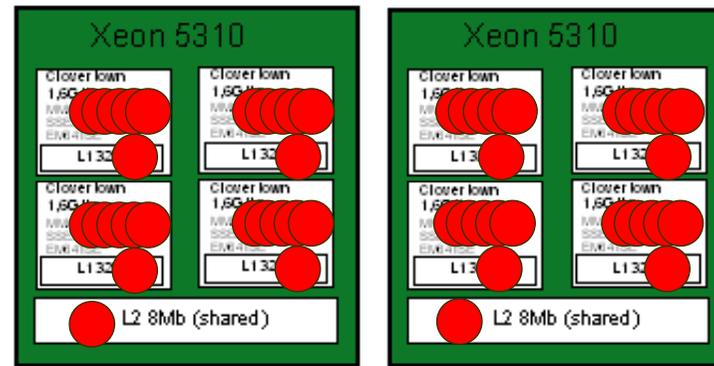
АНАЛИЗ ЭФФЕКТИВНОСТИ СИСТЕМНОГО ПО



АНАЛИЗ КОНФИГУРАЦИИ КОМПЬЮТЕРА

#18 clovertown1

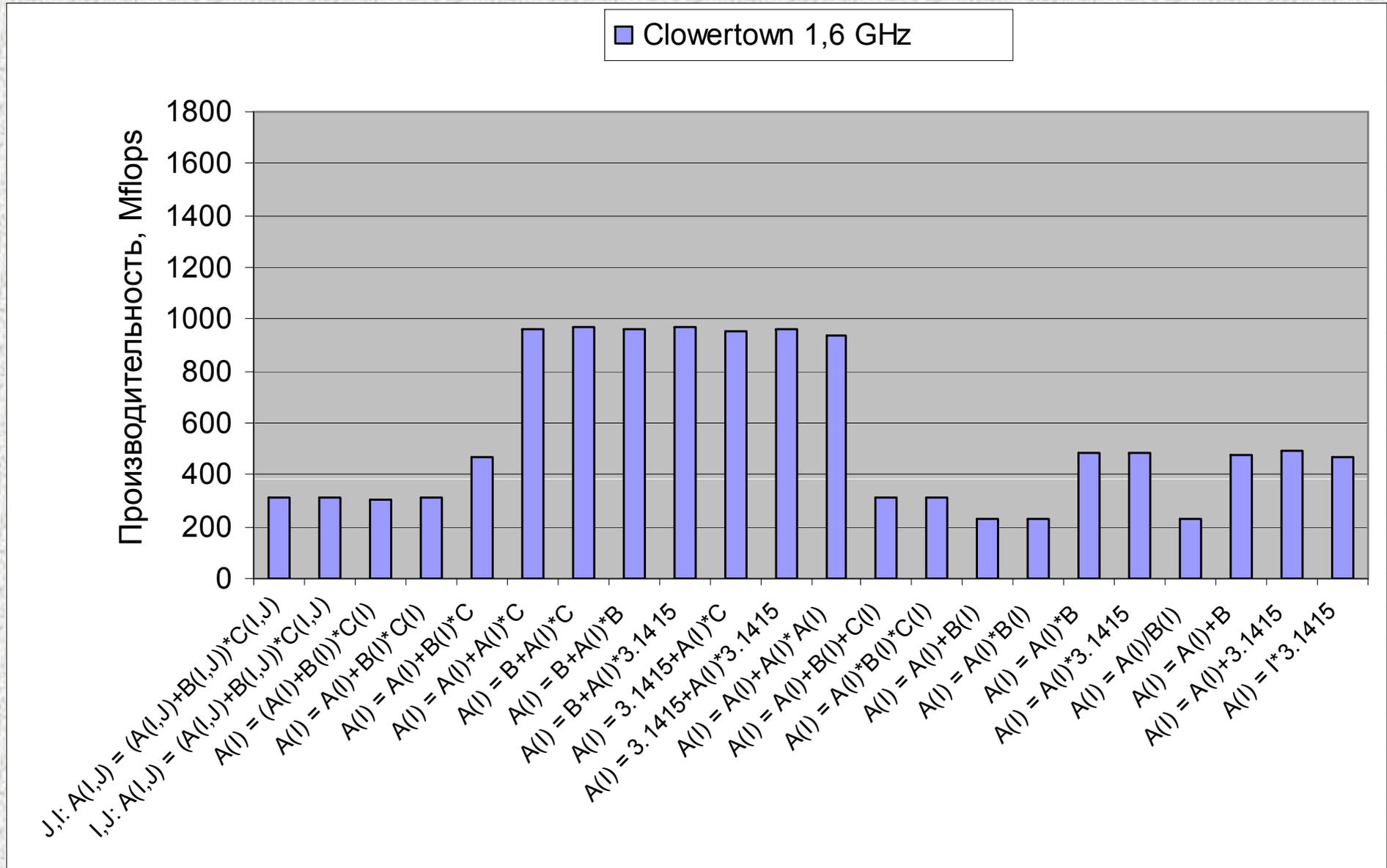
Tyan Tempest i5000PX (S5380) 2*Socket771



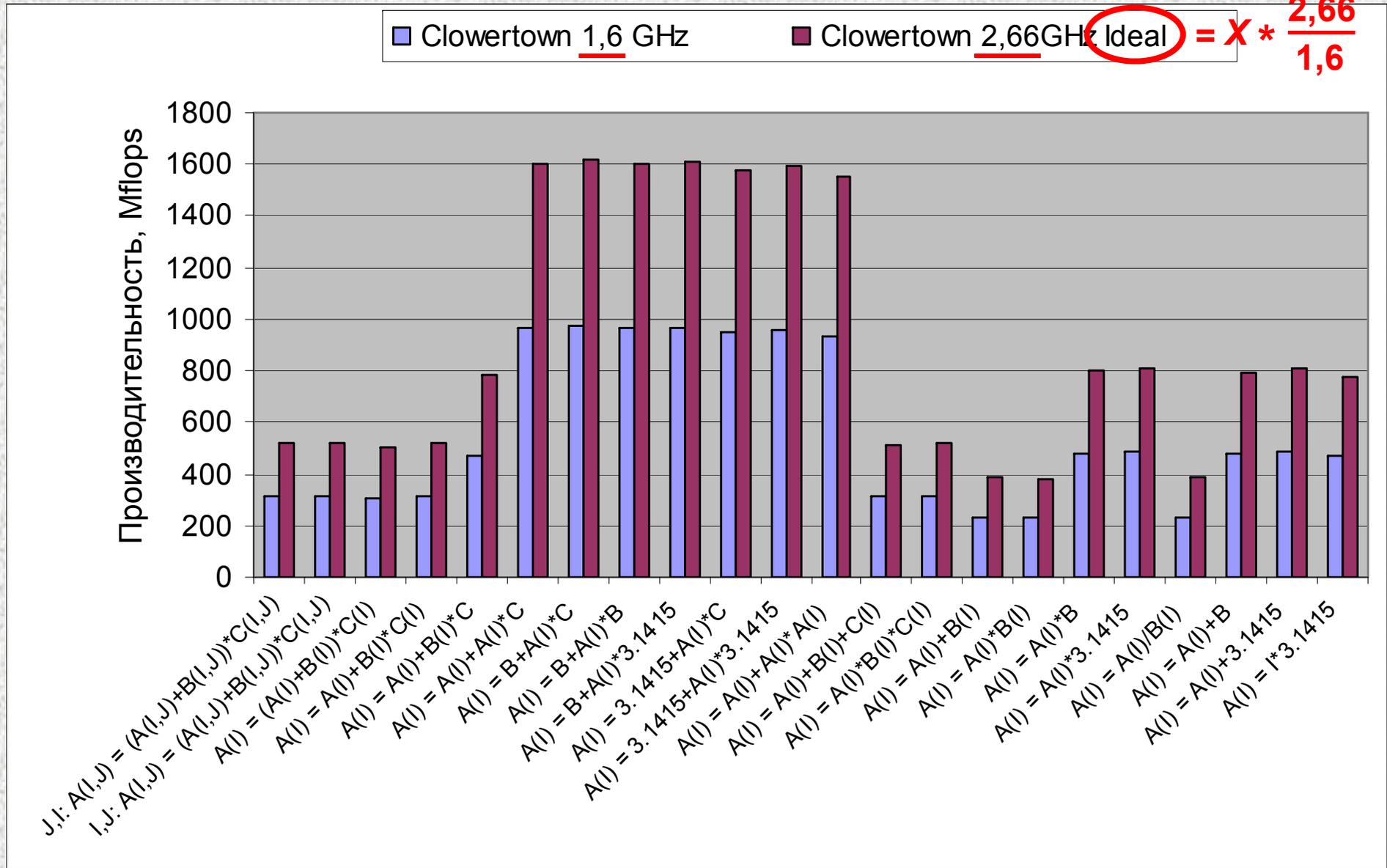
Что влияет на производительность узлов кластеров?

- *использование суперскалярности,*
- *неполная загрузка конвейерных функциональных устройств,*
- *пропускная способность кэш-памяти, основной памяти, каналов передачи данных,*
- *объем кэш-памяти различных уровней и основной памяти,*
- *степень ассоциативности кэш-памяти различных уровней,*
- *несовпадение размера строк кэш-памяти различных уровней,*
- *несовпадение степени ассоциативности кэш-памяти различных уровней,*
- *стратегия замещения строк кэш-памяти различных уровней,*
- *стратегия записи данных, принятая при работе с подсистемами памяти,*
- *расслоение оперативной памяти (структура банков),*
- *частота работы оперативной памяти,*
- *частота FSB,*
- *ширина FSB,*
- *несоответствие базовых частот: процессора, FSB и оперативной памяти,*
- *влияние "NUMA" в серверах с архитектурой ccNUMA,*
- *влияние "cc" в серверах с архитектурой ccNUMA,*
- *влияние ОС (менеджер виртуальной памяти, накладные расходы на сборку мусора и выделение памяти).*

Производительность на базовых операциях

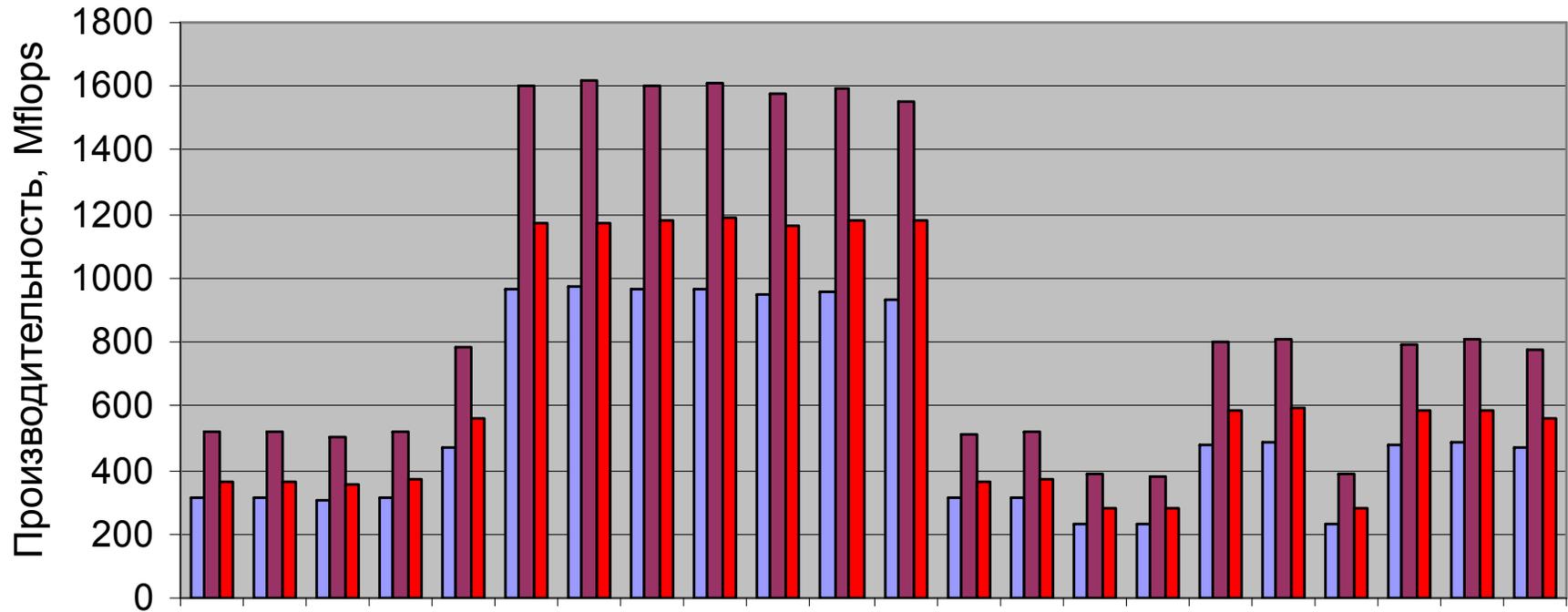


Масштабирование по частоте CPU?



