

Лекция 1. Организация функционирования распределённых вычислительных систем

Перышкова Евгения Николаевна

к.т.н. доцент Кафедры вычислительных систем

Сибирский государственный университет

телекоммуникаций и информатики

e-mail: e.peryshkova@gmail.com

О курсе «Архитектура вычислительных систем»

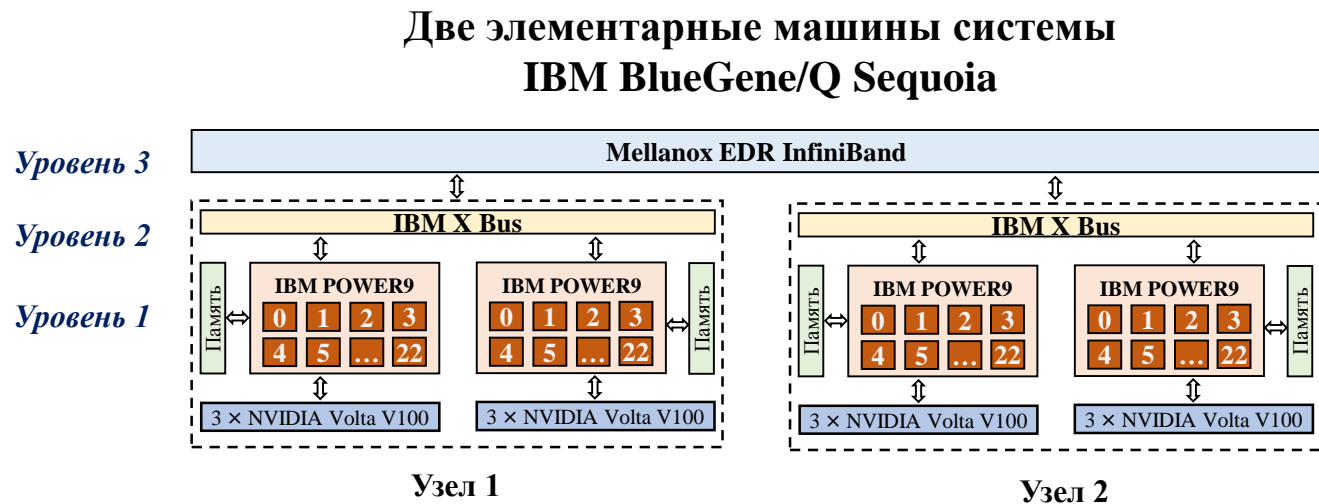
- Лекции – каждую неделю
- Лабораторные работы – 4 работы
- Контрольная работа
- Итоговая аттестация: зачет

Условия получения зачета:

- Оценка посещаемости и успеваемости
- Сдача лабораторных работ в срок
- Сдача контрольной работы на удовлетворительно

АКТУАЛЬНОСТЬ

- **Вычислительная система (ВС)** – совокупность множества элементарных машин (ЭМ) и коммуникационной сети, связывающих их
- **Архитектурные свойства современных ВС [1]:**
 - *мультиархитектура* вычислительных узлов
 - *иерархическая организация* коммуникационной среды
 - *большемасштабность*



[1] Хорошевский В.Г. *Распределенные вычислительные системы с программируемой структурой* // Вестник СибГУТИ – 2010. – №2. – С. 3-41.

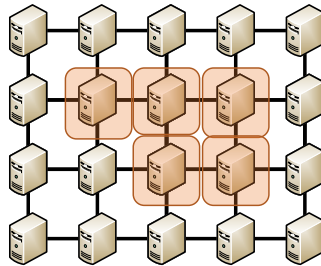
РЕЖИМЫ ФУНКЦИОНИРОВАНИЯ ВС

Мультипрограммные режимы функционирования ВС

Организация одновременного решения множества задач на ресурсах ВС

Режим обслуживания потока задач

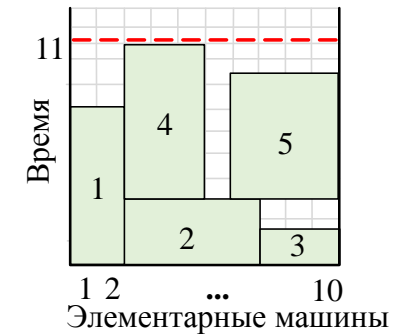
Формирование (суб)оптимальных подсистем элементарных машин для параллельных задач



- Техника теории игр, стохастическое программирование (Хорошевский, 1973), (Юдин, 1974), (Ермольев, 1976)
- Алгоритмы на графах (Корнеев, Монахов, 1985), (Livingston, 2002)

Режим обработки набора задач

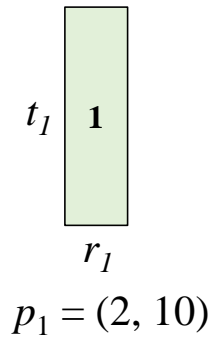
Построение расписаний решения параллельных задач (task scheduling)



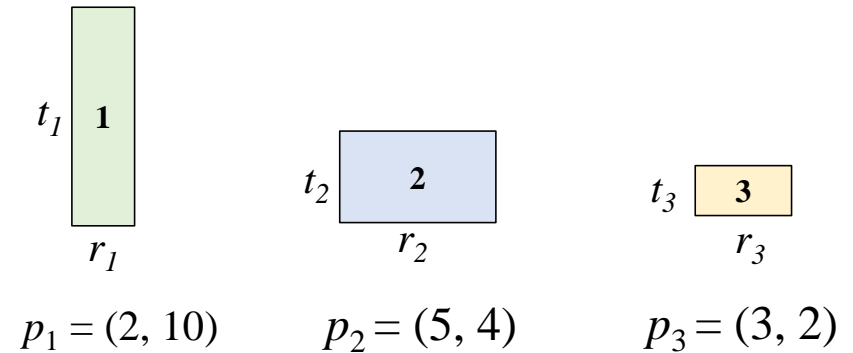
- Точные методы: перебор, метод ветвей и границ
- Сведение к задаче упаковки объектов в контейнеры: 1DBPP, 2DSPP
- Алгоритмы локального поиска: генетические, имитация отжига, поиск с запретами, роевые алгоритмы

РЕЖИМ ОБРАБОТКИ ЗАДАЧ С НЕФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

Задача с *фиксированными* параметрами (rigid)
(1 вариант подсистемы ЭМ):



