



Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики

Кафедра Математического обеспечения ЭВМ

Параллельный алгоритм переупорядочения графа разреженной матрицы для минимизации заполнения в разложении Холецкого

Аспирант каф. МО ЭВМ ф-та ВМК

Пирова А.Ю.

Москва, 5 марта 2015 г.

Содержание

- ❑ Разложение Холецкого
- ❑ Постановка задачи минимизации заполнения фактора
- ❑ Методы переупорядочения
- ❑ Метод вложенных сечений
- ❑ Схема переупорядочения в MORSy
- ❑ Параллельная версия для систем с общей памятью
- ❑ Результаты экспериментов. Сравнение с ParMETIS
- ❑ Заключение



Разложение Холецкого (1)

- Пусть дана система линейных уравнений:

$$Ax = b \quad (1)$$

A – разреженная симметричная положительно определенная матрица, b – плотный вектор, x – вектор неизвестных.

- Прямой метод решения – разложение Холецкого:

$$A = LL^T \quad (2)$$

L – нижнетреугольная матрица, называемая *фактором* матрицы A

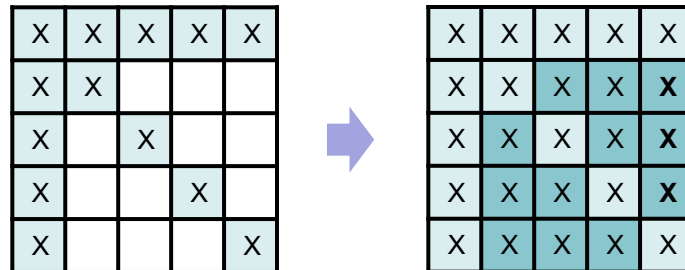
- Переход к решению треугольных систем

$$Lx = b, L^T y = x \quad (3)$$



Разложение Холецкого (2)

- Особенность разложения в разреженном случае – *заполнение* (снижение разреженности) фактора L :

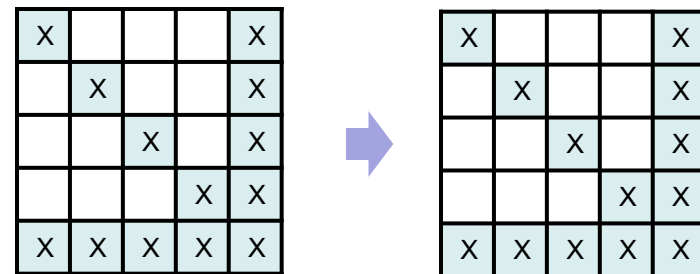


- Способ снизить заполнение – переупорядочение строк матрицы:

$$\underbrace{(PAP^T)}_{\text{matrix}} \underbrace{(Px)}_{\text{vector}} = \underbrace{Pb}_{\text{vector}}$$

Переход к системе $\hat{A}\hat{x} = \hat{b}$ (4)

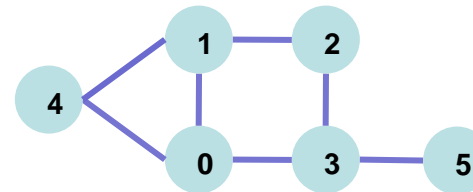
P – матрица перестановки



Постановка задачи минимизации заполнения фактора (1)

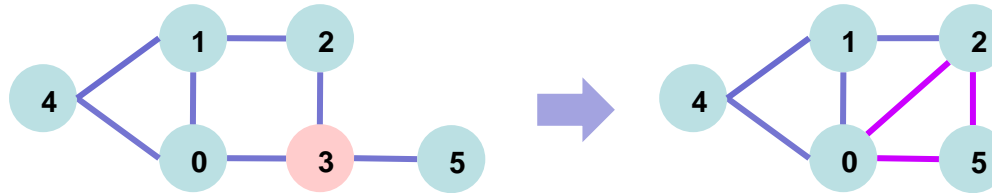
- *Граф матрицы* $A = (a_{ij})$ размера $n \times n$: $G = (V, E)$, где
 - каждая вершина v_i из V соответствует строке i .
 - каждое ребро (v_i, v_j) из E соответствует элементу $a_{ij} \neq 0, i \neq j$. Других ребер нет.

	0	1	2	3	4	5
0	9	1			2	
1	1	8	3		2	
2		3	9	4		1
3			4	7		
4	2	2			5	
5			1			6



Постановка задачи минимизации заполнения фактора (2)

- *Исключение* вершины v из графа:
 - Вершина исключается вместе с исходящими из нее ребрами
 - Множество смежных v вершин становится кликой