Реальное масштабирование на практике...

■ Clowertown 1,6 GHz
 ■ Clowertown 2,66GHz Ideal
 ■ Clowertown 2,66 GHz



$J, I: A(I, J) = (A(I, J) + B(I, J)) * C(I, J)$
 $I, J: A(I, J) = (A(I, J) + B(I, J)) * C(I, J)$
 $A(I) = (A(I) + B(I)) * C(I)$
 $A(I) = A(I) + B(I) * C(I)$
 $A(I) = A(I) + B(I) * C$
 $A(I) = B + A(I) * C$
 $A(I) = B + A(I) * C$
 $A(I) = B + A(I) * B$
 $A(I) = 3.1415 + A(I) * 3.1415$
 $A(I) = 3.1415 + A(I) * 3.1415$
 $A(I) = A(I) + A(I) * A(I)$
 $A(I) = A(I) + B(I) + C(I)$
 $A(I) = A(I) * B(I) * C(I)$
 $A(I) = A(I) + B(I)$
 $A(I) = A(I) * B(I)$
 $A(I) = A(I) * B$
 $A(I) = A(I) * 3.1415$
 $A(I) = A(I) / B(I)$
 $A(I) = A(I) + B$
 $A(I) = A(I) + 3.1415$
 $A(I) = I * 3.1415$

Реальное масштабирование на практике...

		<i>CPU</i>	<i>FSB</i>
<i>Clowertown</i>	–	1,6 GHz	1,066 GHz
<i>Clowertown</i>	–	2,66 GHz	1,333 GHz

CPU / FSB – это число тактов процессора на каждый такт работы системной шины:

для *Clowertown* 1,6 GHz – это 1,5

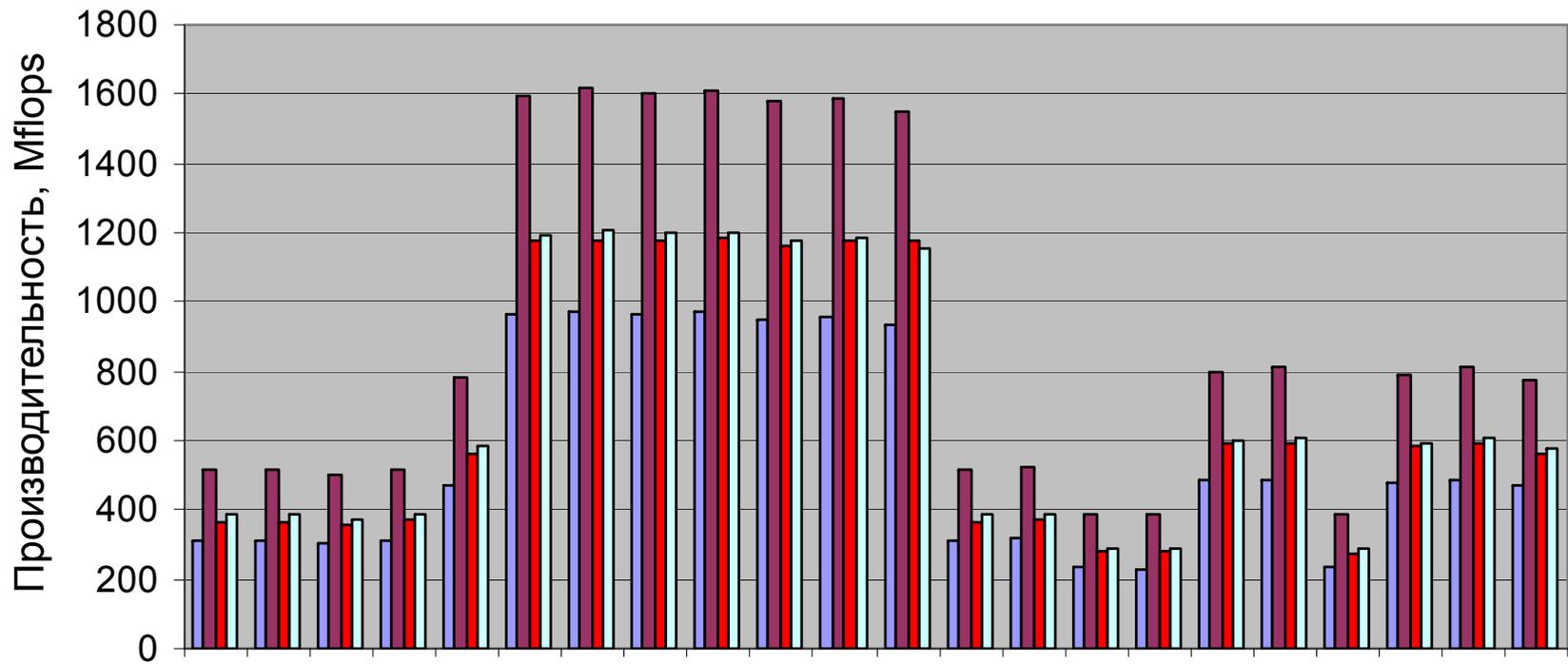
для *Clowertown* 2,66 GHz – это 2

$1,5 / 2 = 0,75$ – замедление работы с памятью

$(2,66 / 1,6) * 0,75 = 1,24$ – реальное ускорение

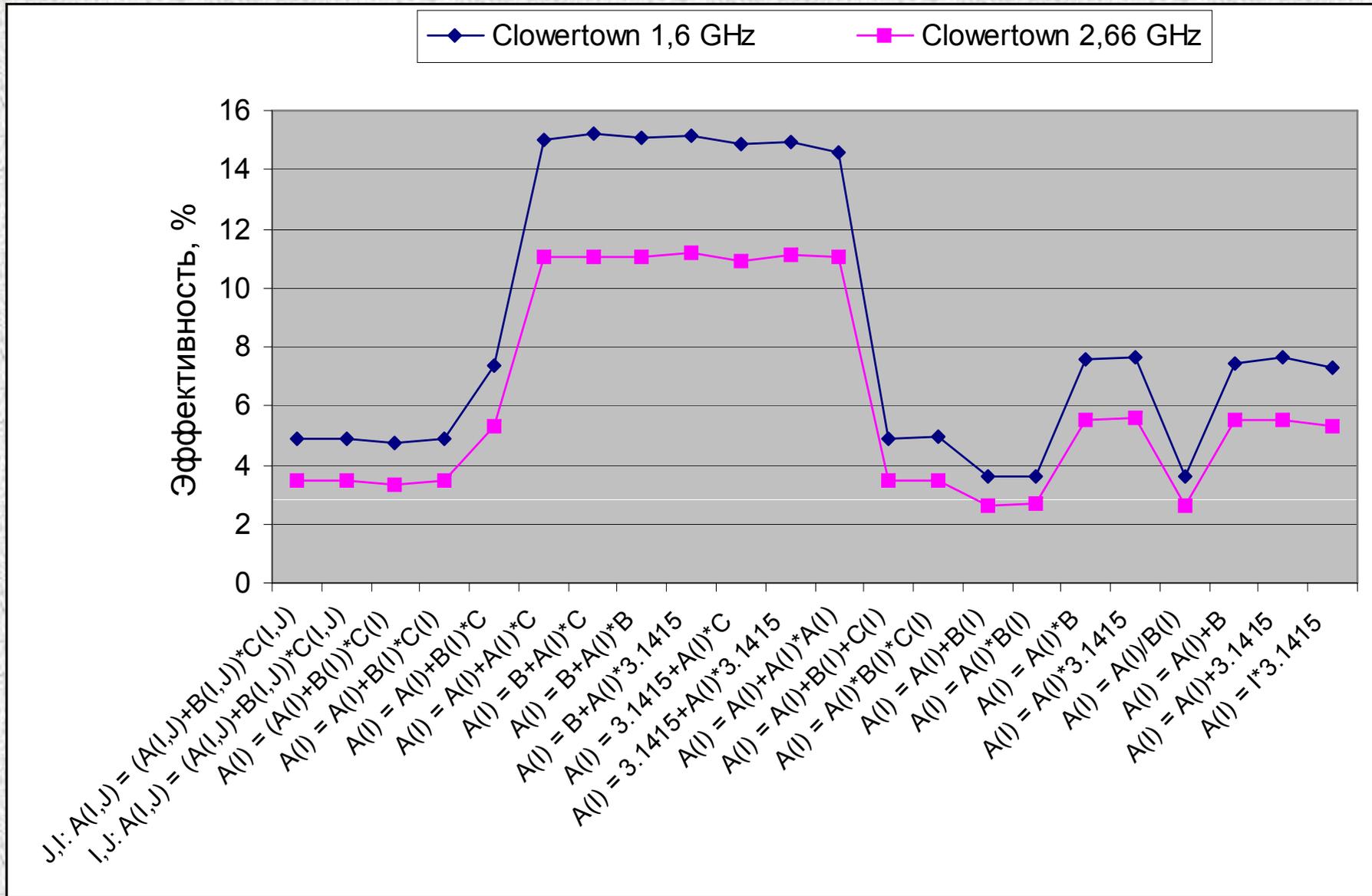
Теория и практика масштабирования

■ Clowertown 1,6 GHz
 ■ Clowertown 2,66GHz Ideal
 ■ Clowertown 2,66 GHz
 ■ Clowertown *1,24

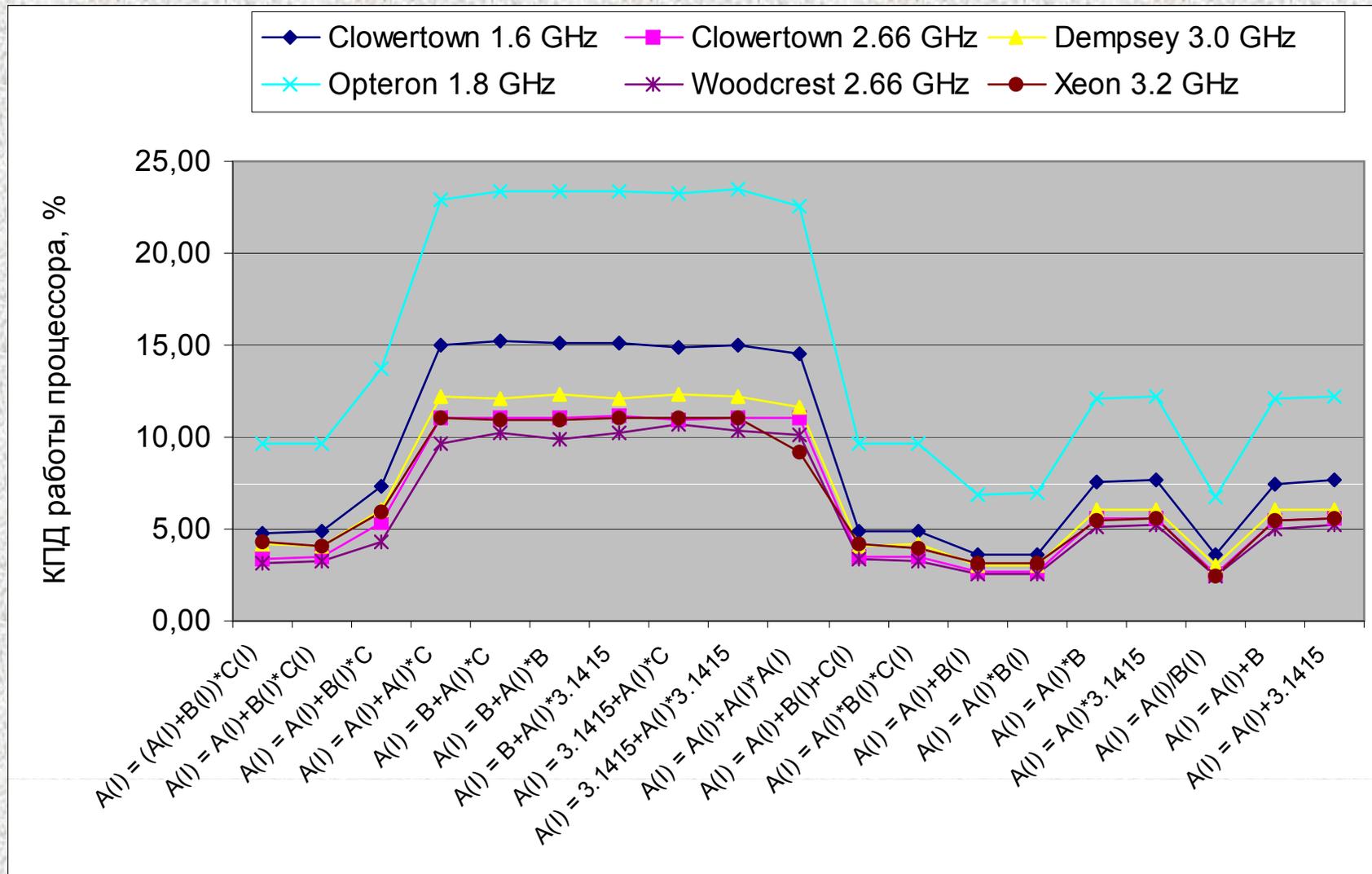


J,I: A(I,J) = (A(I,J)+B(I,J))*C(I,J)
 I,J: A(I,J) = (A(I,J)+B(I,J))*C(I,J)
 A(I) = (A(I)+B(I))*C(I)
 A(I) = A(I)+B(I)*C(I)
 A(I) = A(I)+B(I)*C
 A(I) = B+A(I)*C
 A(I) = B+A(I)*C
 A(I) = B+A(I)*B
 A(I) = 3.1415+A(I)*C
 A(I) = 3.1415+A(I)*C
 A(I) = A(I)+A(I)*A(I)
 A(I) = A(I)+B(I)*A(I)
 A(I) = A(I)*B(I)+C(I)
 A(I) = A(I)*C(I)
 A(I) = A(I)+B(I)
 A(I) = A(I)*B(I)
 A(I) = A(I)*B
 A(I) = A(I)*3.1415
 A(I) = A(I)/B(I)
 A(I) = A(I)+B
 A(I) = A(I)+3.1415
 A(I) = I*3.1415