Задача с *нефиксированными* параметрами (moldable)
(3 варианта подсистем ЭМ):



Преимущества поддержки в системах управления ресурсами параллельных задач с нефиксированными параметрами:

- Сокращение суммарного времени решения задач
- Выполнение технико-экономических ограничений
 - ☐ Восстановление вычислительного процесса на допустимой подсистеме меньшего ранга
 - ☐ Лицензионные ограничения программ на размеры подсистемы

Актуальным является
создание алгоритмов
обработки наборов задач
с нефиксированными
параметрами

ПЛАНИРОВАНИЕ РЕШЕНИЯ ПАРАЛЛЕЛЬНЫХ ЗАДАЧ С НЕФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

ОБРАБОТКА НАБОРА ЗАДАЧ С ФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

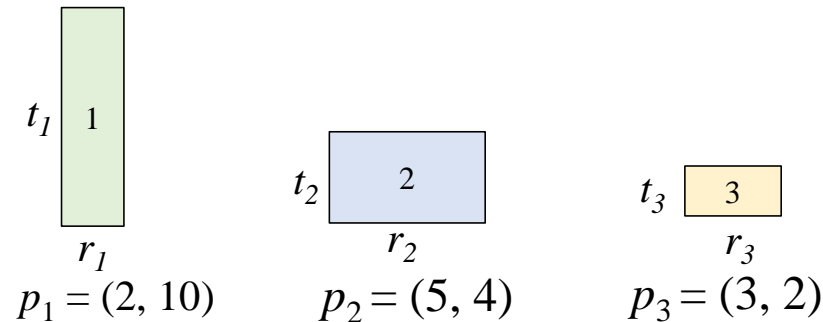
- Задан набор из t задач с *фиксированными* (rigid) рангами r_i и временем t_i решения
- Требуется построить расписание решения на ВС из n ЭМ задач набора – определить для каждой задачи подсистему ЭМ и момент запуска параллельных ветвей на ней
- Для обработки наборов задач с фиксированными параметрами разработаны эффективные методы и алгоритмы:
 - Сведение к задачи одномерной упаковки 1DBPP (В.Г. Хорошевский, 1967), (Поспелов, 1972)
 - Сведение к задаче двумерной упаковки 2DSPP (Coffman, 1980)
 - Стохастические алгоритмы локального поиска (Мухачева, 2001)
 - Метод ветвей и границ (Сидельников, 2006)
 - Детерминированные алгоритмы с гарантированной оценкой точности (Ntene, 2009)
 - Алгоритмы решения задачи календарного планирования (Гимади, 2001), (Кочетов, 2000)

ОБРАБОТКА НАБОРА ЗАДАЧ С НЕФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

- **Задан** набор из t задач с *нефиксированными* (moldable) рангами r_i и временем t_i решения
- **Требуется** построить расписание решения на ВС из n ЭМ задач набора – определить для каждой задачи подсистему ЭМ и момент запуска параллельных ветвей на ней
- Задача с нефиксированными параметрами (moldable job) представлена вектором p_i из q_i различных вариантов параметров задачи:

$$p_i = (p_i^1, p_i^2, \dots, p_i^{q_i})$$

Задача с нефиксированными параметрами ($q = 3$)



[1] Sabin, G. *Moldable parallel job scheduling using job efficiency: an iterative approach*, 2007.

[2] Khandekar, R. *Real-time scheduling to minimize machine busy times*, 2010.

[3] Huang, K-C. *Online Scheduling of Moldable Jobs with Deadline*, 2015.

ЗАДАЧА С НЕФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

- Современные технико-экономические ограничения:
 - Некоторые параллельные задачи допускают реализацию только на подсистемах ЭМ с определенным свойствами, например, с числом процессорных ядер равным степени числа два
 - Лицензии значительной части коммерческих пакетов параллельного моделирования допускают их запуск только на фиксированных конфигурациях подсистем ЭМ
 - Отказы ресурсов в большемасштабных ВС и перспективных системах эксафлопсной производительности требуют поддержки возможности восстановления вычислительного процесса на допустимой подсистеме меньшего ранга
- Известные методы построения расписания решения задач не применимы для задач с нефиксированными параметрами, поэтому актуальным является создание новых алгоритмов формирования расписания решения задач с нефиксированными параметрами

ОБРАБОТКА НАБОРА ЗАДАЧ С НЕФИКСИРОВАННЫМИ ПАРАМЕТРАМИ

- Требуется построить расписание S решения задач на ВС

$$S = ((s_1, x_1, k_1), (s_2, x_2, k_2), \dots, (s_m, x_m, k_m))$$

- s_i – время начала решения i -ой задачи на ВС
- $x_i = (x_{i1}, x_{i2}, \dots, x_{ir_i^{k_i}})$ – подсистема ЭМ для выполнения ветвей программы, x_{ij} – номер ЭМ для выполнения ветви j задачи i

$$T(S) = \max_{i \in J} \{s_i + t_i^{k_i}\} \rightarrow \min_{S \in \Omega}$$

при ограничениях:

$$\sum_{i \in J(t)} r_i^{k_i} \leq n, \quad \forall t \in \mathbb{R},$$

$$\prod_{i \in J(t)} \prod_{i' \in J(t) \setminus \{i\}} (x_{ij} - x_{i'j'}), \quad \forall t \in \mathbb{R}, \quad j = 1, 2, \dots, r_i^{k_i}, j' = 1, 2, \dots, r_{i'}^{k_{i'}},$$

$$\frac{1}{m} \sum_{i=1}^m \frac{w_i^{k_i}}{\max_{k=1, q_i} w_i^k} \geq w.$$

$$\begin{aligned} x_{ij} &\in C, & i &= 1, 2, \dots, m, & j &= 1, 2, \dots, r_i^{k_i}, \\ s_i &\in \mathbb{R}, & k_i &\in \{1, 2, \dots, q_i\}, & i &= 1, 2, \dots, m. \end{aligned}$$

ПОСЛЕДОВАТЕЛЬНЫЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Генетический алгоритм формирования расписания решения задач с нефиксированными параметрами