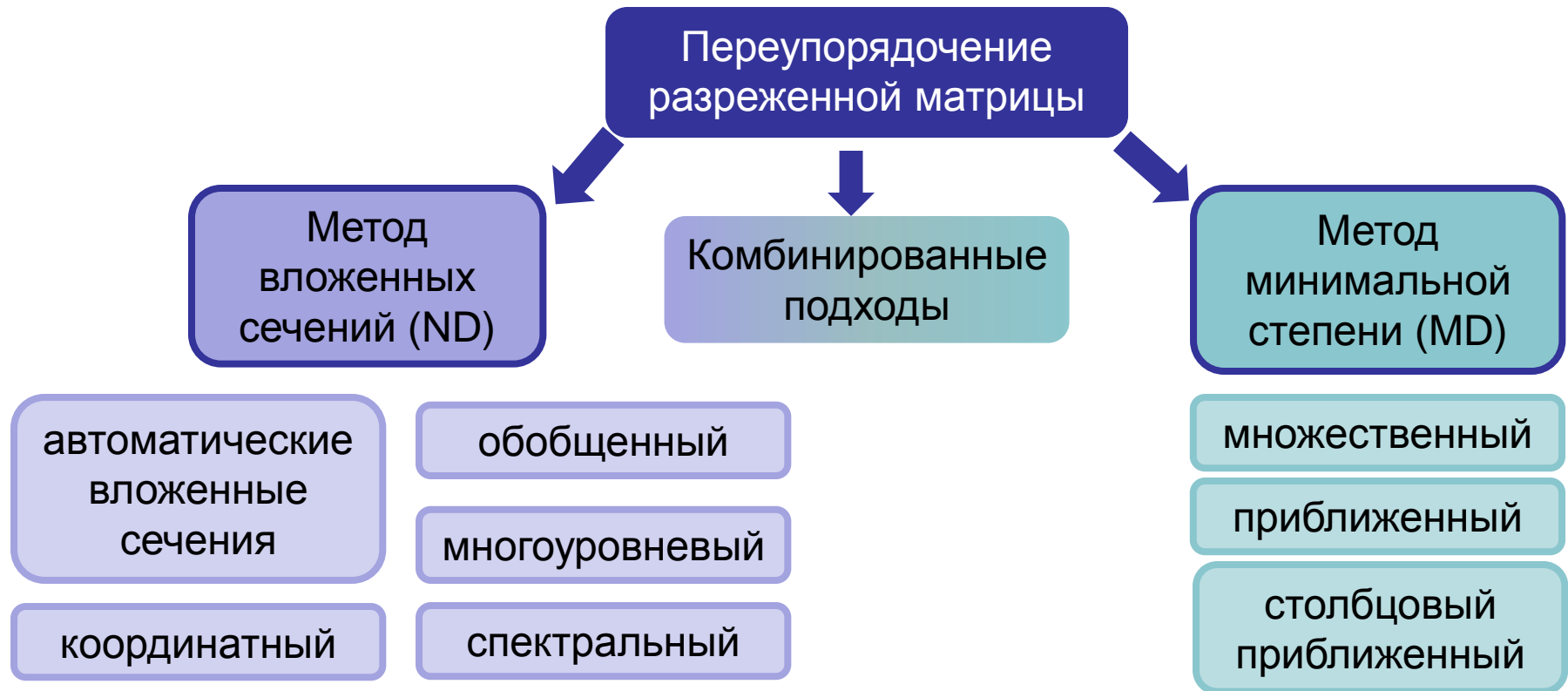


Постановка задачи минимизации заполнения фактора (3)

- Пусть $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ – перестановка множества вершин V .
- *Заполнение, порожденное перестановкой π , $F(\pi)$* – множество ребер, добавленных при последовательном исключении вершин $\pi_1, \pi_2, \dots, \pi_n$.
- Задача: найти перестановку π^* , такую, что $|F(\pi^*)| \rightarrow \min$.
- Задача нахождения такой перестановки NP-полная (*Yannakakis, 1981*).



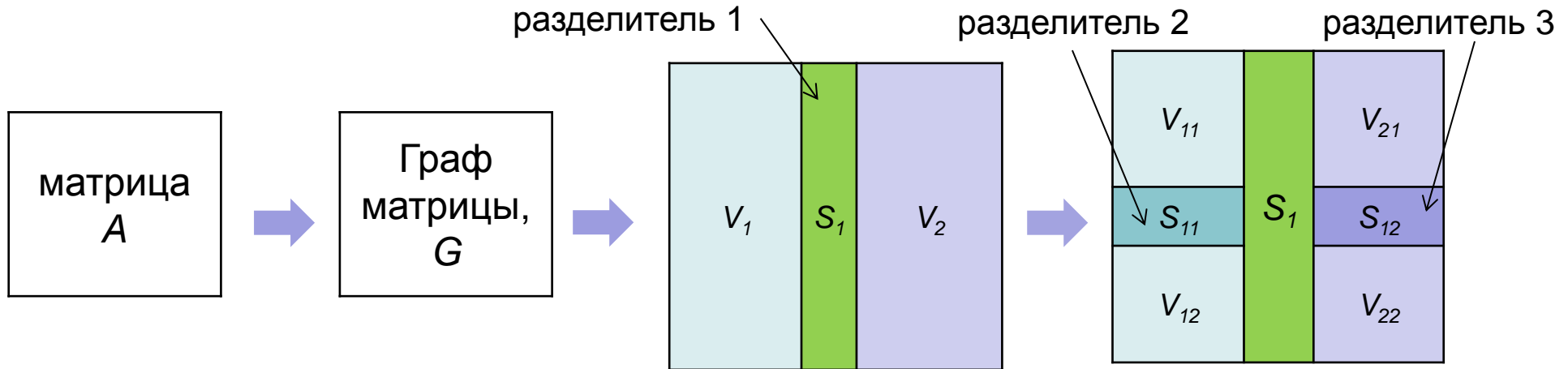
Методы переупорядочения графов разреженных матриц



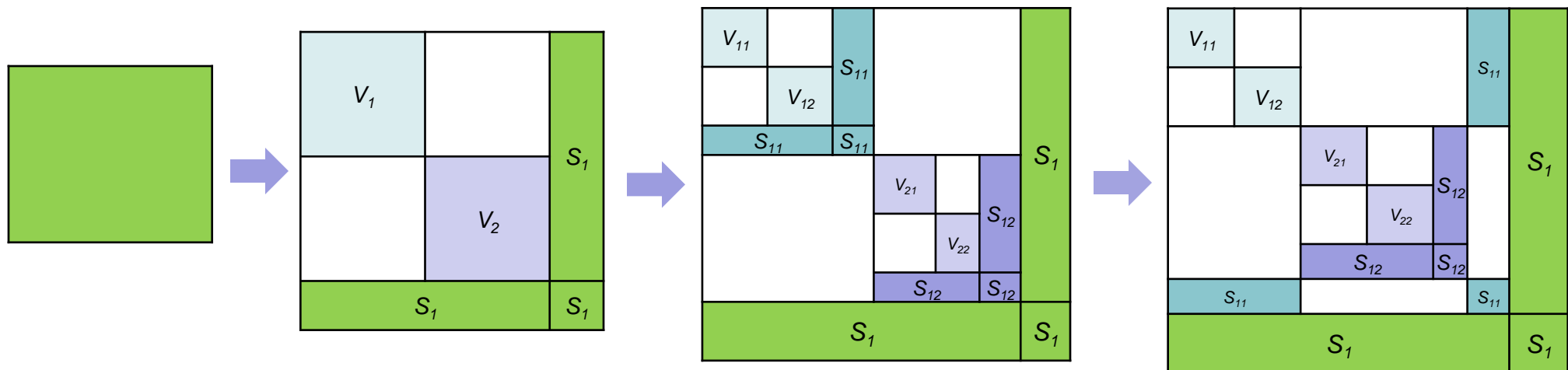
Известные пакеты: METIS, Scotch, Chaco;
переупорядочиватели в составе решателей (MUMPS,
MKL, SuperLU...).