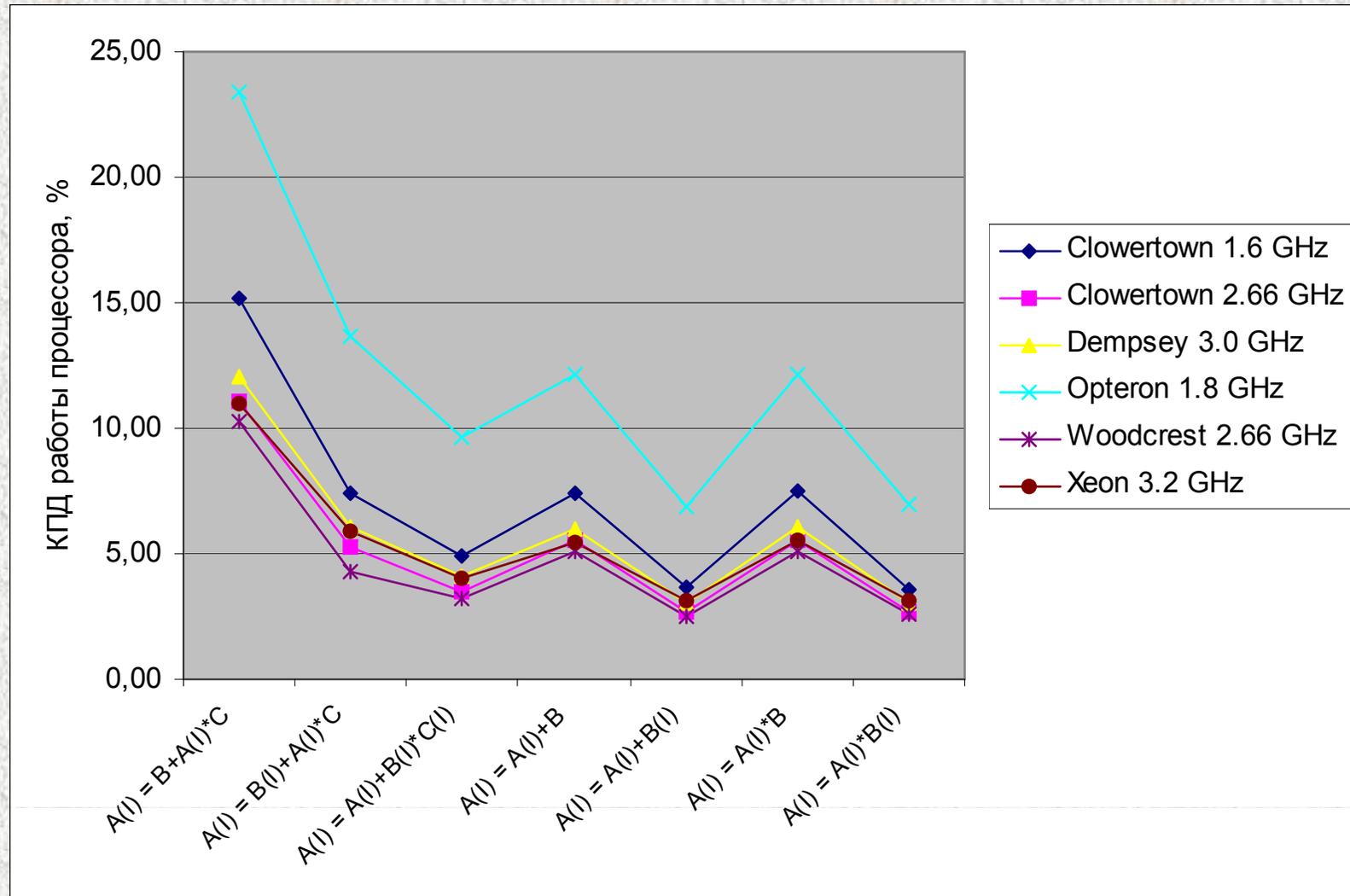
КПД работы процессоров ...



КПД работы процессоров ...



Процессоры и массивы...



Умножение матриц

(параллельная структура и распределение данных)