- **Шаг 1.** Создание популяции из K допустимых расписаний. Расписание формируется с помощью эвристических алгоритмов FFDH или BFDH
- **Шаг 2.** Получение новых особей путем «скрещивания» пары расписаний – два расписания делятся на G частей и переставляются. Если особь не скрещивается, к ней применяется оператор мутации
- **Шаг 3.** Упорядочивание всей популяции по значению целевой функции $T(S)$. В популяции остаются лучшие K особей (расписаний)
- **Шаг 4.** Если количество эволюционных циклов не достигло предельного значения V , то возврат к шагу 2
- **Шаг 5.** За итоговое решение принимается особь с экстремальным значением целевой функции $T(S)$ в текущей популяции

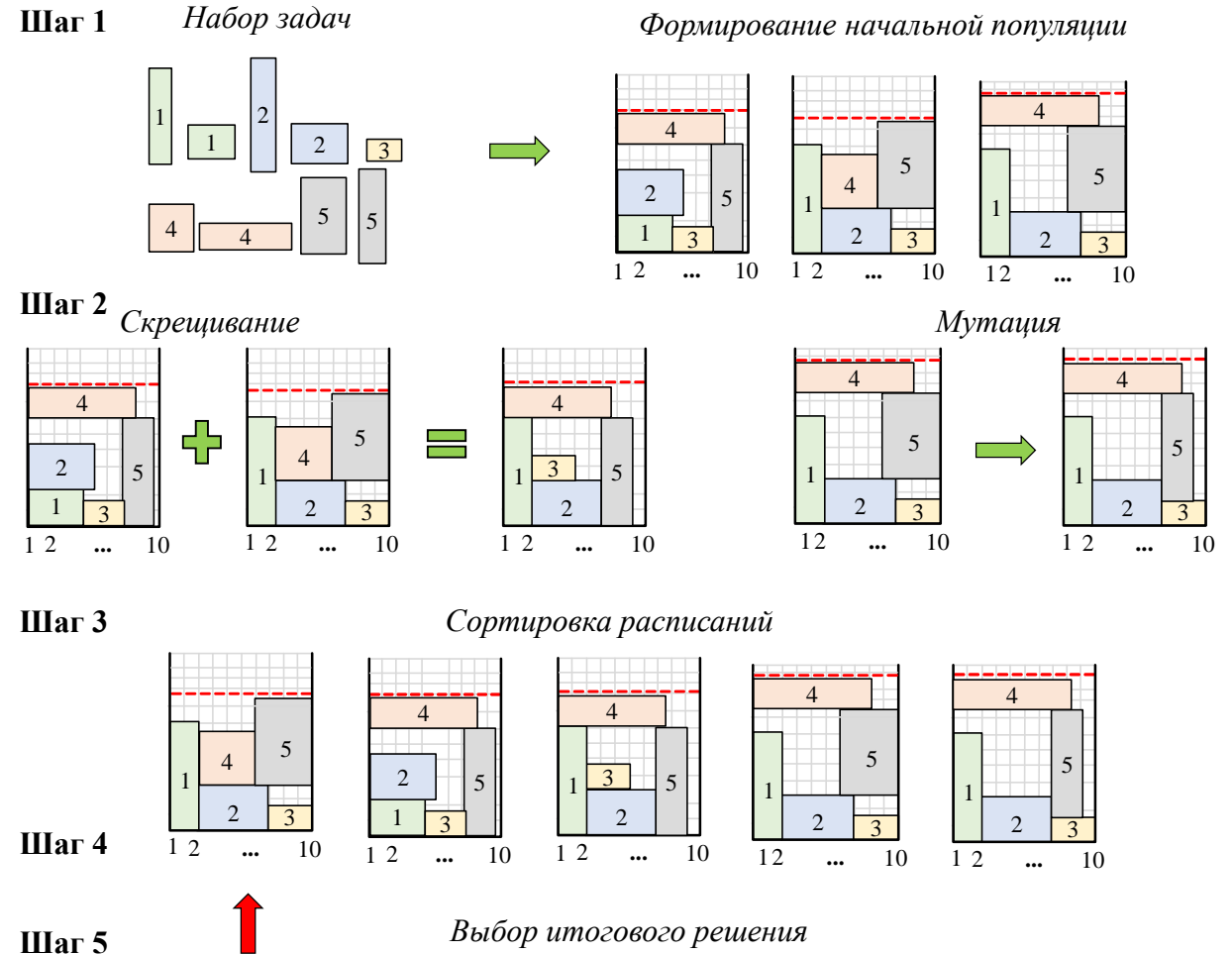
Вычислительная сложность алгоритма равна:

$$T = O(K \cdot m + K \cdot T_{2DSPP} + V \cdot K \log K + V \cdot K),$$

где T_{2DSPP} – время работы алгоритмов упаковки,

K – размер популяции,

V – количество эволюционных циклов.



ИССЛЕДОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

- Алгоритм реализован на C++ в пакете MOJOS
- Проведено моделирование алгоритма на тестовых наборах задач для моделей ВС с числом ЭМ $n = 1024, 4096, 16384, 65536$ и количеством задач в наборе $m = 1000, 2000$ и 3000
- Показатели эффективности алгоритма:

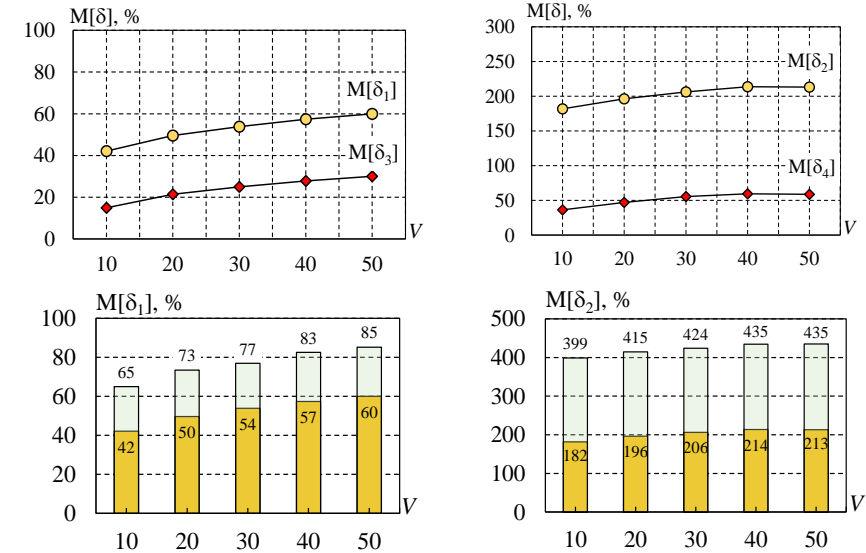
$$\delta_1 = (T_{BFDH} - T)/T, \quad \delta_2 = (T_{FFDH} - T)/T,$$