Параллельные версии: ParMETIS, PT-Scotch

Метод вложенных сечений



Заполнение фактора L :



Последовательная версия

- ❑ MORSy – свободно распространяемая кросс-платформенная библиотека для переупорядочения графов симметричных разреженных матриц
<http://hpc-education.unn.ru/en/research/overview/sparse-algebra/morsy>
- ❑ Основана на многоуровневом методе вложенных сечений с модификациями отдельных стадий

Схема переупорядочения в MORSy



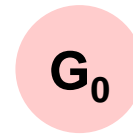
Параллельная версия для систем с общей памятью. Использование очереди задач (1)

- ❑ Технология OpenMP
- ❑ Особенности параллельного алгоритма:
 - Общая очередь задач
 - 1 задача = найти разделитель в подграфе
 - Новые задачи попадают в очередь **после** выполнения потоком текущей задачи



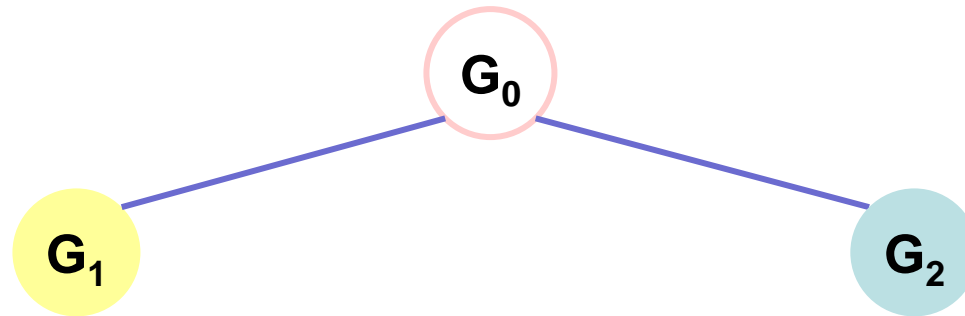
Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач



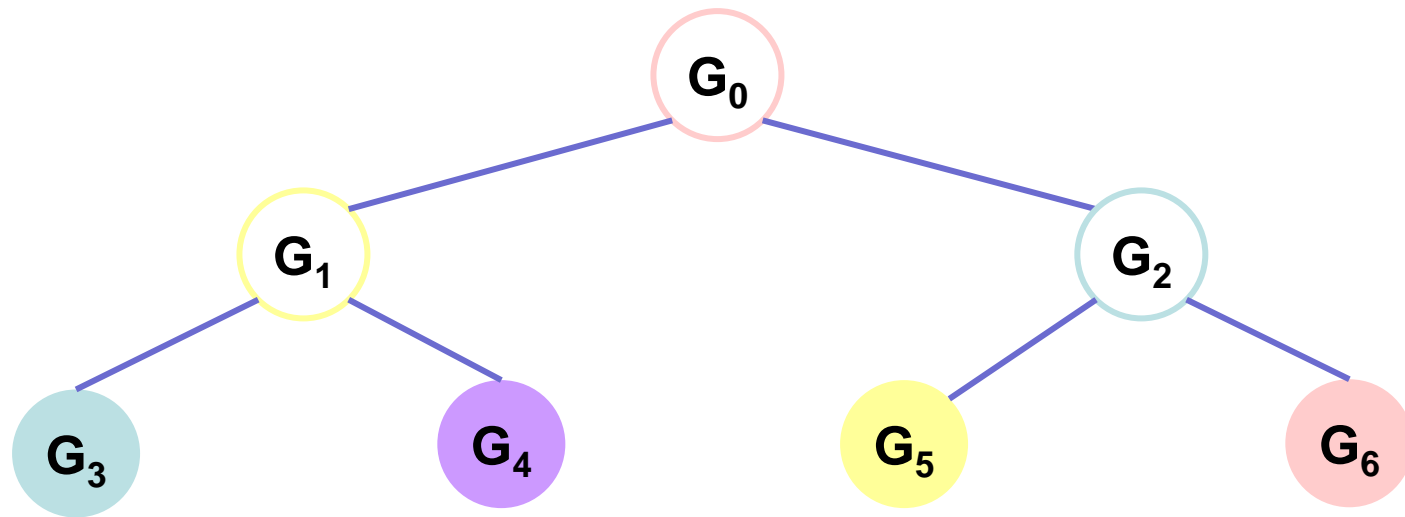
Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач



Параллельная версия для систем с общей памятью. Использование очереди задач (2)