Do i = 1, n

Do j = 1, n

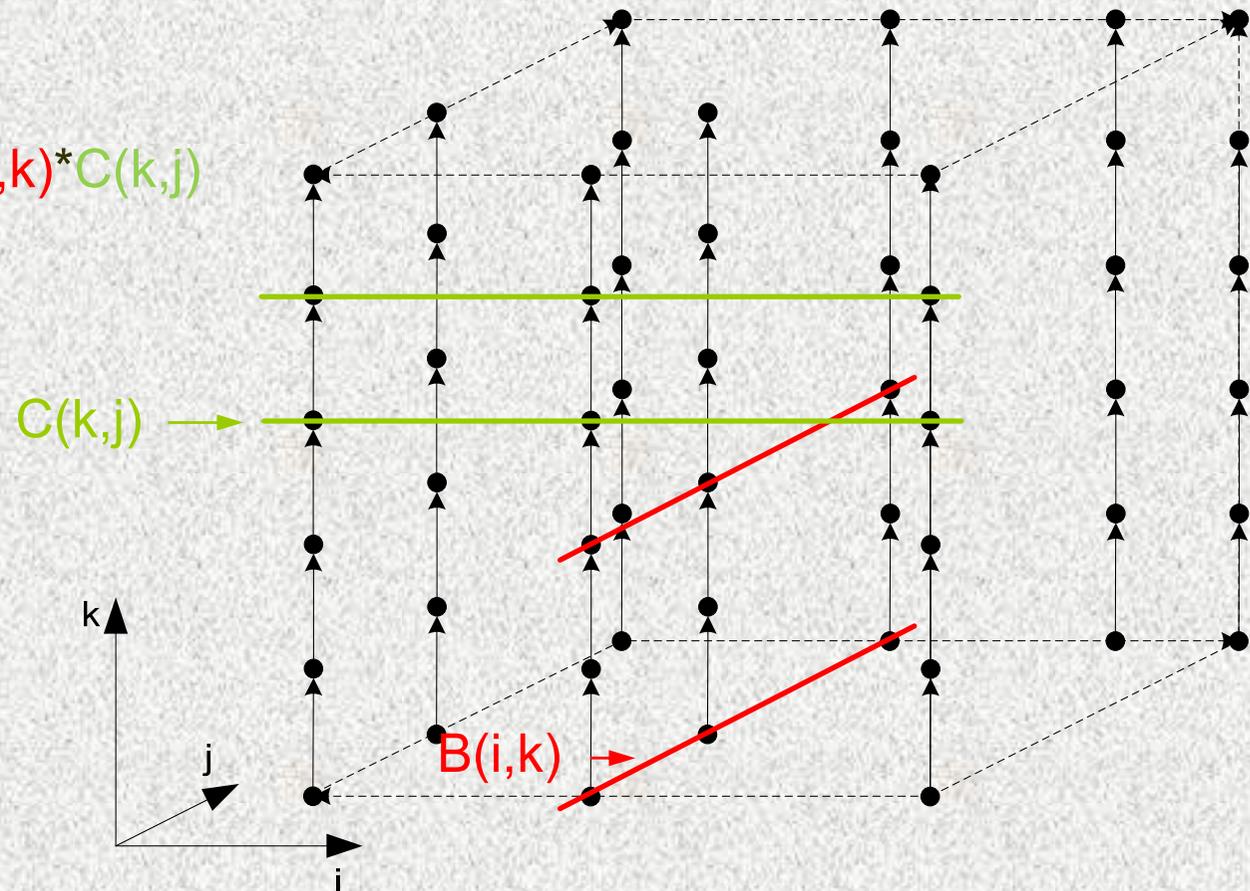
Do k = 1, n

$A(i,j) = A(i,j) + B(i,k) * C(k,j)$

End do

End do

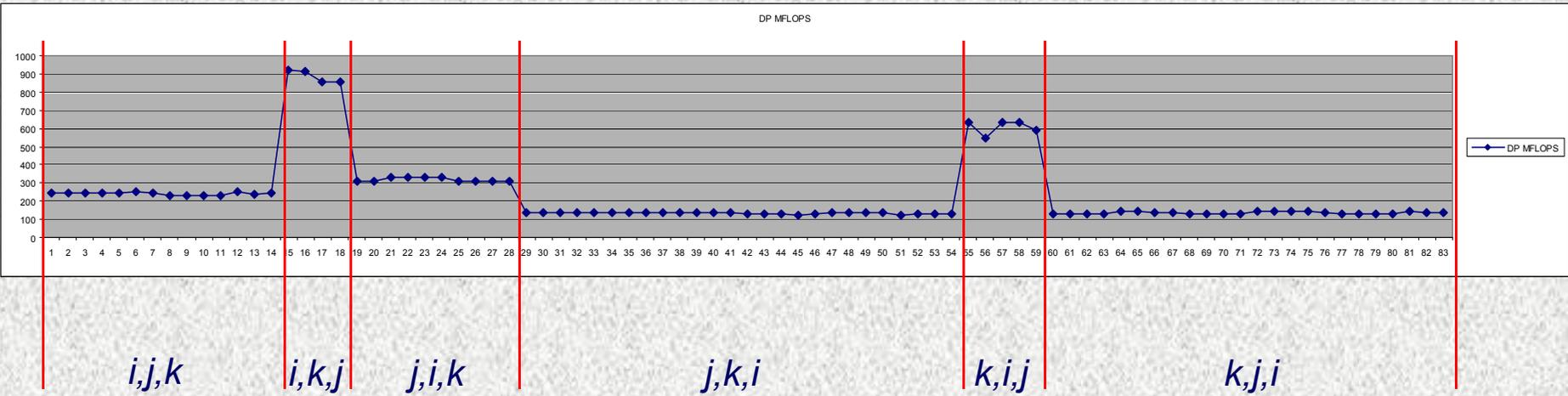
End do



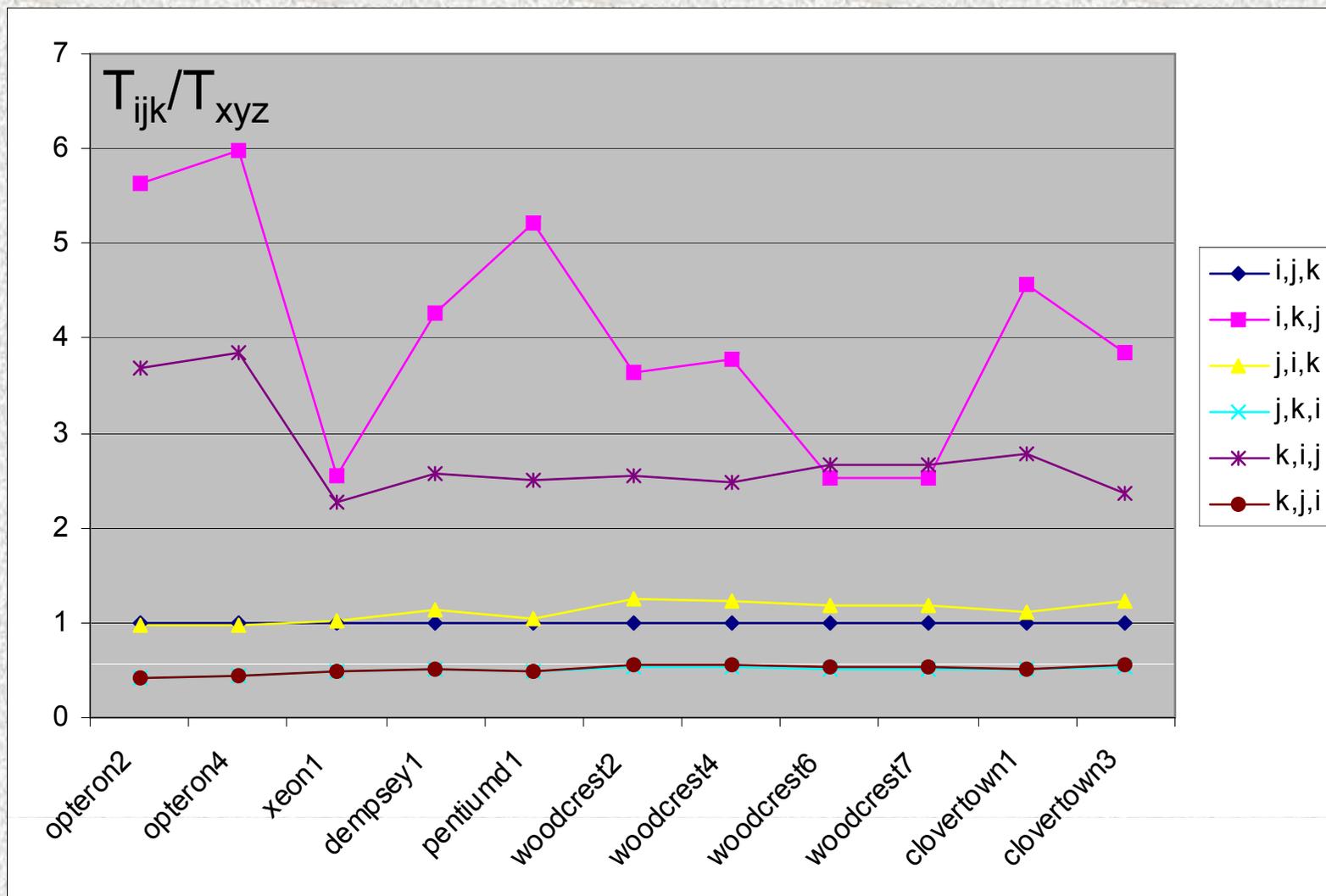
Умножение матриц

(Intel -fast, run-time, перестановка циклов, Woodcrest2)

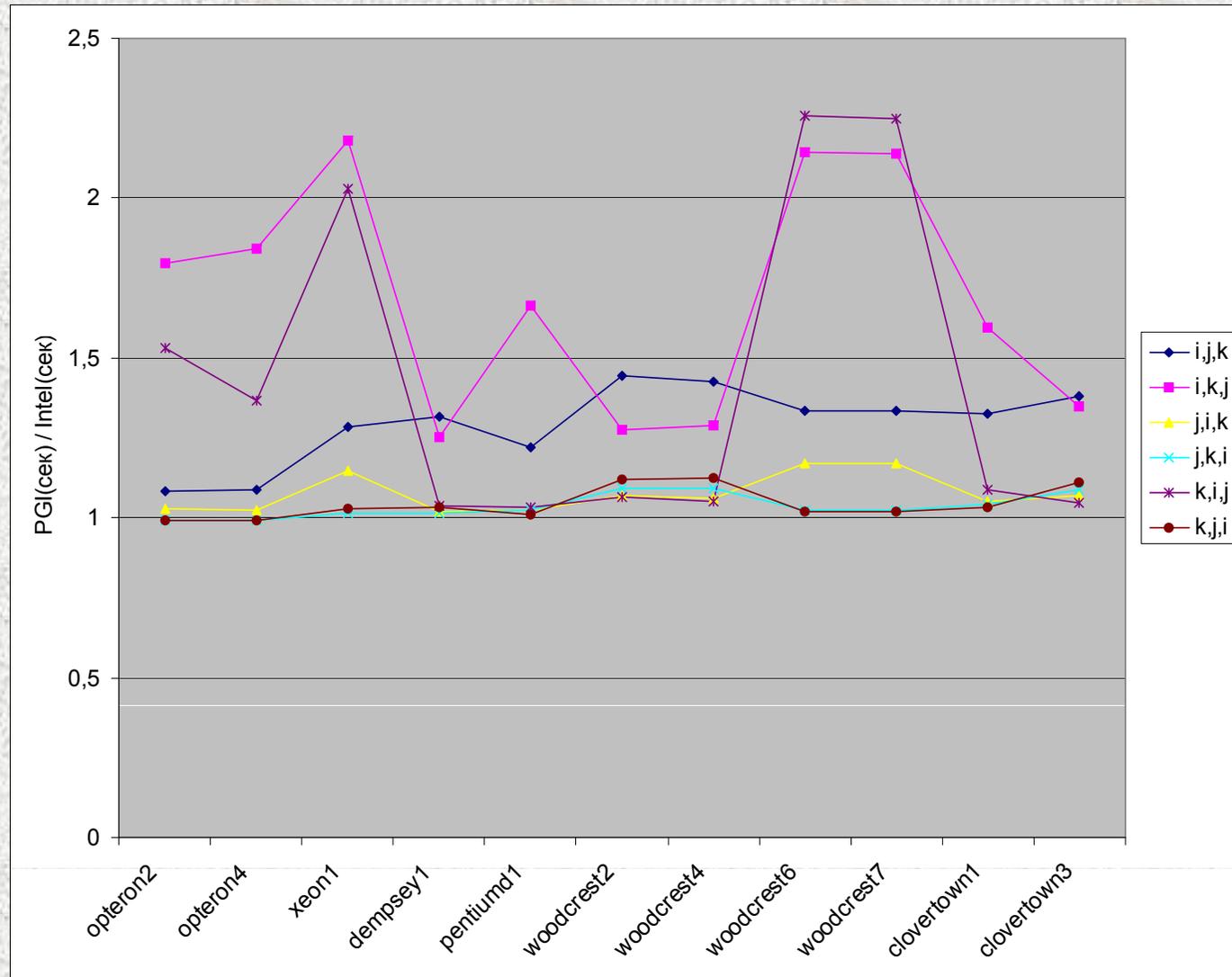
Реальная производительность, Mflops



Умножение матриц (перестановка циклов, сравнение с i,j,k)



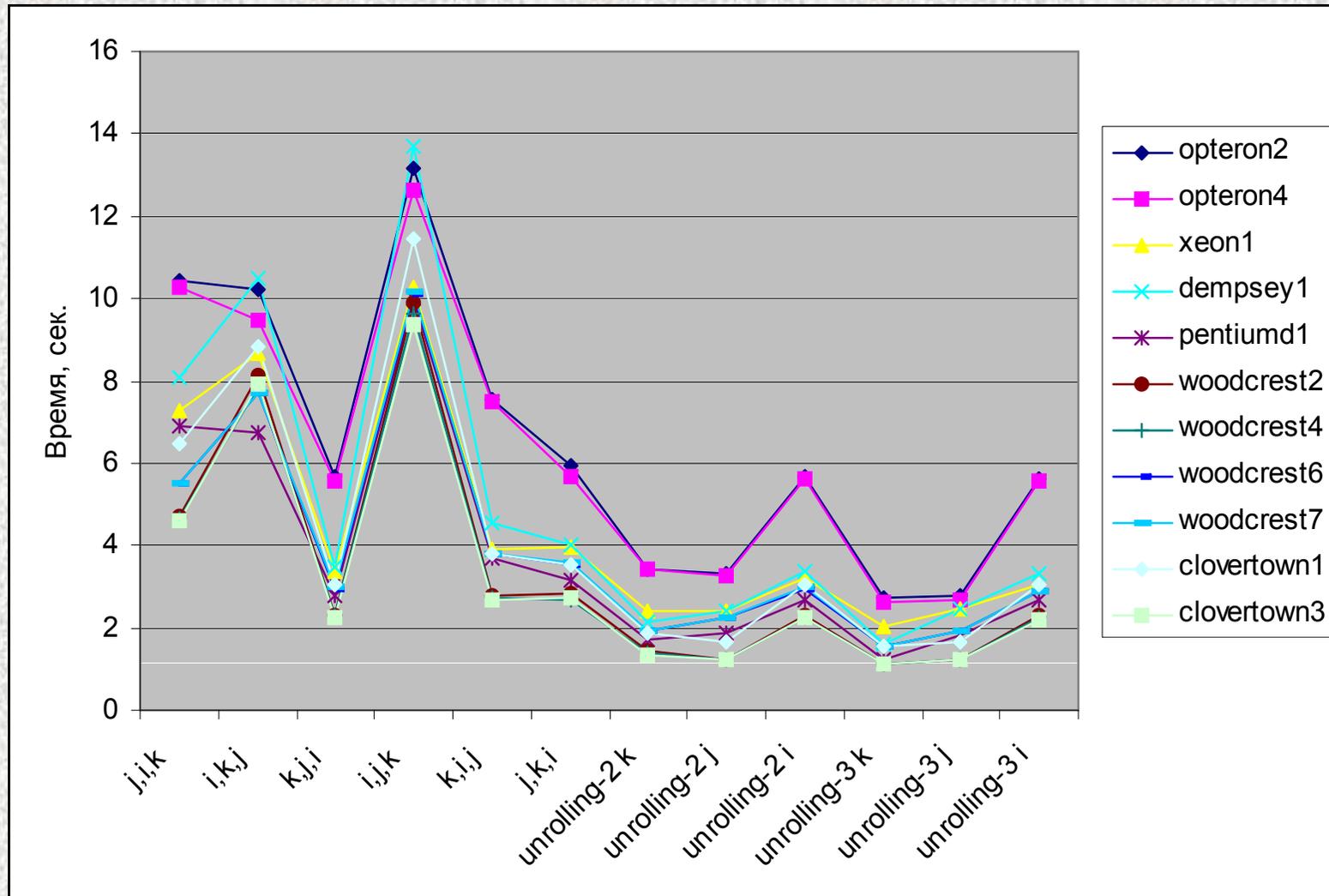
Сравнение компиляторов: Intel и PGI (умножение матриц, PGI/Intel)