$$\delta_3 = (T_{BFDH_INIT} - T)/T, \quad \delta_4 = (T_{FFDH_INIT} - T)/T$$

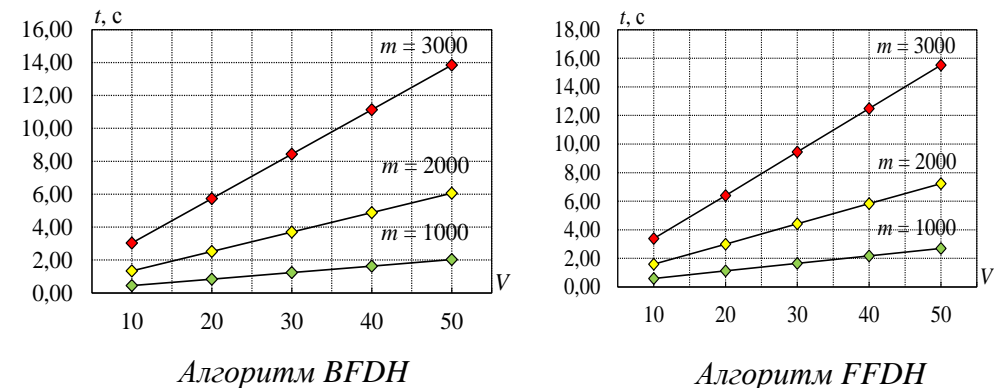
- T_{BFDH_INIT} – значение целевой функции от начального решения, полученного алгоритмом BFDH
- T_{FFDH_INIT} – значение целевой функции от начального решения, полученного алгоритмом FFDH
- T – значение целевой функции от решения, полученного генетическим алгоритмом
- T_{BFDH} и T_{FFDH} – значение целевой функции от решений, полученных алгоритмами BFDH и FFDH

Генетический алгоритм построения расписаний решения задач с нефиксированными параметрами обеспечивает **сокращение суммарного времени выполнения задач в среднем на 45%** относительно начальных решений, получаемых известными алгоритмами FFDH и BFDH

Зависимость математического ожидания и среднеквадратического отклонения от количества эволюционных циклов V
($n = 1024, m = 1000; K = 16, w = 75 \%, P = 90 \%, G = 2$)



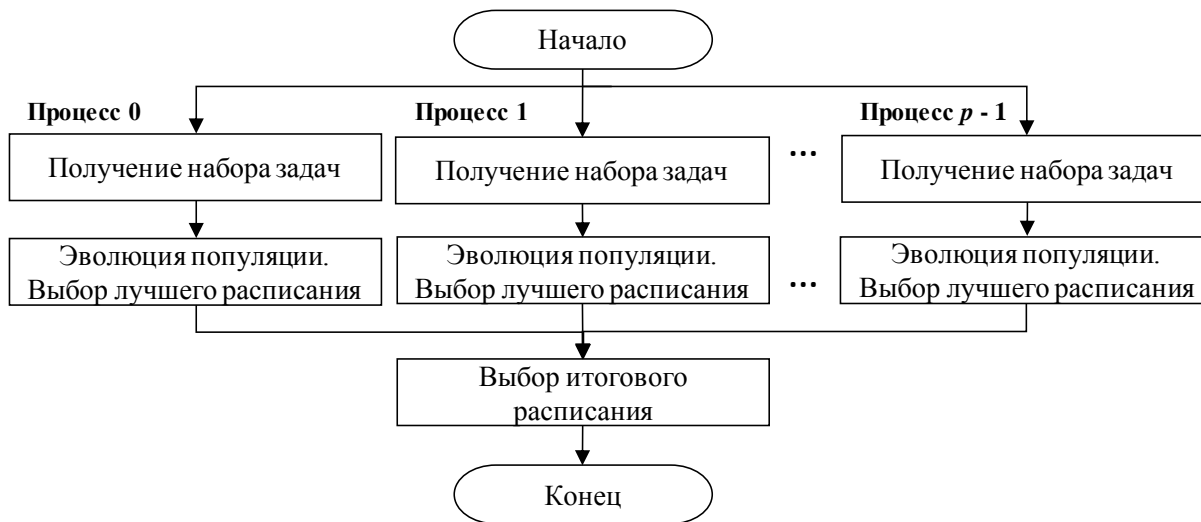
Зависимость времени работы алгоритма от количества эволюционных циклов V и числа m задач в наборе (1024 ЭМ)



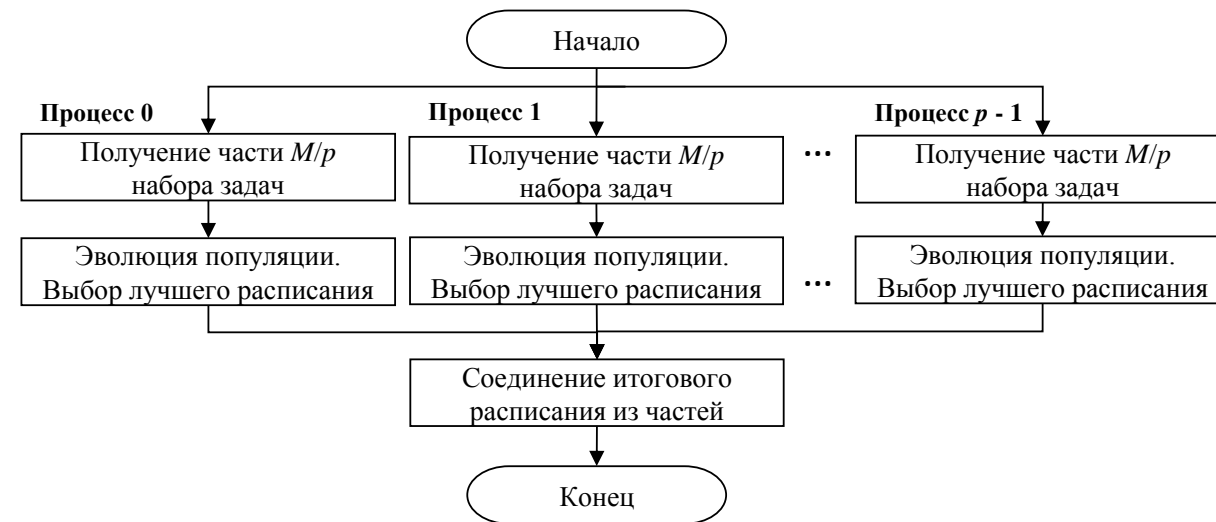
ПАРАЛЛЕЛЬНЫЕ ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Параллельные генетические алгоритмы формирования расписания решения задач с нефиксированными параметрами