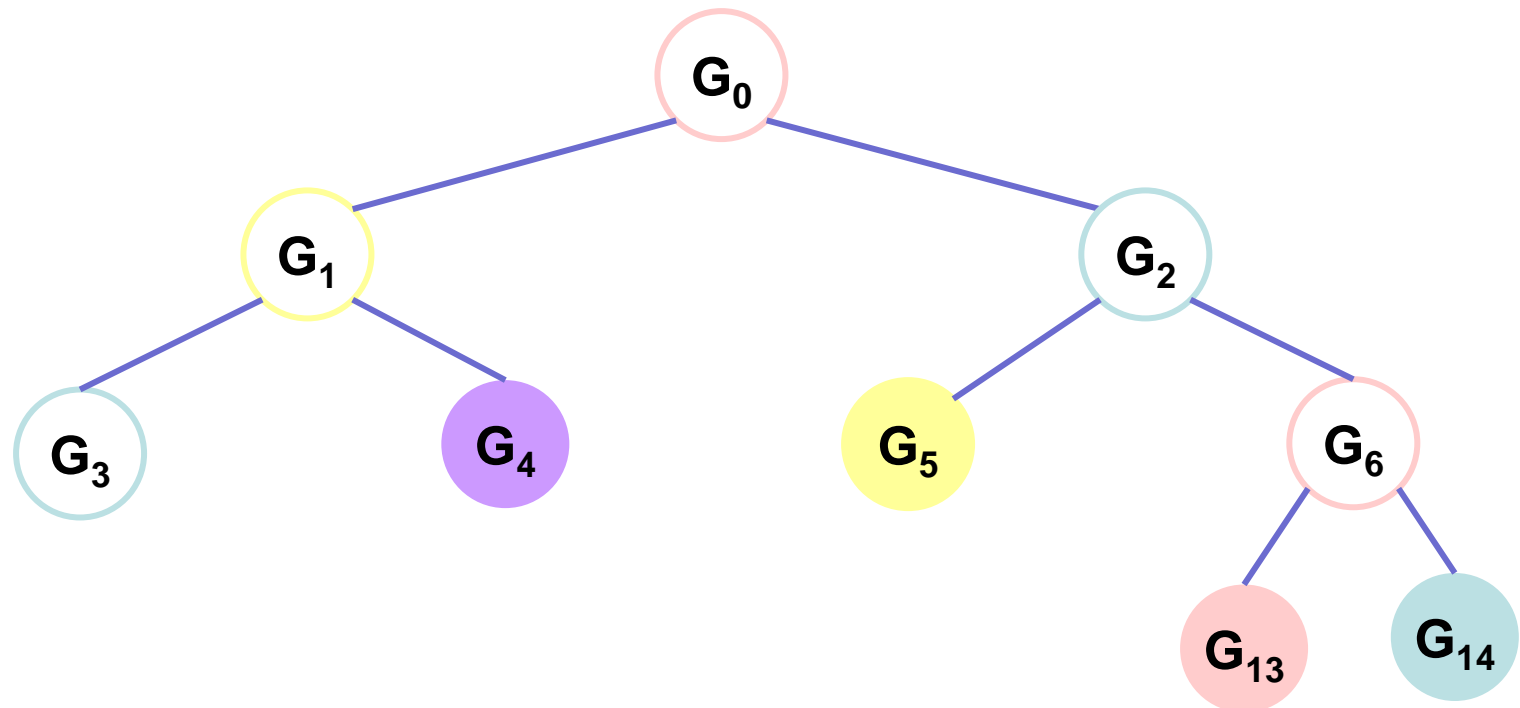
Очередь задач



Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач

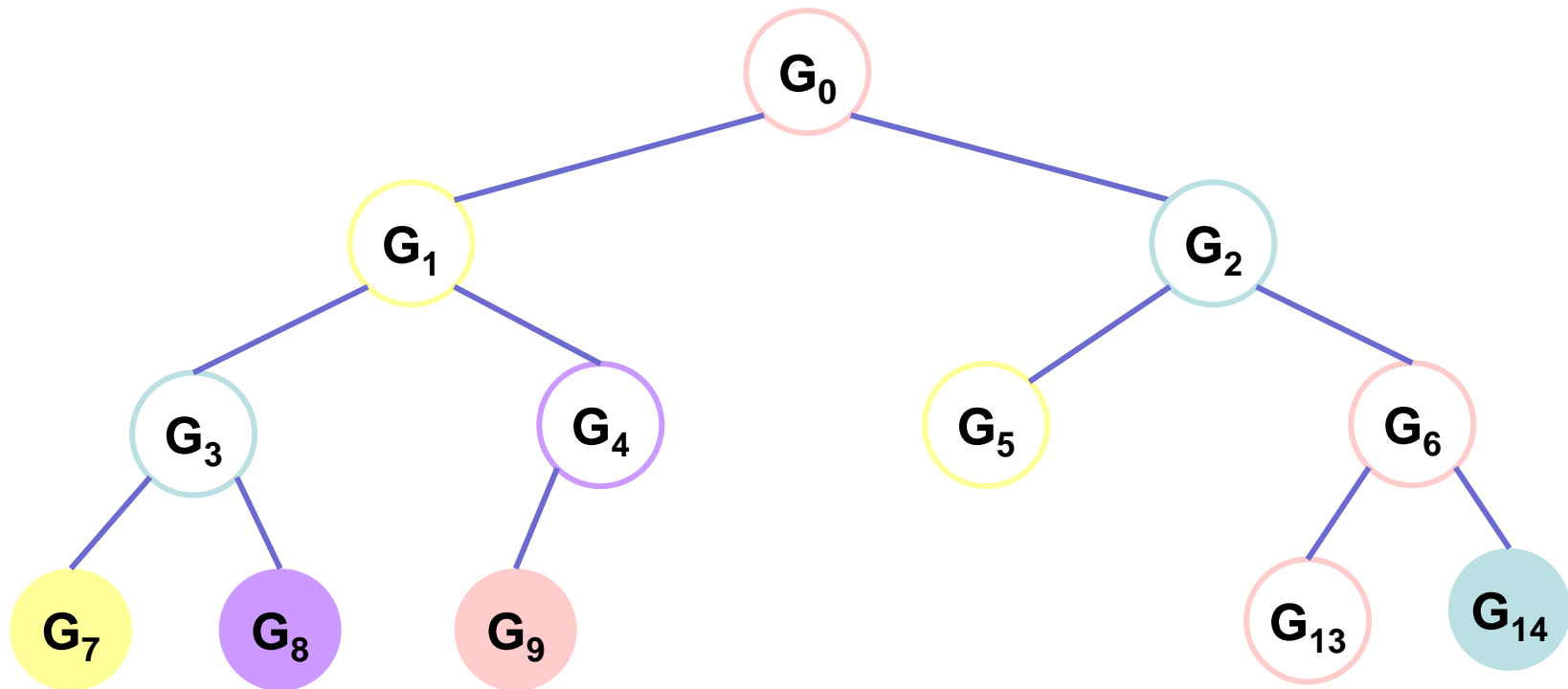
G_0	G_1	G_2	G_5	G_6	G_3	G_4	G_{13}	G_{14}	G_7	G_8					
-------	-------	-------	-------	-------	-------	-------	----------	----------	-------	-------	--	--	--	--	--



Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач

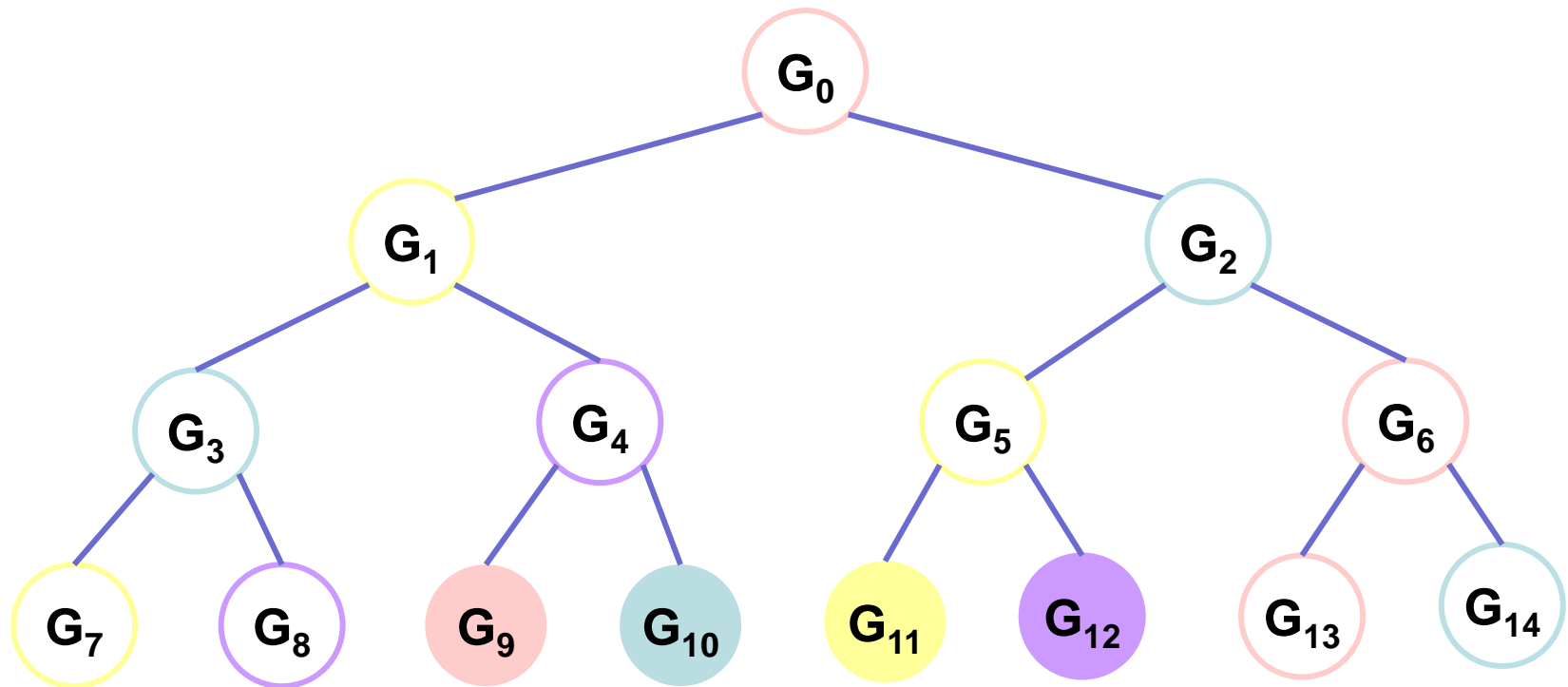
G_0	G_1	G_2	G_5	G_6	G_3	G_4	G_{13}	G_{14}	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}
-------	-------	-------	-------	-------	-------	-------	----------	----------	-------	-------	-------	----------	----------	----------



Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач

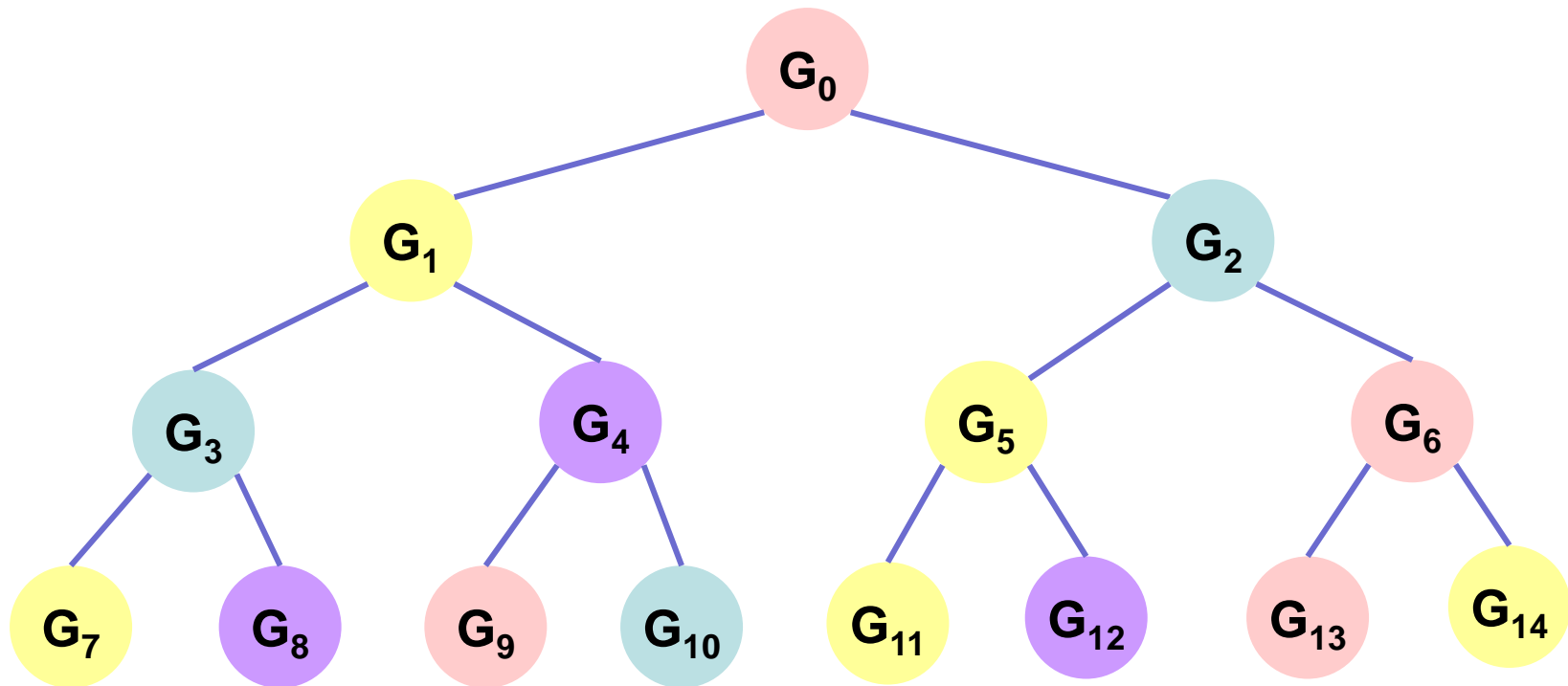
G_0	G_1	G_2	G_5	G_6	G_3	G_4	G_{13}	G_{14}	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}
-------	-------	-------	-------	-------	-------	-------	----------	----------	-------	-------	-------	----------	----------	----------



Параллельная версия для систем с общей памятью. Использование очереди задач (2)

Очередь задач

G_0	G_1	G_2	G_5	G_6	G_3	G_4	G_{13}	G_{14}	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}
-------	-------	-------	-------	-------	-------	-------	----------	----------	-------	-------	-------	----------	----------	----------



Параллельная версия для систем с общей памятью. Использование задач OpenMP

- ❑ 1 задача = найти разделитель в подграфе
- ❑ Рекурсивный алгоритм обхода дерева задач
- ❑ Задачи запускаются директивой `omp task`



Тестовые матрицы

□ Коллекция университета Флориды:

Название	Порядок, N	Число ненулевых элементов, NZ	Описание
pwtk	217 918	11 524 432	structural problem
msdoor	415 863	19 173 163	structural problem
parabolic_fem	525 825	3 674 625	fluid dynamics problem
tmt_sym	726 713	5 080 961	electromagnetics problem
boneS10	914 898	40 878 708	3D structural problem
Emilia_923	923 136	40 373 538	3D structural problem
audikw_1	943 695	77 651 847	3D problem
bone010	986 703	47 851 783	3D problem
ecology2	999 999	4 995 991	2D structural problem
thermal2	1 228 045	8 580 313	unstructured FEM
StocF-1465	1 465 137	21 005 389	fluid dynamics problem
Hook_1498	1 498 023	59 374 451	3D structural problem
Flan_1565	1 564 794	114 165 372	3D structural problem



Тестовая инфраструктура

□ Параметры тестового окружения:

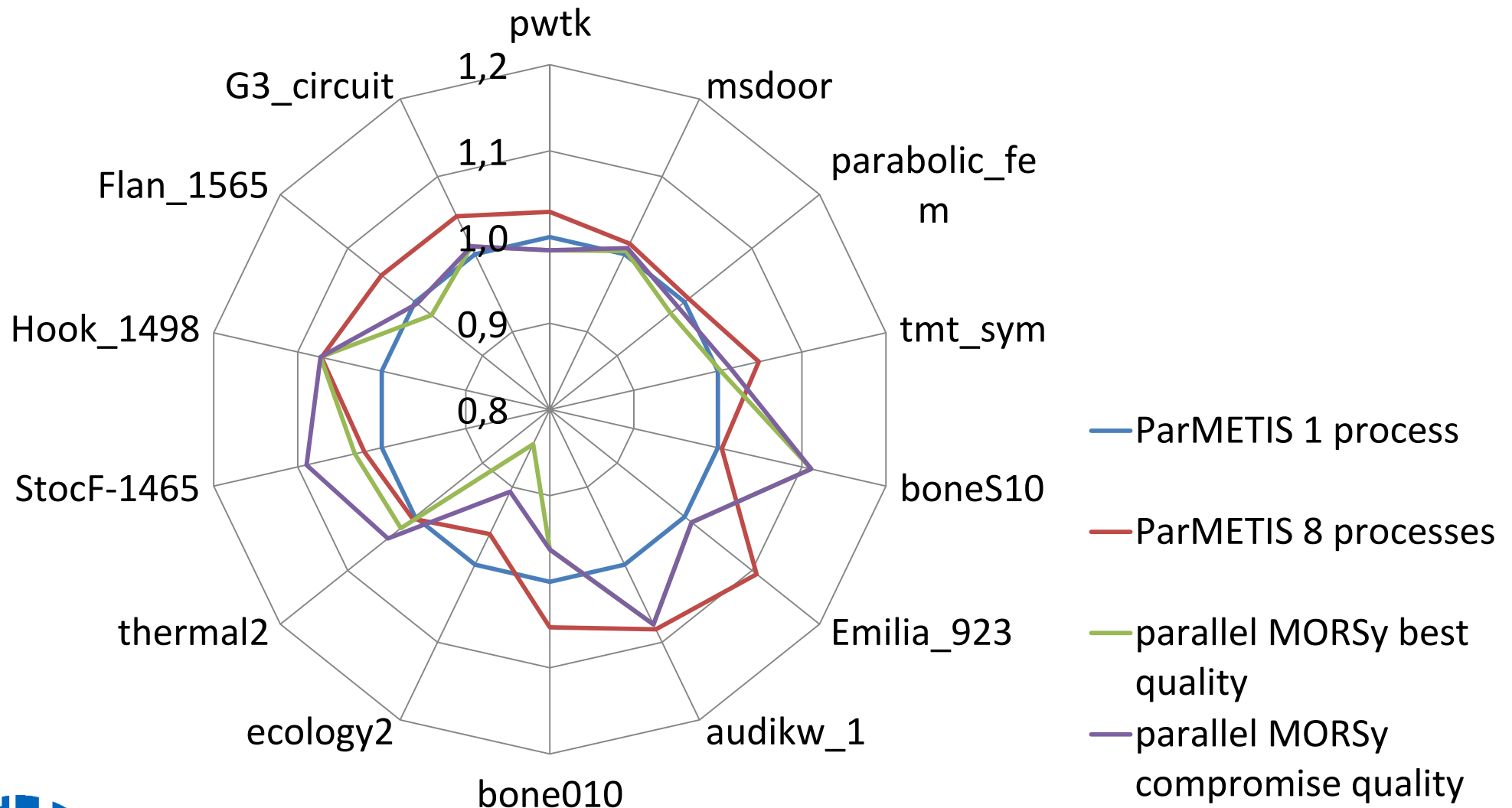
Процессор	Два 4-ядерных процессора Intel® Xeon L5630 (2.13 GHz)
Память	24 GB
Операционная система	Windows Server SP2
Компилятор	Intel C++ Composer (Intel Parallel Studio XE 2013)

□ Сравнение:

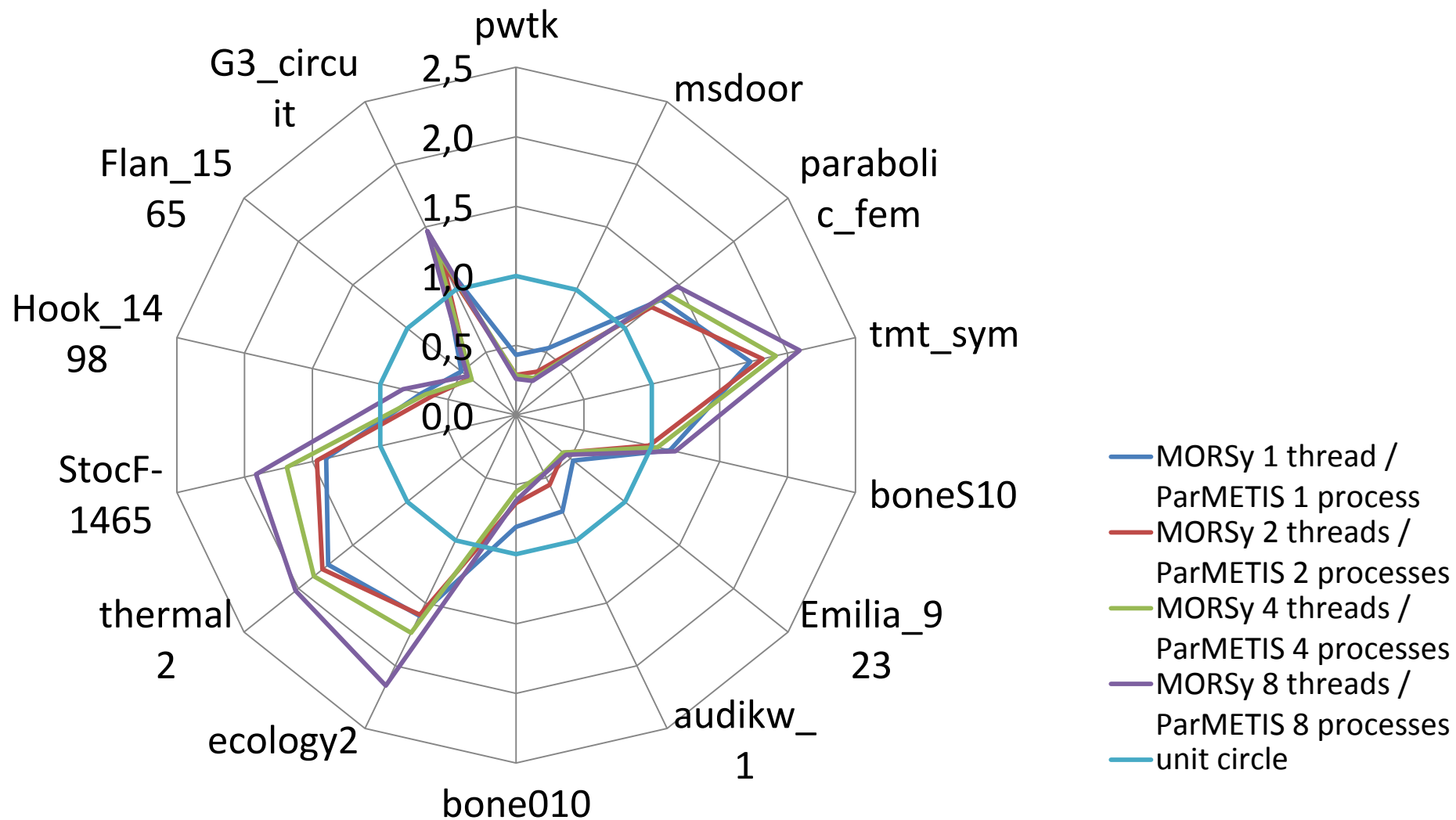
- MORSy, параллельная версия (OpenMP)
- ParMETIS 4.0.3 (MPI)