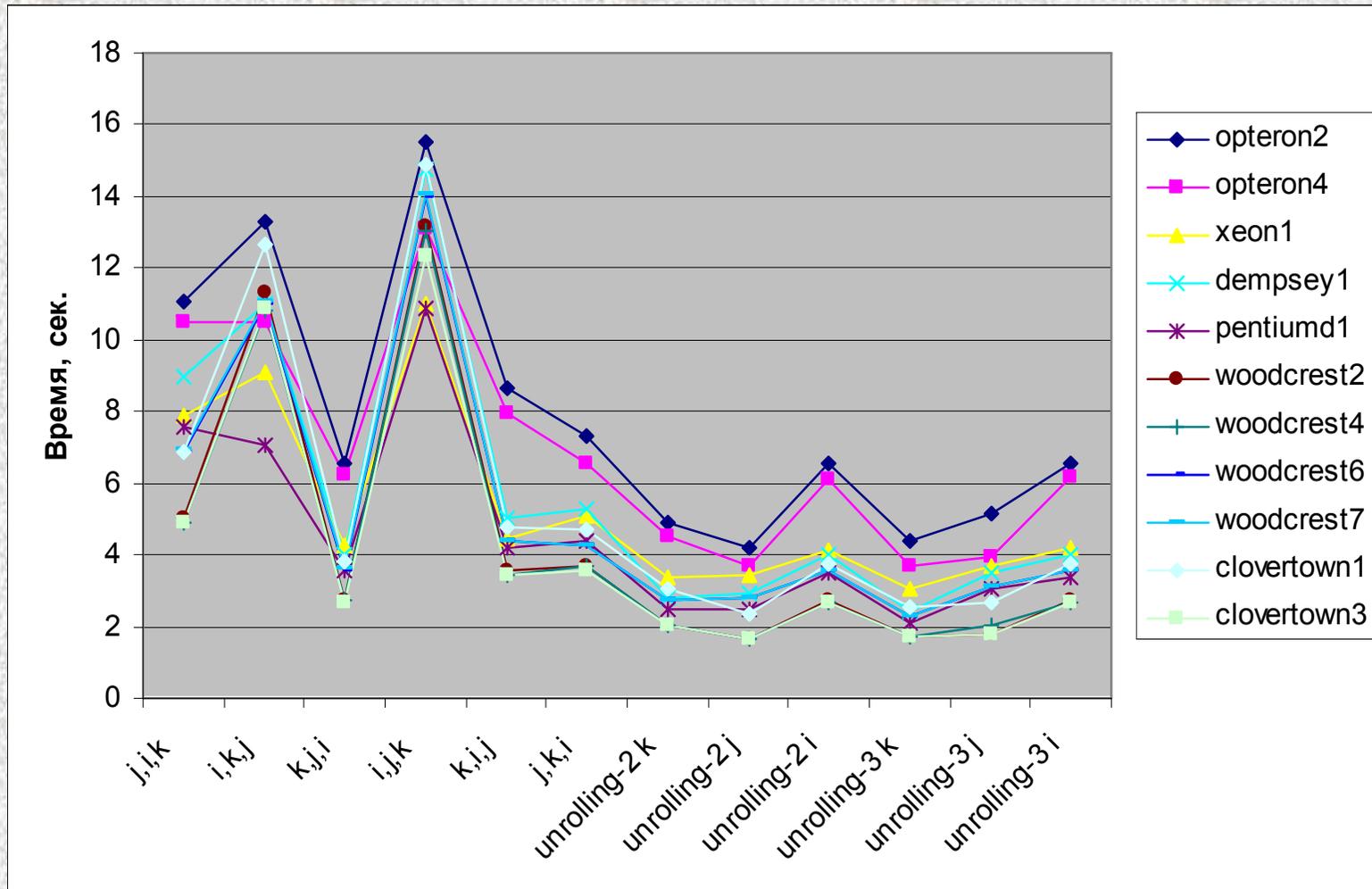
Простой пример. Исходный текст

```
for ( i = 1; i < N; i++) {  
    for ( j = 1; j < N; j++) {  
        for ( k = 1; k < N; k++) {  
            DSUM[i][k] = DSUM[i][k] + S[k] * A[k][j][i] + P[i][j] * A[k][j][i-1] +  
                P[i][k] * A[k][j-1][i] + P[j][k] * A[k-1][j][i];  
        }  
    }  
}
```

Простой пример. Эффект от преобразований (перестановка циклов, раскрытие, Intel -fast)

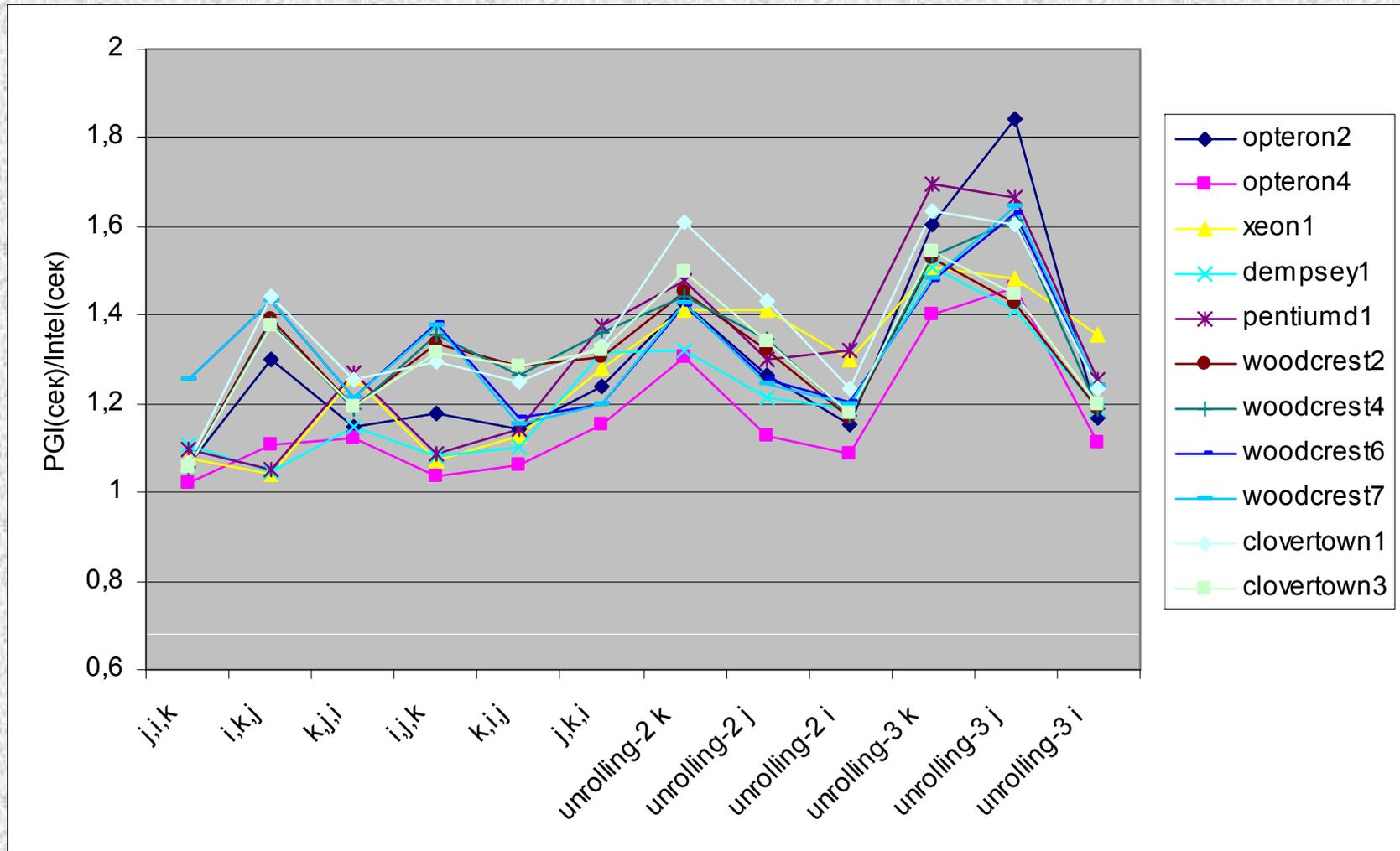


Простой пример. Эффект от преобразований (перестановка циклов, раскрытие, PGI)



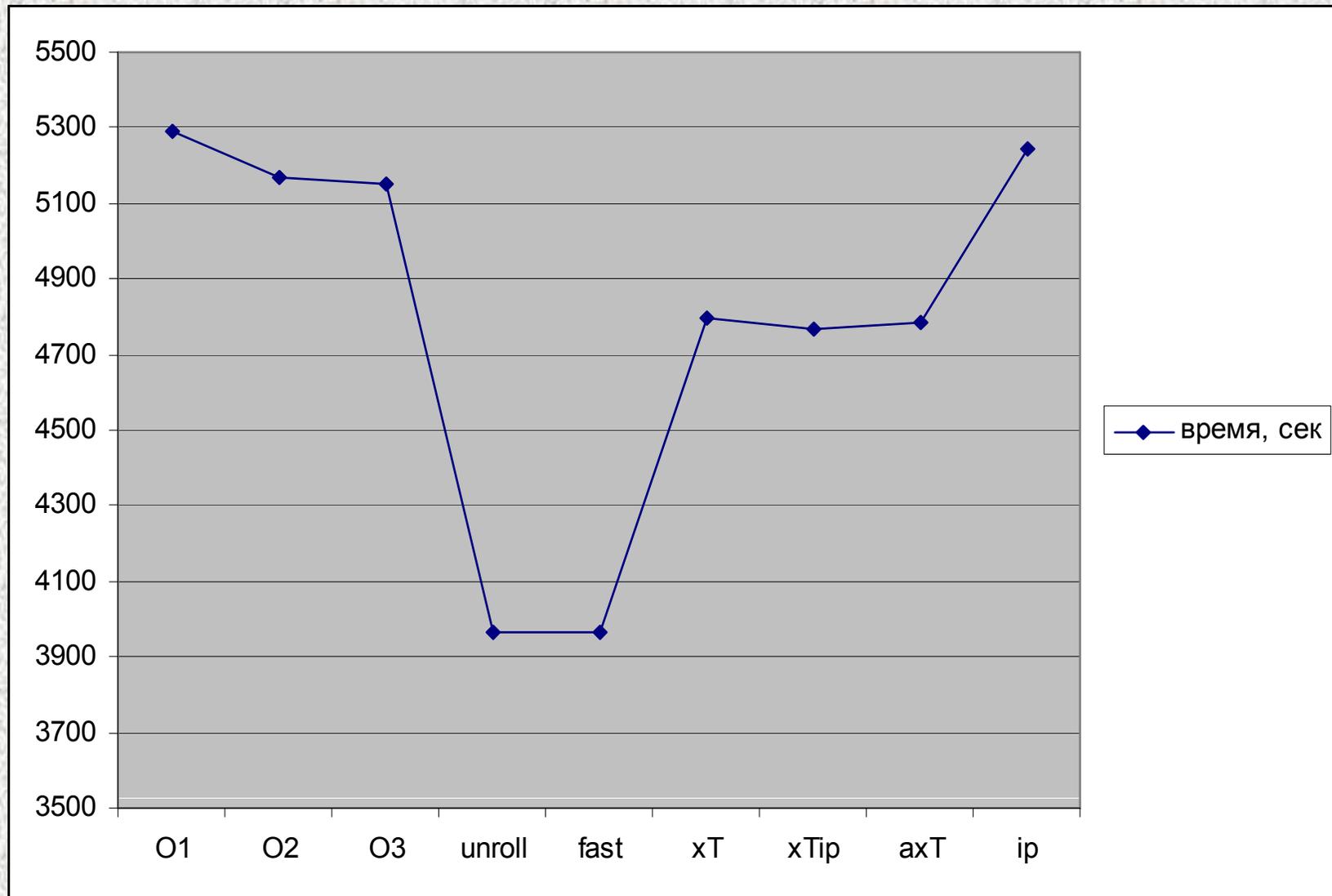
Сравнение компиляторов: Intel и PGI

(простой пример, PGI/Intel)



Выбор опций компилятора Intel

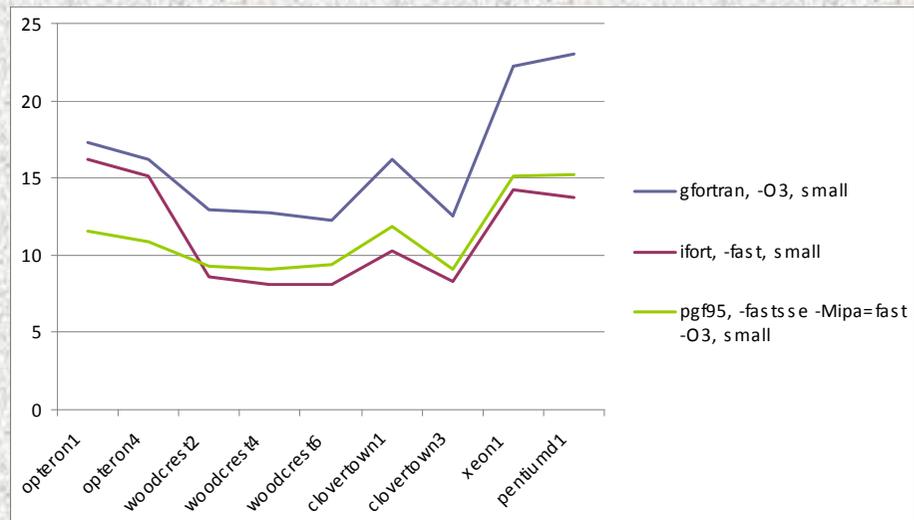
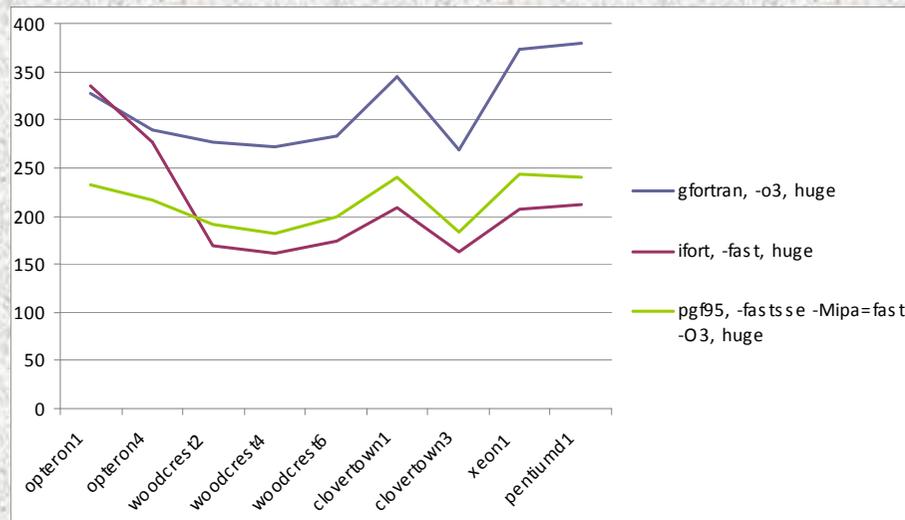
(мезомасштабная модель атмосферы)