Алгоритм на основе метода мультистарта



Алгоритм на основе декомпозиции набора задач



M – количество задач в наборе, p – количество процессов программы

- Алгоритмы реализованы на языке C в модели передачи сообщений MPI
- Характеризуются линейной зависимостью ускорения от числа процессов

ИНТЕГРАЦИЯ В СИСТЕМУ УПРАВЛЕНИЯ РЕСУРСАМИ TORQUE И ПЛАНИРОВЩИК MAUI

- Выполнена интеграция пакета MOJOS с системой TORQUE и планировщик Maui
- Язык запросов расширен новой структурой для описания вектора допустимых подсистем ЭМ

`-L nodes = value @ ppn = value @
weight = value @ walltime = value,`

- Эксперименты на вычислительном кластере: 18 узлов, сеть Gigabit Ethernet; наборы задач: $m = 100, 200, 400$ и 800
- Показатели эффективности:

$$\delta_1 = (T_T - T)/T, \quad \delta_2 = (T_M - T)/T,$$

$$\delta_3 = (T_{TQ} - T_Q)/T_Q, \quad \delta_4 = (T_{MQ} - T_Q)/T_Q$$

- T_T и T_{TQ} – время решения задач и время ожидания задач в очереди, обеспечиваемое системой TORQUE
- T_M и T_{MQ} – время решения задач и время ожидания задач в очереди, обеспечиваемое планировщиком Maui
- T и T_Q – время решения задач и время ожидания задач в очереди, обеспечиваемое разработанными автором средствами

Количество задач в наборе	T_T , с	T_M , с	T , с	$M[\delta_1]$, %	$M[\delta_2]$, %
100	13403	12317	11188	19,80	10,09
200	50426	45171	43195	16,74	4,57
400	207528	202364	175033	18,57	15,61
800	836394	796453	713580	17,21	11,61

Количество задач в наборе	T_{TQ} , с	T_{MQ} , с	T , с	$M[\delta_3]$, %	$M[\delta_4]$, %
100	323	310	246	31,30	26,02
200	372	356	326	14,11	9,20
400	684	697	605	13,06	15,21
800	1324	1294	1200	10,33	7,83

В среднем применение разработанных средств для системы TORQUE на рассмотренных наборах **позволяет сократить суммарное время решения задач на 24 % и на 21 %** для планировщика Maui

АЛГОРИТМЫ ФОРМИРОВАНИЯ ПОДСИСТЕМ ЭЛЕМЕНТАРНЫХ МАШИНА

АЛГОРИТМЫ ФОРМИРОВАНИЯ ПОДСИСТЕМ ЭЛЕМЕНТАРНЫХ МАШИН

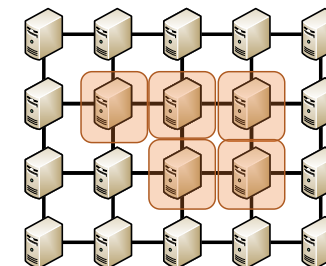
Заданы распределенная ВС и количество M ветвей в параллельной программе.

Требуется сформировать подсистему, обеспечивающую эффективную реализацию коллективных схем межмашинных обменов.

Обозначения:

- l_{pq} – кратчайшее расстояние между ЭМ p и q в структуре ВС.
- b_{pq} – пропускная способность канала связи между ЭМ p и q .

Распределенная ВС
 $N = 20, M = 5$



ВС с однородной структурой сети

$$L(X) = \left(\prod_{p=1}^{n-1} \prod_{q=p+1}^n (x_p x_q (l_{pq} - 1) + 1) \right) \rightarrow \min_{(x_p)}$$

при ограничениях:

$$\sum_{p=1}^n x_p = M,$$

$$x_p \in \{0, 1\}, \quad p = 1, 2, \dots, n.$$

ВС с иерархической организацией

$$B(X) = \left(\prod_{p=1}^{n-1} \prod_{q=p+1}^n (x_p x_q (b_{pq} - 1) + 1) \right) \rightarrow \max_{(x_p)}$$

при ограничениях:

$$\sum_{p=1}^n x_p = M,$$

$$x_p \in \{0, 1\}, \quad p = 1, 2, \dots, n.$$

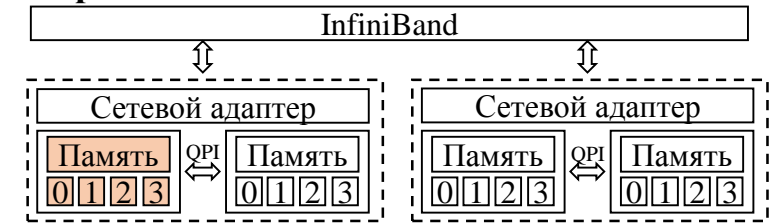
АЛГОРИТМЫ ФОРМИРОВАНИЯ ПОДСИСТЕМ ЭЛЕМЕНТАРНЫХ МАШИН

- Задан ранг R подсистемы ЭМ (количество необходимых процессорных ядер), конфигурация ВС на базе многопроцессорных узлов с общей памятью
- Требуется из K допустимых вариантов подсистем ранга R выбрать подсистему ЭМ, обеспечивающую минимум времени выполнения информационных обменов

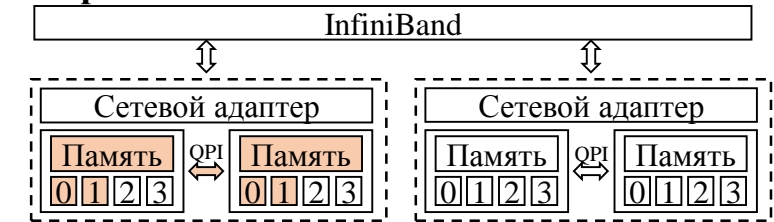
Алгоритм формирования подсистем ЭМ, минимизирующий время реализации коллективных обменов типа «all-to-all» и учитывающий загруженность каналов связи, возникающую в следствии их конкурентного использования процессами параллельных программ