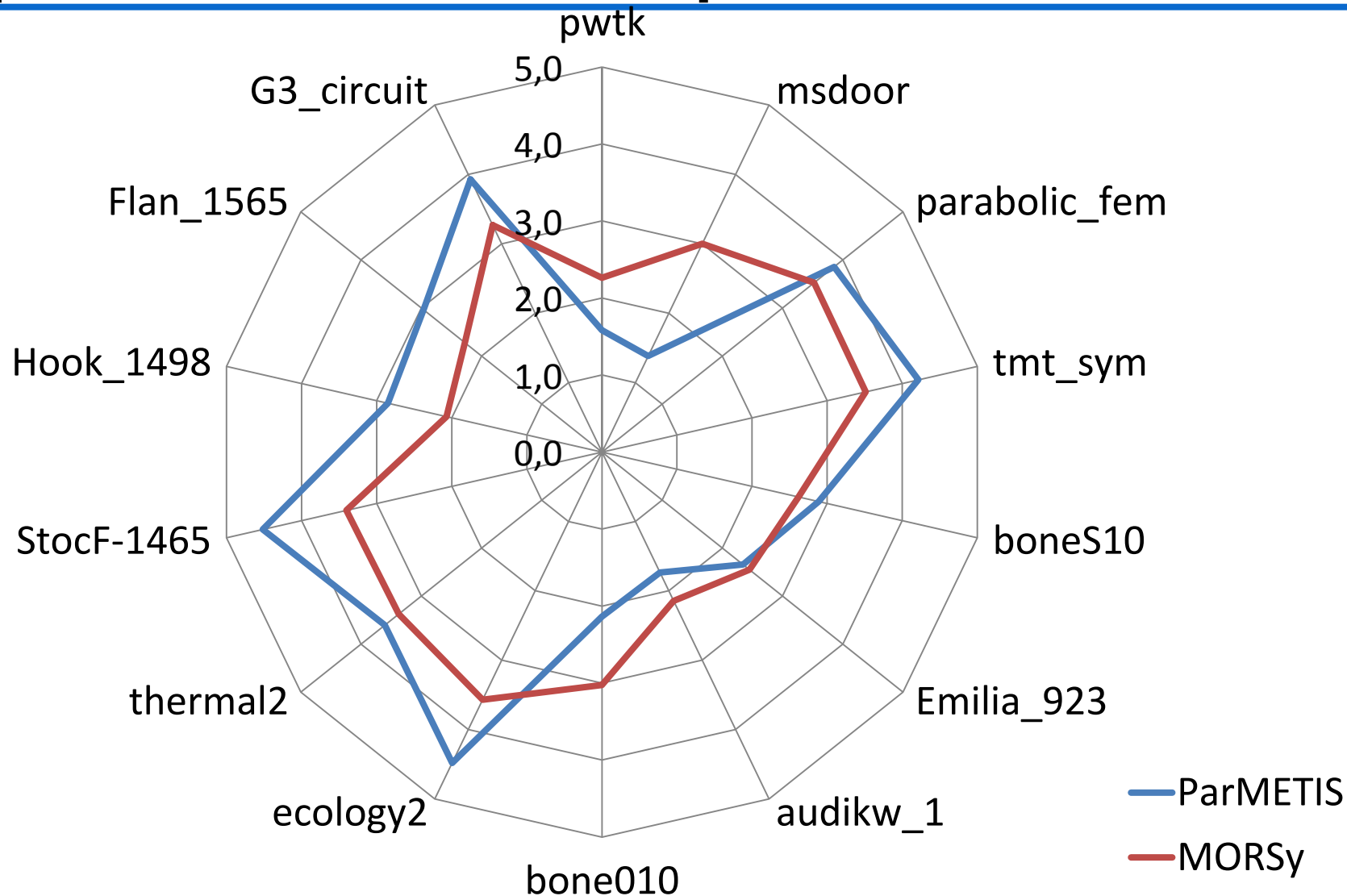
Сравнение качества переупорядочения относительно METIS



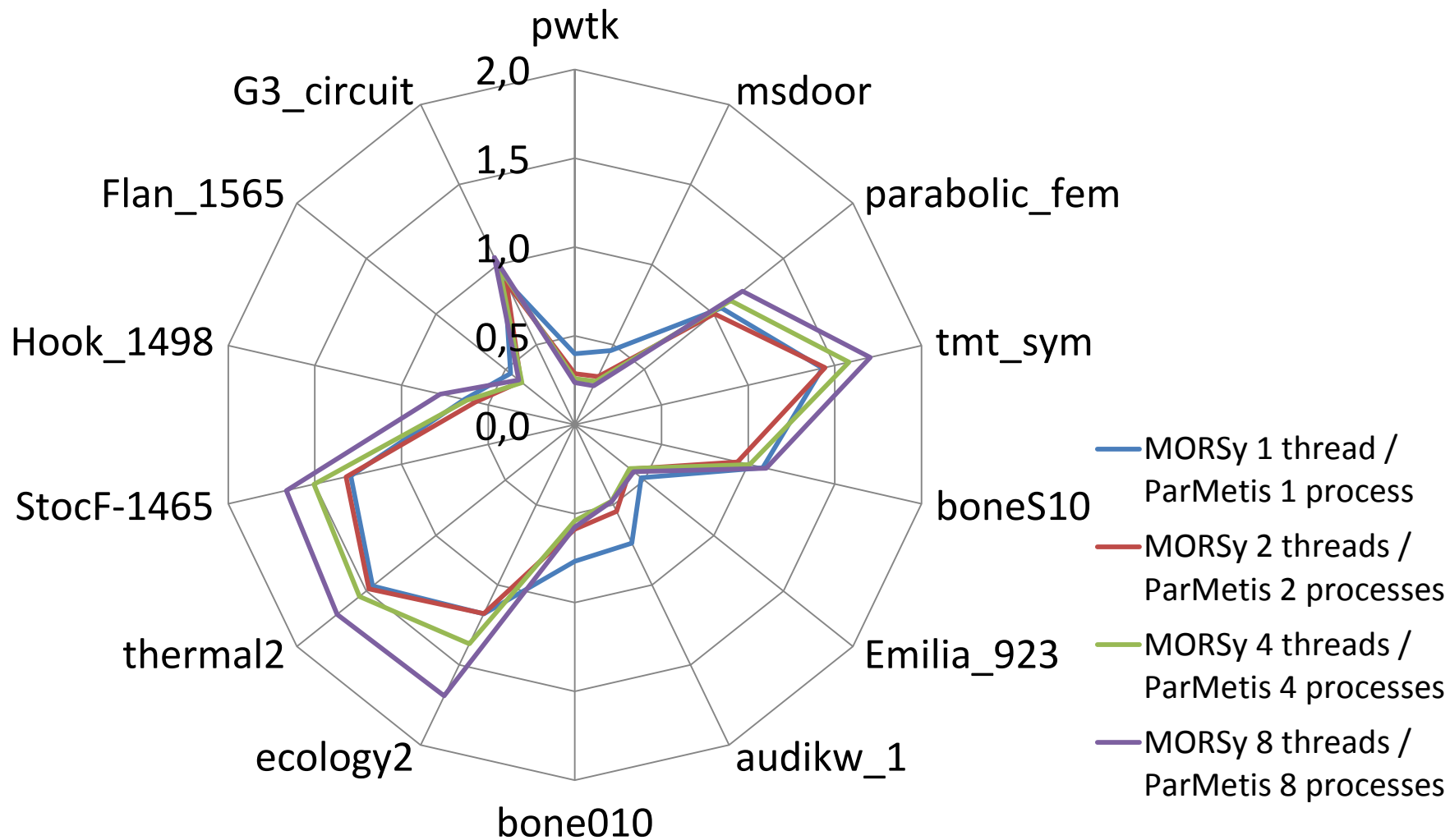
Сравнение времени работы относительно ParMETIS. Использование очереди