Выбор компилятора под задачу и платформу

(компиляторы: GNU, Intel, PGI,

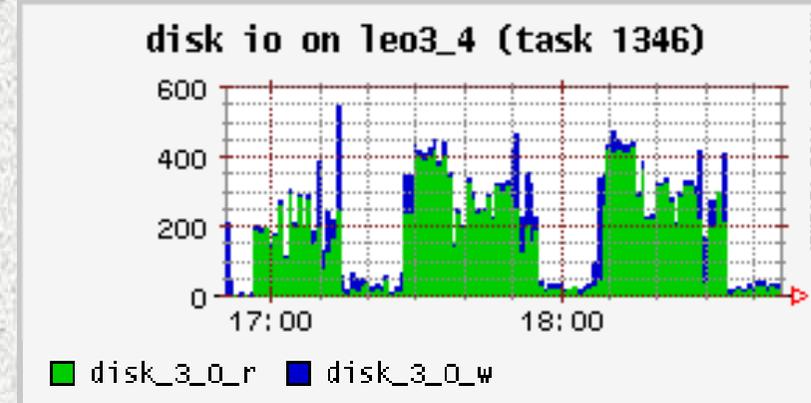
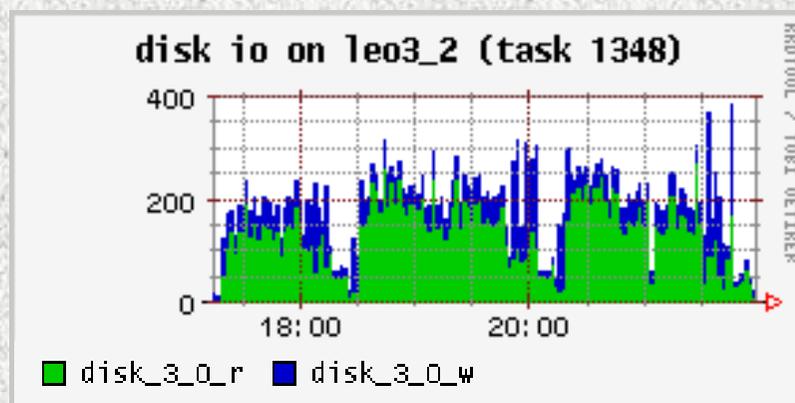
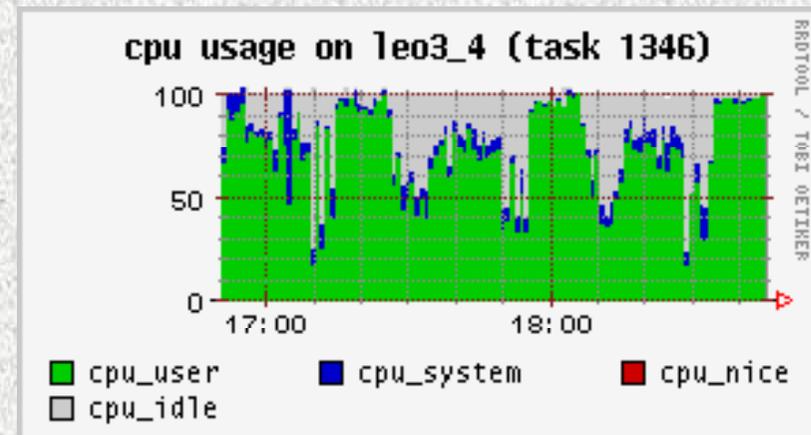
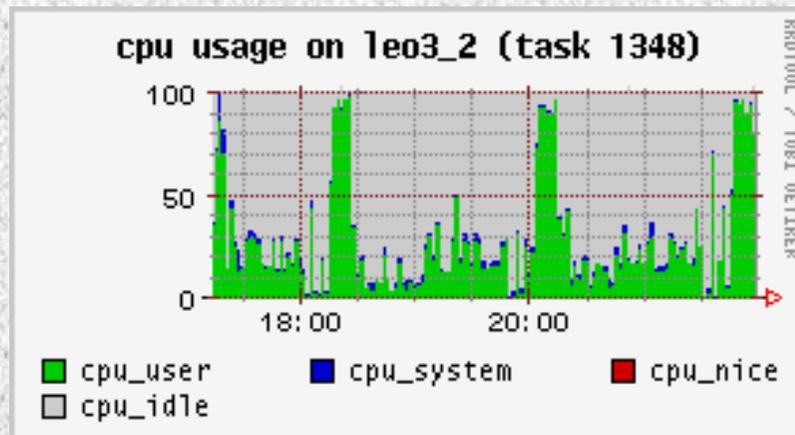
задача: вычислительная гидродинамика, Flo52)



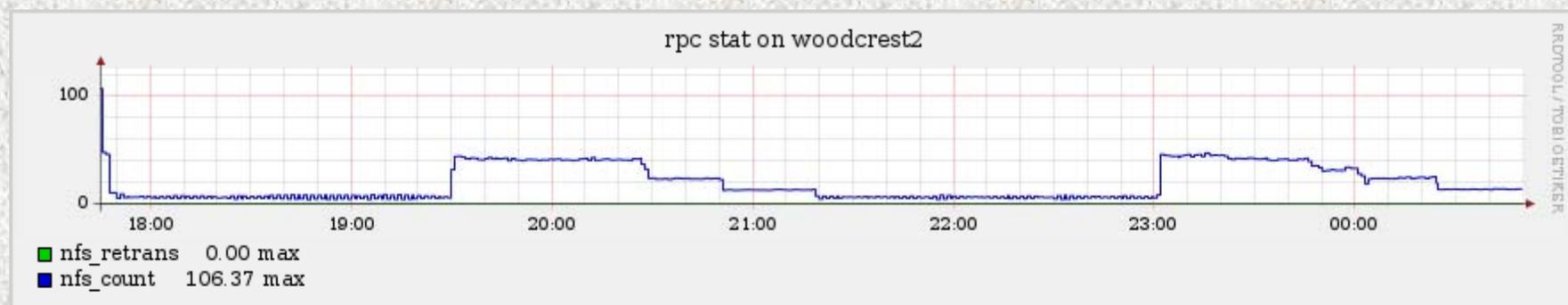
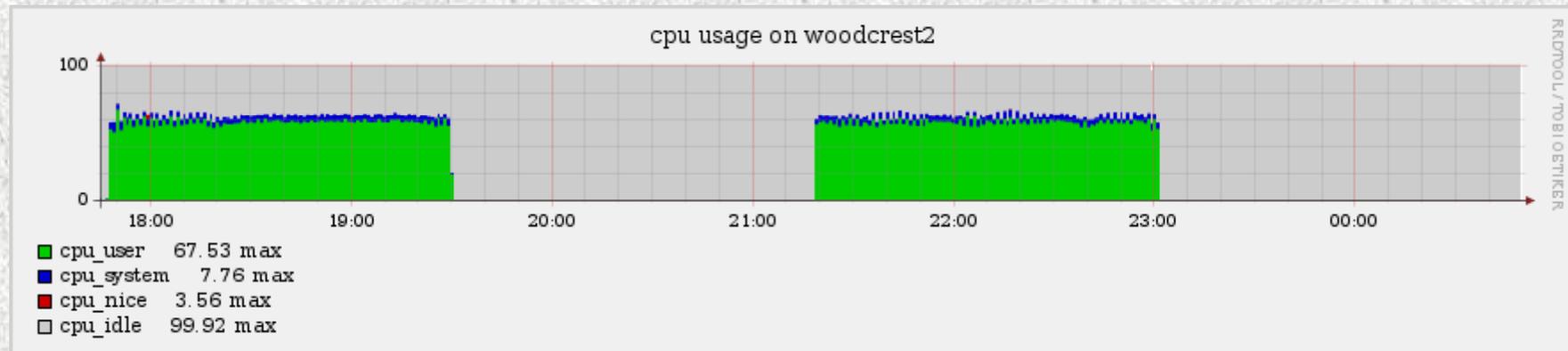
Характеристики работы программно-аппаратной среды

- *Количество задач в состоянии счёта на узле*
- *Число переключений контекста*
- *Процент использования CPU программами пользователя*
- *Процент использования CPU системой*
- *Процент использования CPU программами с приоритетом nice*
- *Процент простоя CPU*
- *Длина очереди процессов на счёт*
- *Объём памяти, занятой под системные кэши*
- *Объём памяти, свободной*
- *Объём памяти, используемой*
- *Общий объём памяти*
- *Количество принятых пакетов по сети Ethernet; Количество отправленных пакетов по сети Ethernet; Количество принятых байт по сети Ethernet; Количество отправленных байт по сети Ethernet; Количество ошибок типа carrier (отсутствие сигнала) в Ethernet; Количество ошибок типа collision (коллизия при передаче) в Ethernet; Количество ошибок типа drop (потеря пакета) в Ethernet; Количество ошибок типа err (прочие ошибки) в Ethernet; Количество ошибок типа fifo (переполнение буфера) в Ethernet; Количество ошибок типа frame (приём неверно сконструированного пакета) в Ethernet;*
- *Количество принятых блоков по NFS; Количество отправленных блоков по NFS; Число авторизаций на NFS сервере; Число операций на NFS сервере; Число перепосылок при общении с NFS сервере;*
- *Количество блоков, считанных из файла подкачки (raging); Количество блоков, записанных в файл подкачки (raging); Количество блоков, считанных из файла подкачки (swaping); Количество блоков, записанных в файл подкачки (swaping)*
- *Чтение с локального жёсткого диска; Запись на локальный жёсткий диск;*
- *Свободное место в /tmp*