Формирование подсистемы ЭМ ранга 4

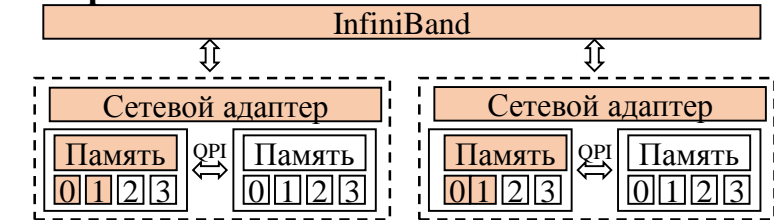
Вариант 1



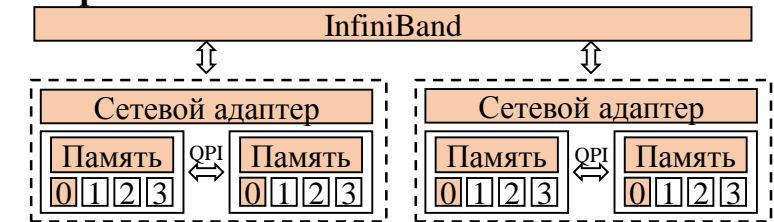
Вариант 2



Вариант 3

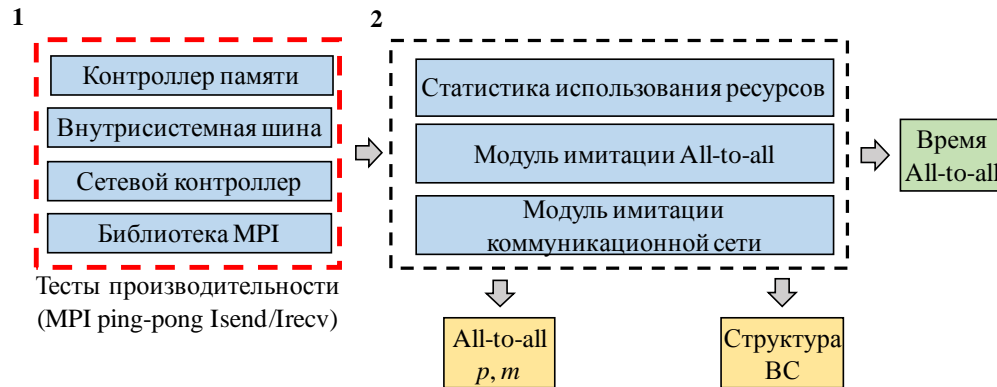


Вариант 4



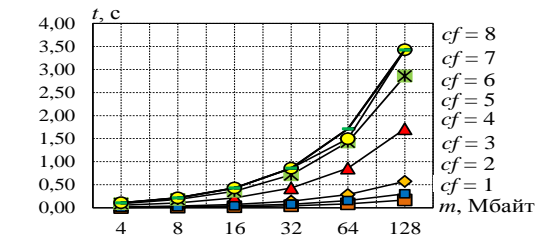
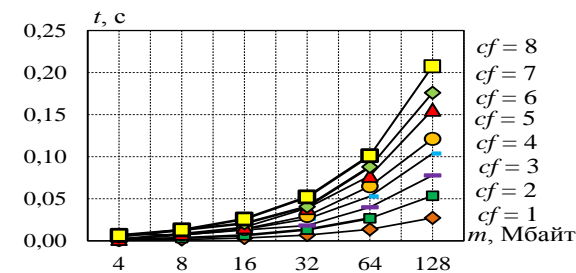
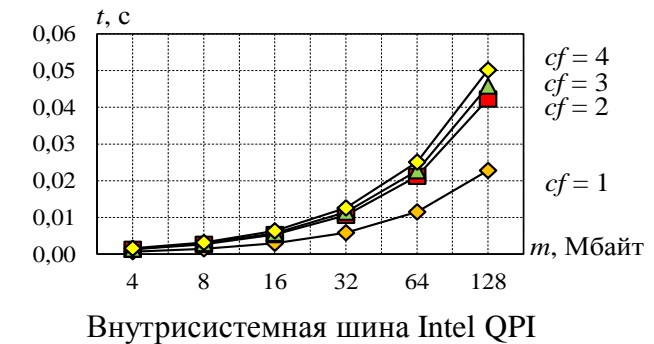
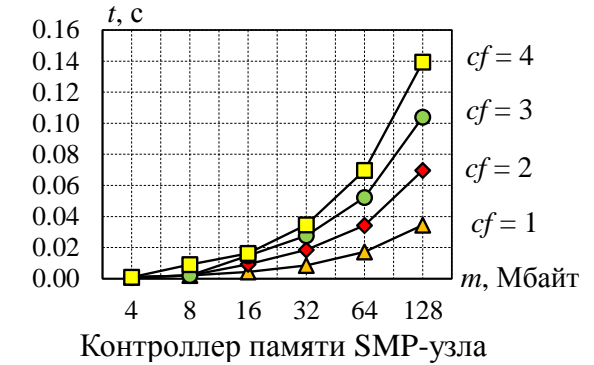
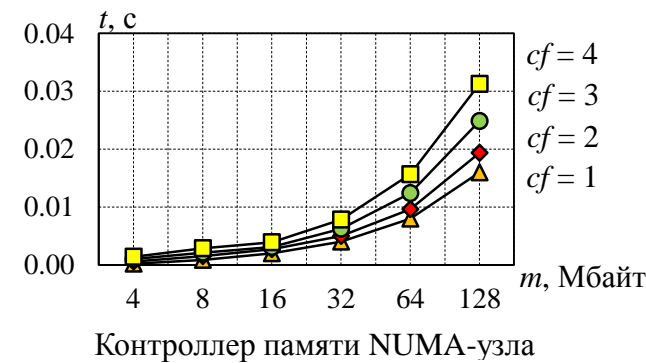
АНАЛИЗ КОНКУРЕНТНОГО ИСПОЛЬЗОВАНИЯ КАНАЛОВ СВЯЗИ