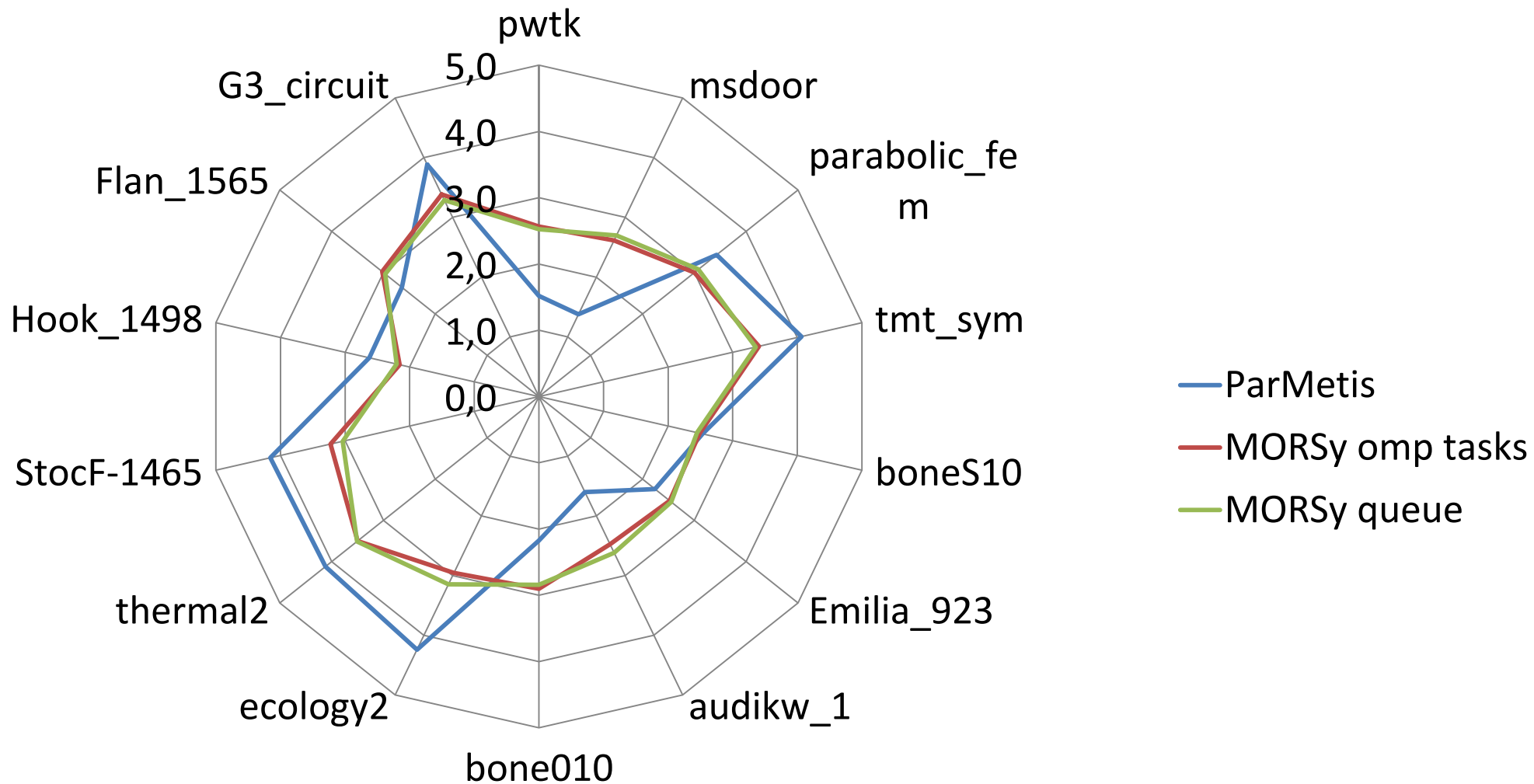
Сравнение масштабируемости при работе в 8 потоков / процессов



Сравнение времени работы относительно ParMETIS. Использование задач OpenMP



Сравнение масштабируемости при работе в 8 потоков / процессов



Заключение

- ❑ Текущая последовательная программная реализация показывает результаты, близкие по качеству и скорости работы к библиотеке METIS – одной из лидирующих в данной области.
- ❑ Текущая параллельная реализация работает быстрее, чем ParMetis на 1 вычислительном узле, на половине тестовых матриц
- ❑ Масштабируемость параллельной реализации характерна для данного класса задач, однако уступает ParMETIS на половине тестовых матриц
- ❑ MORSy доступен со страницы:

<http://hpc-education.unn.ru/research/overview/sparse-algebra/morsy>



Дальнейшие исследования

- Основные направления дальнейших исследований:
 - Развитие программной реализации с целью уменьшения времени решения задачи, увеличения масштабируемости
 - Разработка гибридной MPI + OpenMP версии



Вопросы

□ Вопросы?

Спасибо за внимание!



Многоуровневое выделение разделителя в MORSy