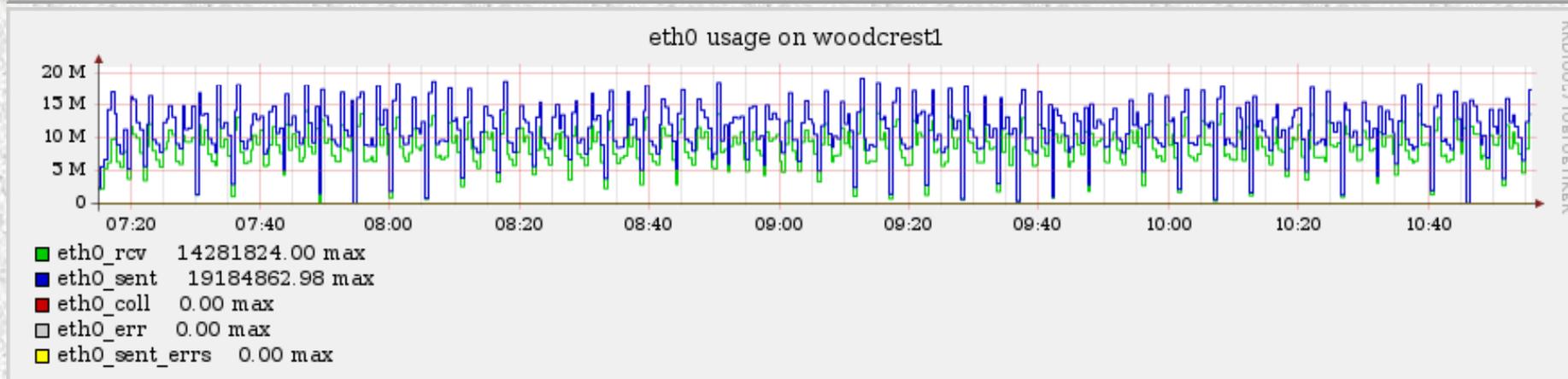
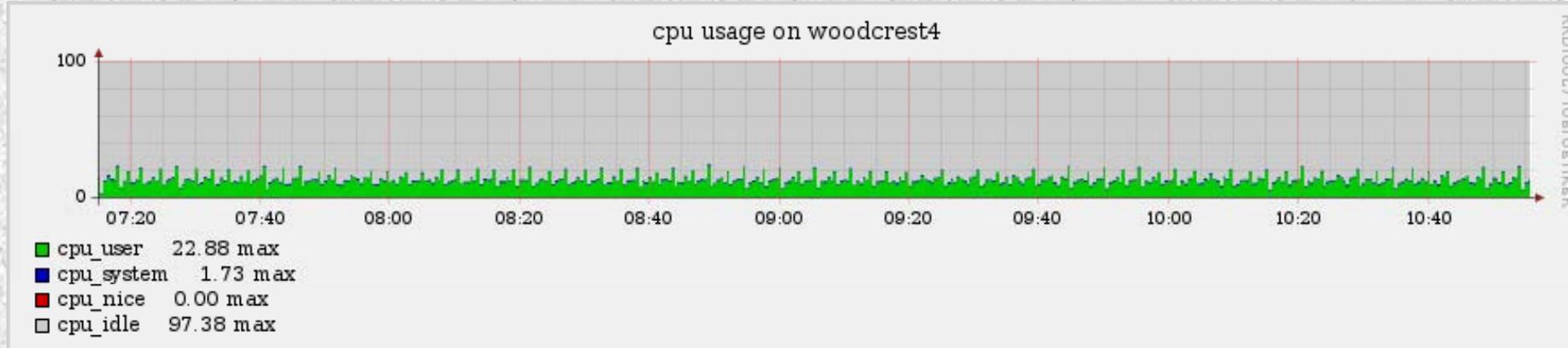
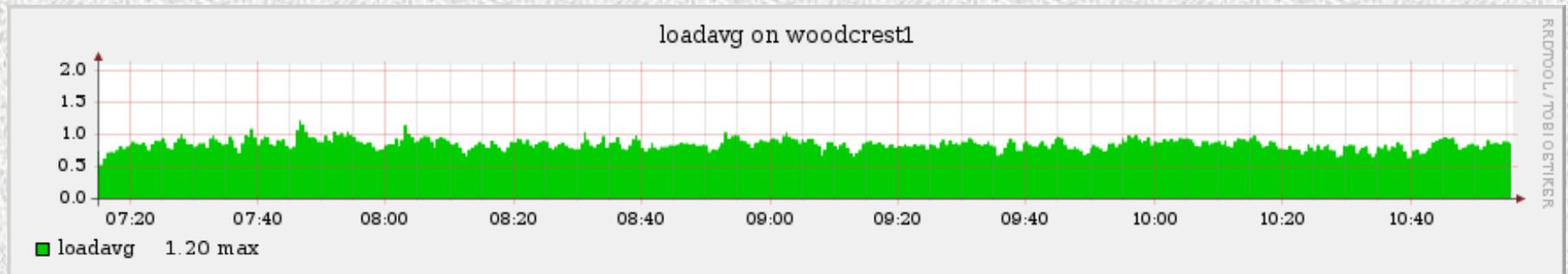
Мониторинг работы программ (использование дисков)



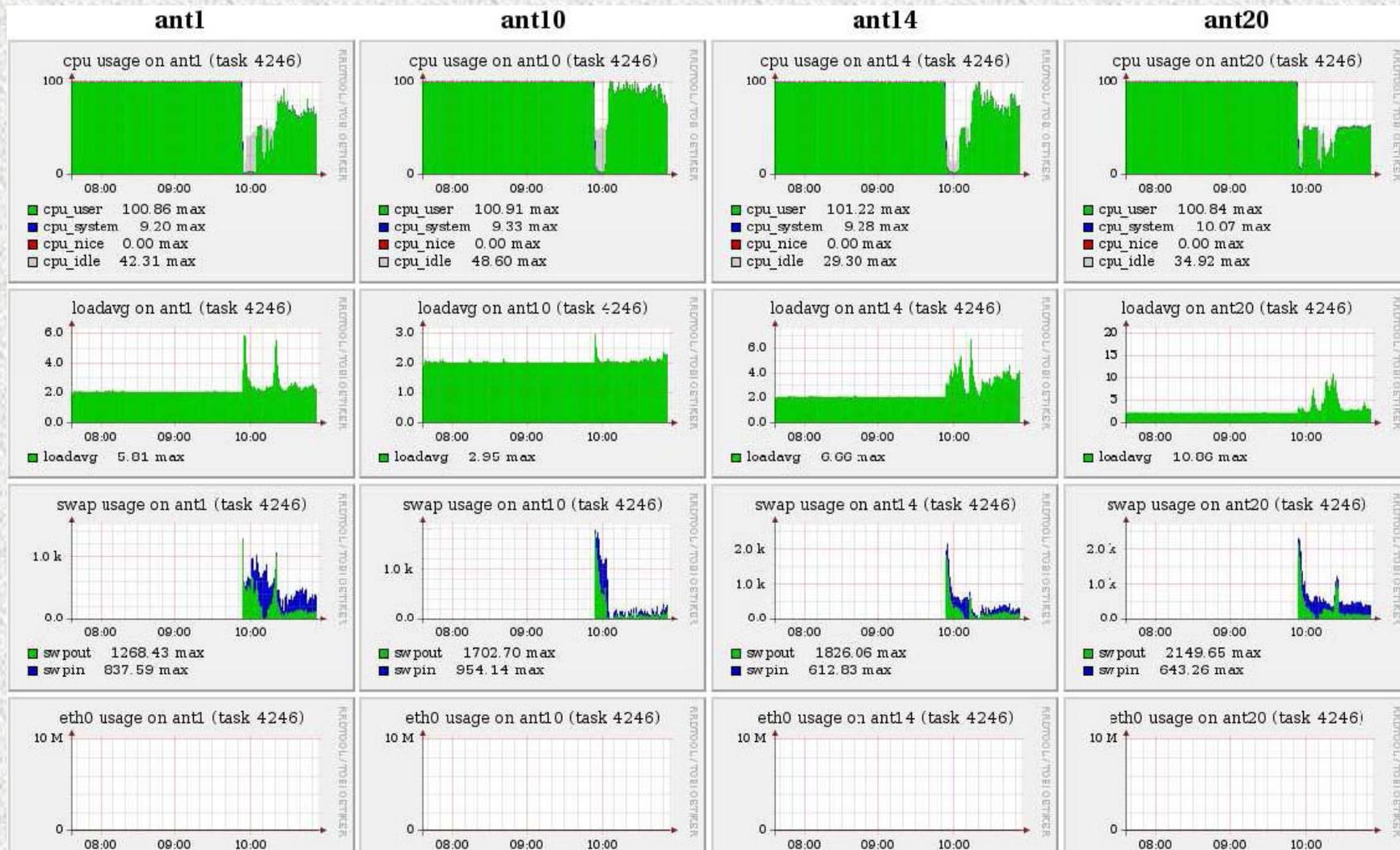
Исследование динамических свойств программ



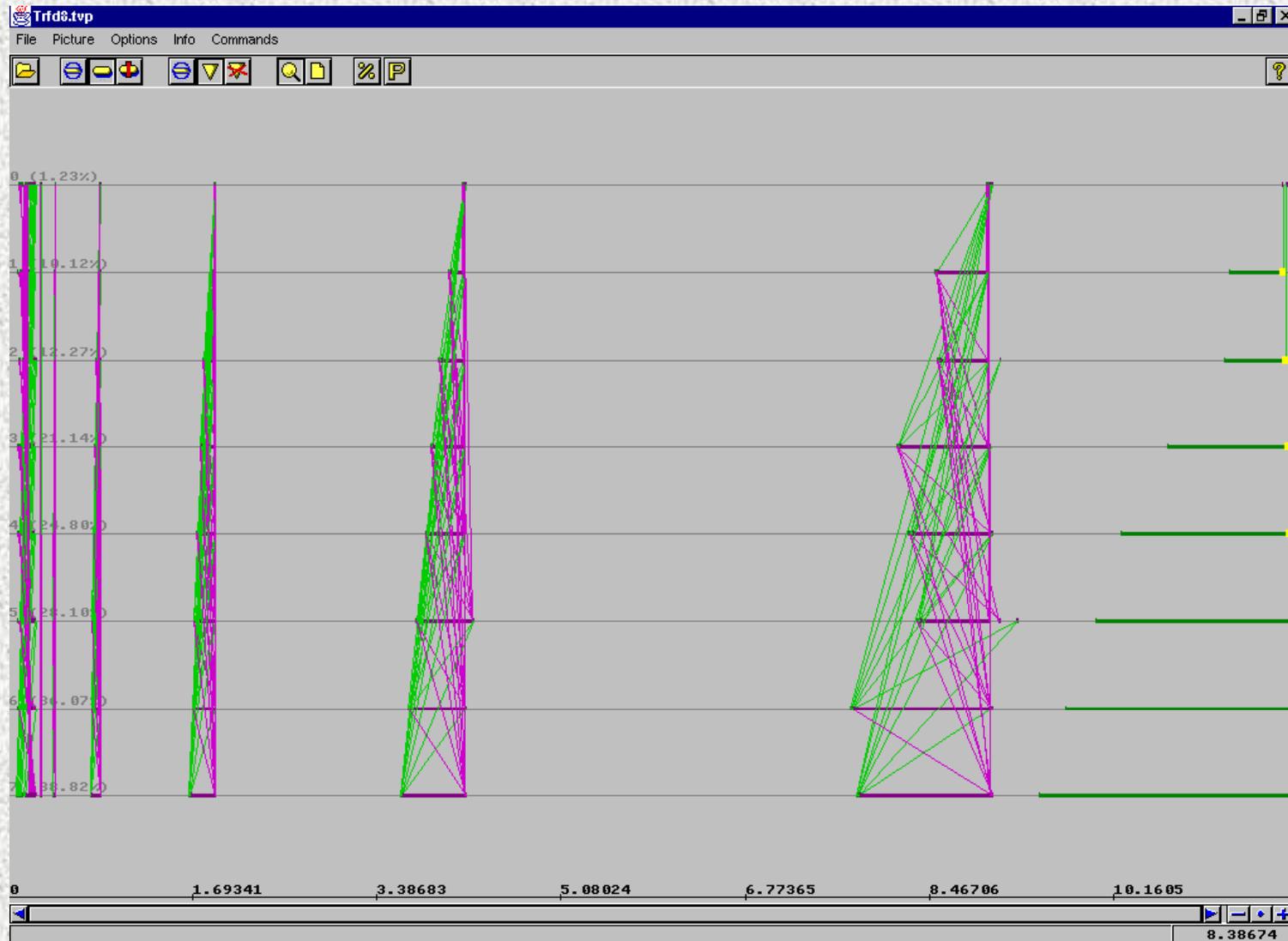
Исследование динамических свойств программ



Исследование динамических свойств программ



Исследование свойств программ



Сертификация эффективности параллельных программ

- Эффективность последовательная*
- Эффективность параллельная*

Объекты исследования:

Задача – Алгоритм – Программа – Системное ПО – Компьютер

Необходимы методика, технологии и программные инструменты сертификации эффективности и для пользователей, и для администраторов больших машин

Необходима развитая инфраструктура ПО для решения задачи отображения программ и алгоритмов на архитектуру современных вычислительных систем

Технологии эффективного программирования Intel

Компиляторы Intel (языки C/C++, Fortran77/Fortran90) поддерживают и различные уровни оптимизации для 32-х и 64-х разрядных приложений в одном пакете, и технологию параллельного программирования OpenMP, что позволяет создавать эффективные программы для современных многоядерных процессоров.

С компиляторами поставляется символьный отладчик Intel Debugger, который может работать в режимах совместимости с gdb или dbx и интегрируется с такими графическими оболочками для отладки, как ddd, Eclipse, Allinea. Отладчиком поддерживаются как многопоточные приложения OpenMP, так и написанные с использованием интерфейса native threads.

Порожденные нити автоматически попадают под контроль отладчика, причем большинство его команд можно применять либо к одной, либо ко всем нитям одновременно.



Технологии эффективного программирования Intel

***Анализатор Intel VTune** позволяет находить и устранять узкие места, препятствующие повышению производительности программы, за счет сбора, анализа и отображения данных вплоть до отдельных функций, модулей и команд. Замеряя производительность различных участков кода и помогая интерпретировать результаты замеров, VTune позволяет быстро найти участки для оптимизации. Поддерживаются многоплатформенные приложения для различных операционных систем и разных сред разработки.*



Технологии эффективного программирования Intel

Библиотека Intel Math Kernel Library (MKL) широко используется для решения вычислительно сложных задач, где от платформ Intel требуется максимальная производительность. К функциональным возможностям этой библиотеки можно отнести модули линейной алгебры (BLAS, Sparse BLAS, LAPACK и пакет Sparse Solvers), функции быстрых преобразований Фурье (FFT), векторные математические функции (VML), генераторы случайных чисел.