Метод формирования подсистем ЭМ: 1 этап



m , Мбайт	Коэффициент падения производительности $t(m, cf) / t(m, 1)$			
	$cf = 1$	$cf = 2$	$cf = 3$	$cf = 4$
128	1	2,03	3,03	4,06
64	1	2,00	3,04	4,06
32	1	2,17	3,23	4,03
16	1	2,17	3,43	3,80
8	1	1,01	1,01	4,27
4	1	0,86	0,82	0,96

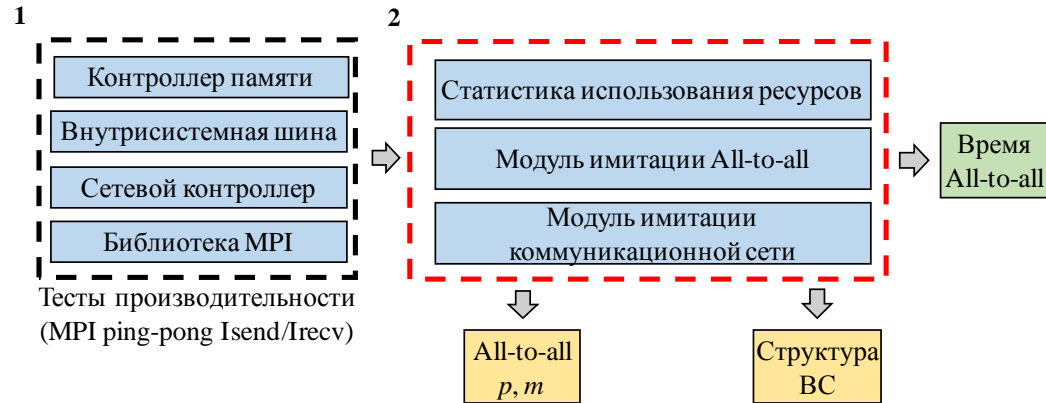
Время t передачи сообщения размером m байт при одновременном разделении канала связи cf процессами



[*] E. Peryshkova, M. Kurnosov Experimental Study of Network Contention Effects on All-to-All Operation // Proc. of the 14th International Scientific-Technical Conference «Actual Problems of Electronic Instrument Engineering» (APEIE-2018), 2018. – Vol. 1, Part 4. – P. 506-510. (Scopus)

АНАЛИЗ КОНКУРЕНТНОГО ИСПОЛЬЗОВАНИЯ КАНАЛОВ СВЯЗИ

Метод формирования подсистем ЭМ: 2 этап



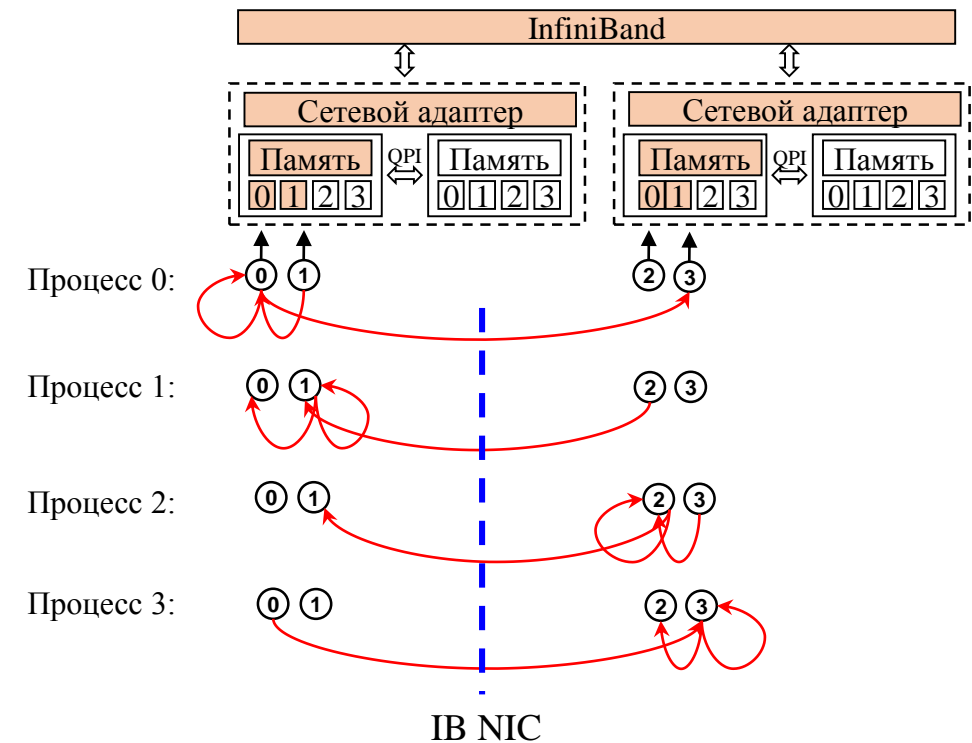
Шаг 1, размер блока 2:

- **Процесс 0:** отправляет и принимает сообщение процессу 0, принимает от процесса 1 и отправляет 3
- **Процесс 1:** отправляет и принимает сообщение процессу 1, принимает от процесса 2 и отправляет 0
- **Процесс 2:** отправляет и принимает сообщение процессу 2, принимает от процесса 3 и отправляет 1
- **Процесс 3:** отправляет и принимает сообщение процессу 3, принимает сообщение от 0 и отправляет 2

Сетевой адаптер первого узла разделяют 4 процесса

$$t(m, 4)$$

Шаг блочного алгоритма операции MPI_Alltoall



ЭВРИСТИЧЕСКИЙ АЛГОРИТМ ФОРМИРОВАНИЯ ПОДСИСТЕМ ЭМ

Входные данные: S – множество симметричных подсистем

Выходные данные: S – упорядоченное по возрастанию оценочного времени выполнения операции «all-to-all» множество симметричных подсистем

```
function ALLOCATESUSBSYS ( $S$ )  
    for each  $subsystem[i]$  in  $S$  do  
         $time[i] = ESTIMATEONSUBSYSTEM(i)$   
    end for  
    SORT( $S$ , COMPARETIME)  
    return  $S$   
end function
```

Входные данные: $subsystem$ – симметричная подсистема ЭМ, m – размер передаваемого сообщения

Выходные данные: $totaltime$ – прогнозируемое время выполнения операции All-to-All

```
function ESTIMATEONSUBSYSTEM ( $subsystem, m$ )  
     $totaltime = 0$   
    for all  $steps$  of All-to-All do  
         $time = 0$   
         $l = ESTIMATELAYER(step)$   
         $cf = ESTIMATECONTENTIONFACTOR(step, l)$   
         $time = t(l, m, cf)$   
         $totaltime = totaltime + time$   
    end for  
    return  $argmax \{totaltime\}$   
end function
```