Технологии эффективного программирования Intel

Анализатор Intel Thread Checker позволяет упростить разработку и сопровождение многопоточных приложений. В паре с **Intel Thread Profiler** он призван решить большинство проблем, связанных с многопоточностью. Выявляет недетерминировано работающие участки кода, являющиеся причиной трудно находимых "плавающих" ошибок. Определяет ситуации неопределённости порядка записи данных, потерянные нити, блокировки, потерю сигналов, неверно отмененные блокировки. Поддерживает стандарты OpenMP, POSIX и Windows API. Вообще говоря, Intel Thread Checker не зависит от компилятора, способен работать с любым исполняемым файлом, но в сочетании с компилятором Intel проводит более глубокий анализ.



Технологии эффективного программирования Intel

Инструменты для программирования кластеров объединены в **пакет Intel Cluster Tools**. Сюда входит библиотека Intel MPI, оптимизированная параллельная математическая библиотека Intel Cluster MKL и специальный инструмент **Intel Trace Analyzer & Collector**, предназначенный для создания эффективных масштабируемых параллельных программ.



Технологии эффективного программирования Intel

Хорошим вариантом тестирования производительности системы является использование пакета **Intel MPI Benchmarks**, ранее известного под названием *Pallas MPI Benchmarks*. Пакет измеряет базовые характеристики кластерной системы, а также показывает эффективность основных функций MPI, предназначенных для реализации коллективных операций и передачи сообщений между двумя процессами, односторонних операций, функций для выполнения операций ввода/вывода и ряда других.



Технологии эффективного программирования Intel

Хорошим вариантом тестирования производительности системы является использование пакета **Intel MPI Benchmarks**, ранее известного под названием *Pallas MPI Benchmarks*. Пакет измеряет базовые характеристики кластерной системы, а также показывает эффективность основных функций MPI, предназначенных для реализации коллективных операций и передачи сообщений между двумя процессами, односторонних операций, функций для выполнения операций ввода/вывода и ряда других.



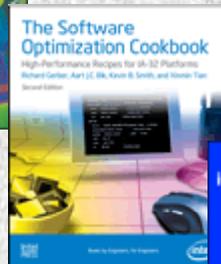
HPC-издания Intel Press



Programming with Hyper-Threading Technology ...



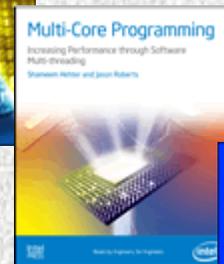
Scientific Computing on Itanium®-based Systems...



The Software Optimization Cookbook, Second Edition...



The Software Vectorization Handbook...



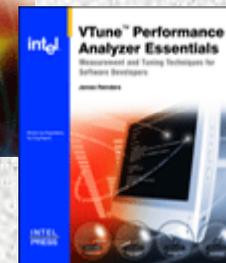
Multi-Core Programming...



Intel® Integrated Performance Primitives...



Programming with Intel® Extended Memory 64 Technology...



VTune™ Performance Analyzer Essentials

Алгоритмическая эффективность

Задача о числе счастливых билетиков :

```
count = 0;
for ( i1 = 0; i1 < 10; ++i1)
  for ( i2 = 0; i2 < 10; ++i2)
    for ( i3 = 0; i3 < 10; ++i3)
      for ( i4 = 0; i4 < 10; ++i4)
        for ( i5 = 0; i5 < 10; ++i5)
          for ( i6 = 0; i6 < 10; ++i6) {
            if ( i1+i2+i3+i4+i5+i6 == 6 )
              count++;
          }
}
```

Intel Core Duo 2.6 ГГц:

8 цифр – 0.1 сек

10 цифр – 10 сек

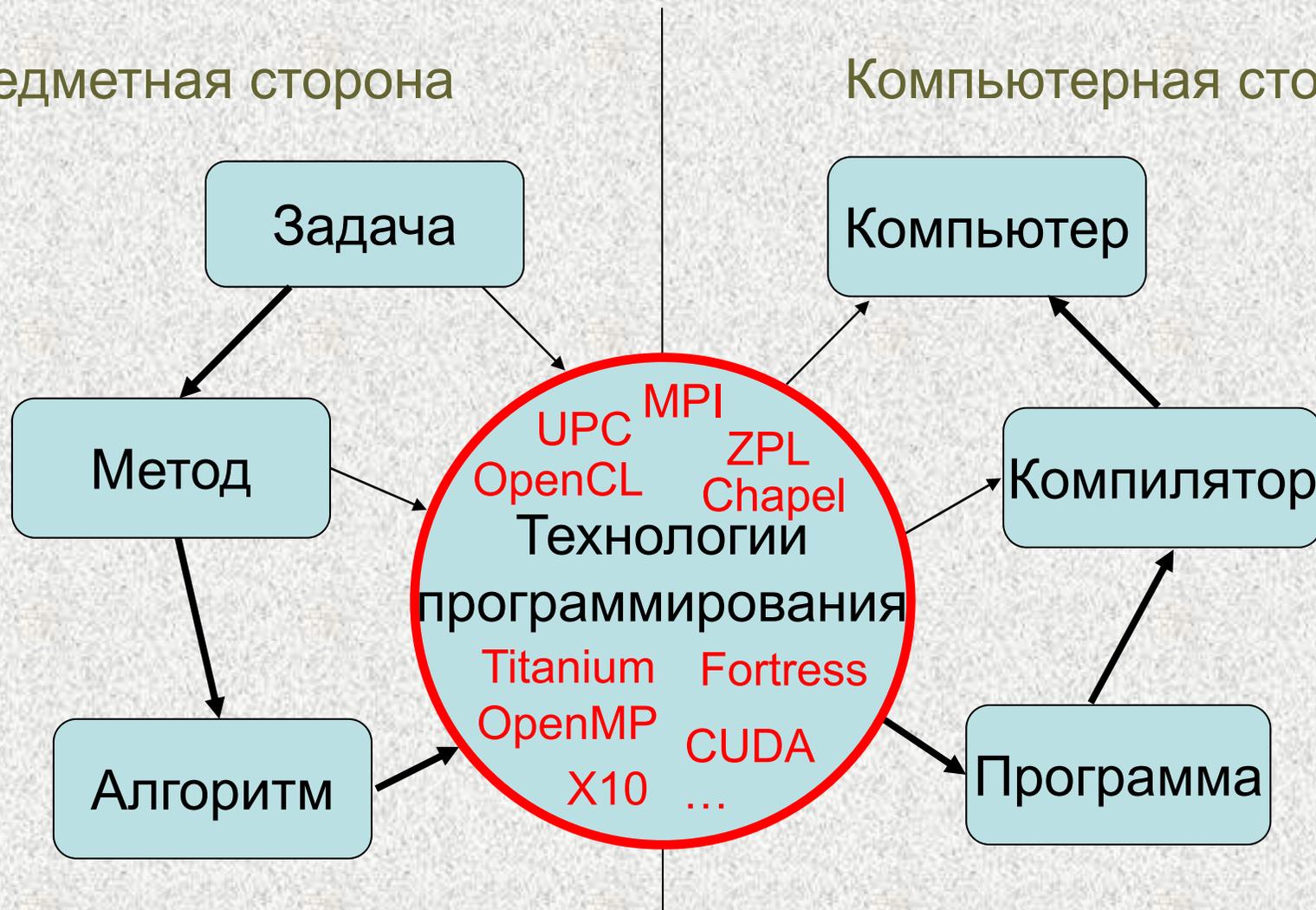
12 цифр – 30 мин



Решение задачи на компьютере

Предметная сторона

Компьютерная сторона



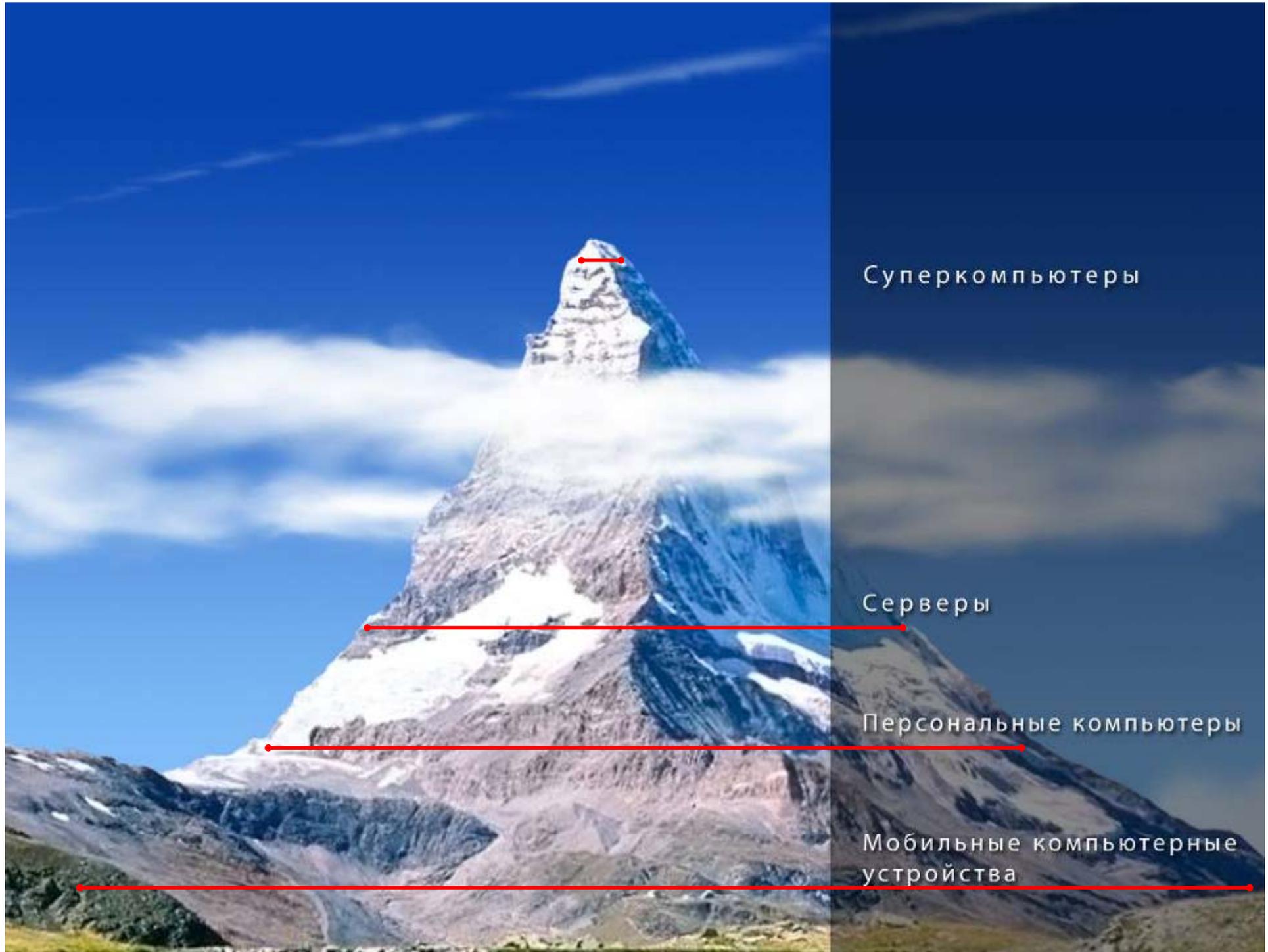
Эффективность, переносимость, продуктивность

Эффективность? HPC-образование...

- *Вычислительное ядро: $A = B+C$*
- *Производительность: ' X ' Mflops на Cray T90
(например)*
- *Пользователь: Я хочу больше, чем 'X'...*

Эффективность? HPC-образование...

- *Вычислительное ядро: $A = B + C * 1$*
- *Производительность: '2*X' Mflops на Cray T90 :))*
А время осталось неизменным!!! :((
- *Пользователь: Я хочу больше, чем 'X'...*
- *System Guru: Хочешь больше Mflops... Нет проблем...*



Суперкомпьютеры

Серверы

Персональные компьютеры

Мобильные компьютерные
устройства



Суперкомпьютеры

Серверы

Персональные компьютеры

Мобильные компьютерные устройства

*Всероссийская молодежная школа
“Суперкомпьютерные технологии и высокопроизводительные
вычисления в образовании, науке и промышленности”*

Суперкомпьютеры и КПД паровоза

*Вл.В.Воеводин
Заместитель директора НИВЦ МГУ,
член-корреспондент РАН,
voevodin@parallel.ru*

ННГУ – 31 октября 2009 г.