ИССЛЕДОВАНИЕ АЛГОРИТМА ФОРМИРОВАНИЯ ПОДСИСТЕМ ЭМ

- Алгоритм реализован на языке C++
- Конфигурация ВС: 6 NUMA-узлов (2 x Intel Quad Xeon E5620, RAM 24 GiB), InfiniBand QDR
- Программное обеспечение: GNU/Linux x86-64, MVAPICH 2.3

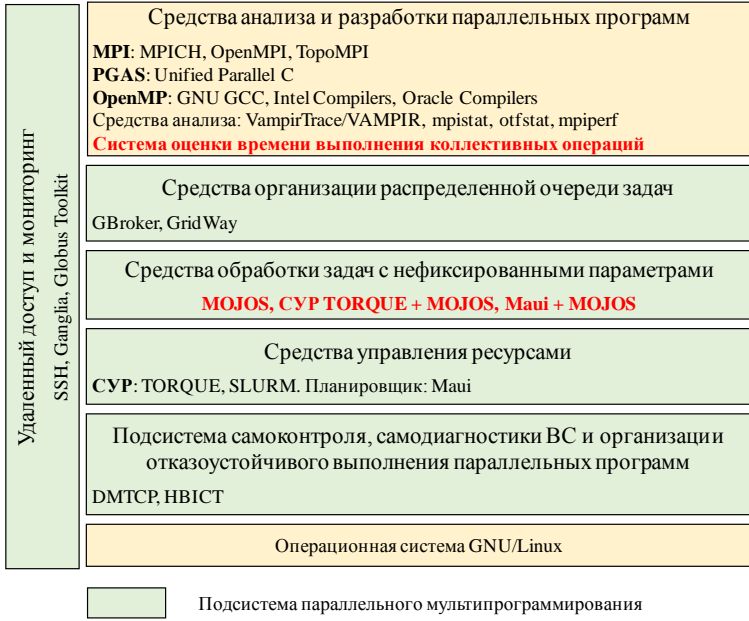
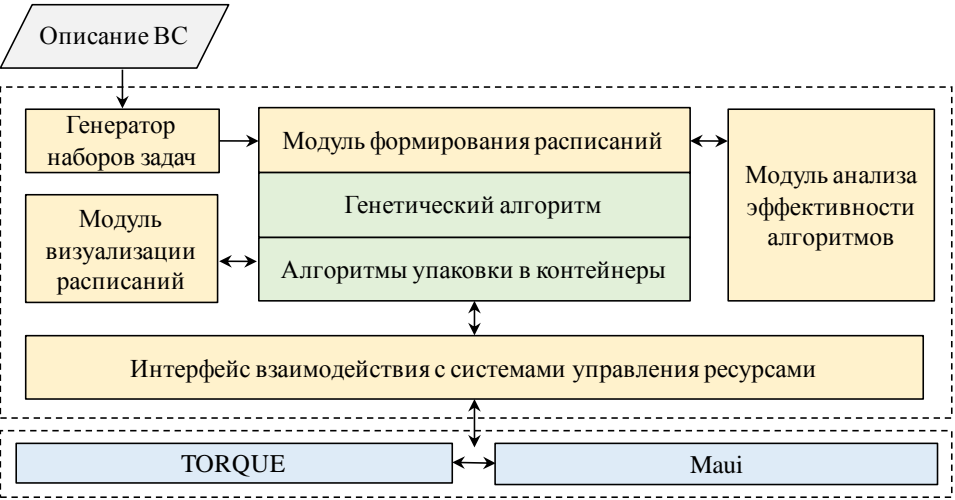
Требуемое количество процессов	Время выполнения операции All-to-all, с		
2	1 ВУ, 2 процессора	2 ВУ, 1 процессор	
Система моделирования	0,00016	0,00033	
Реальный запуск	0,00044	0,00061	
Установленный порядок	1	2	
4	1 ВУ, 4 процессора	2 ВУ, 2 процессора	4 ВУ, 1 процессор
Система моделирования	0,0019	0,0021	0,0018
Реальный запуск	0,0031	0,0036	0,0029
Установленный порядок	2	3	1
8	1 ВУ, 8 процессоров	2 ВУ, 4 процессора	4 ВУ, 2 процессора
Система моделирования	0,00384	0,19	0,0058
Реальный запуск	0,00754	0,09	0,0076
Установленный порядок	1	3	2

На вычислительных кластерах с многопроцессорными NUMA-узлами и сетью связи стандарта InfiniBand алгоритм *обеспечивает сокращение времени информационных обменов от 16% до 31%* по сравнению с известным алгоритмом формирования подсистем FF (first fit)

МУЛЬТИКЛАСТЕРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА

ИНСТРУМЕНТАРИЙ ПАРАЛЛЕЛЬНОГО МУЛЬТИПРОГРАММИРОВАНИЯ

- На основе созданных алгоритмов разработаны программные пакеты MOJOS обслуживания потоков параллельных задач с нефиксированными параметрами, программное расширение системы управления ресурсами TORQUE, программный пакет поддержки режима обслуживания потока задач с нефиксированными параметрами для планировщика Maui (в соавторстве), программные средства оценки времени реализации коммуникационных операции типа «all-to-all» (в соавторстве)
- Предложенные пакеты вошли в состав инструментария параллельного мультипрограммирования пространственно-распределенной мультикластерной ВС, созданной членами ведущей научной школы РФ (НШ-9505.2006.9, НШ-2121.2008.9, НШ-5176.2010.9, НШ-2175.2012.9, основатель – чл.-корр. РАН В.Г. Хорошевский)



[*] А.В. Ефимов, С.Н. Мамоиленко, Е.Н. Перышкова Модернизация системы управления ресурсами PBS/TORQUE и планировщика Maui для обслуживания масштабируемых задач // Вестник компьютерных и информационных технологий. – 2016. – № 2. – С. 34-39.



Спасибо за внимание!

Перышкова Евгения Николаевна

к.т.н. доцент Кафедры вычислительных систем

Сибирский государственный университет

телекоммуникаций и информатики

e-mail: e.peryshkova@gmail.com