Лекция 13. Алгоритмы коллективных обменов

Пазников Алексей Александрович

к.т.н., ст. преп. Кафедры вычислительных систем

Сибирский государственный университет

телекоммуникаций и информатики

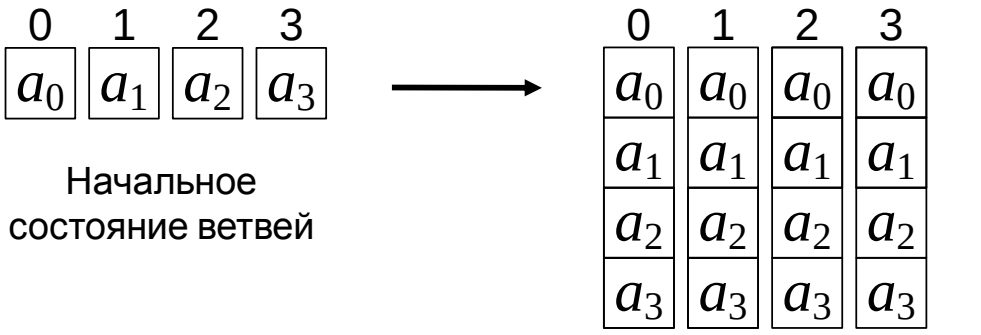
http://cpct.sibsutis.ru/~apaznikov

Трансляционно-циклическая передача данных между ветвями параллельных программ

Каждая ветвь $i \in \{0, 1, \ldots, n-1\}$ параллельной программы располагает локальным сообщением a_i размером m байт.

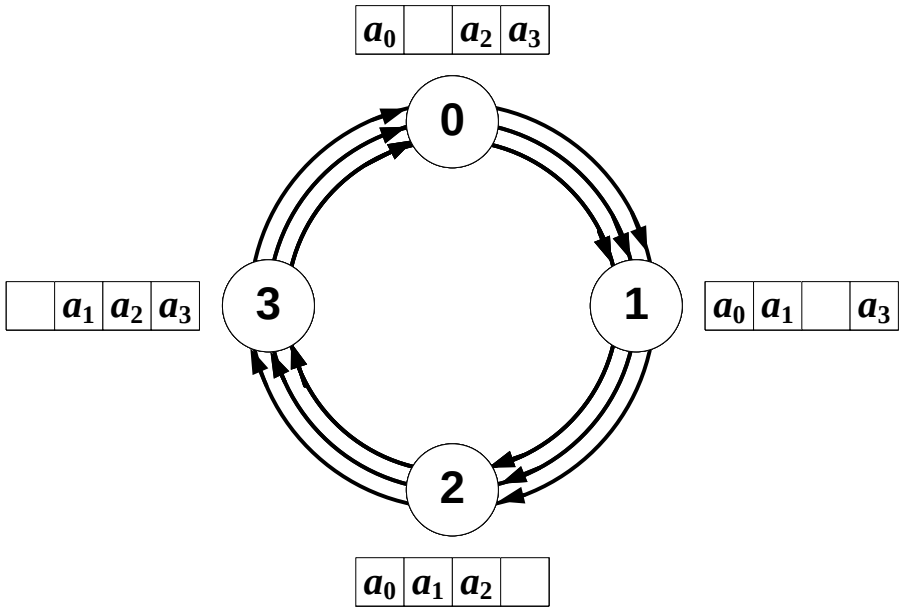
Требуется отправить сообщение a_i всем ветвям и получить сообщения от каждой.

По окончанию ТЦП все ветви должны содержать в своей памяти n сообщений $a_0, a_1, \ldots, a_{n-1}$, упорядоченных по номерам ветвей.



- Message Passing Model (MPI libraries): `MPI_Allgather`
- Partitioned Global Address Space Model (Unified Parallel C): `upc_all_gather_all`

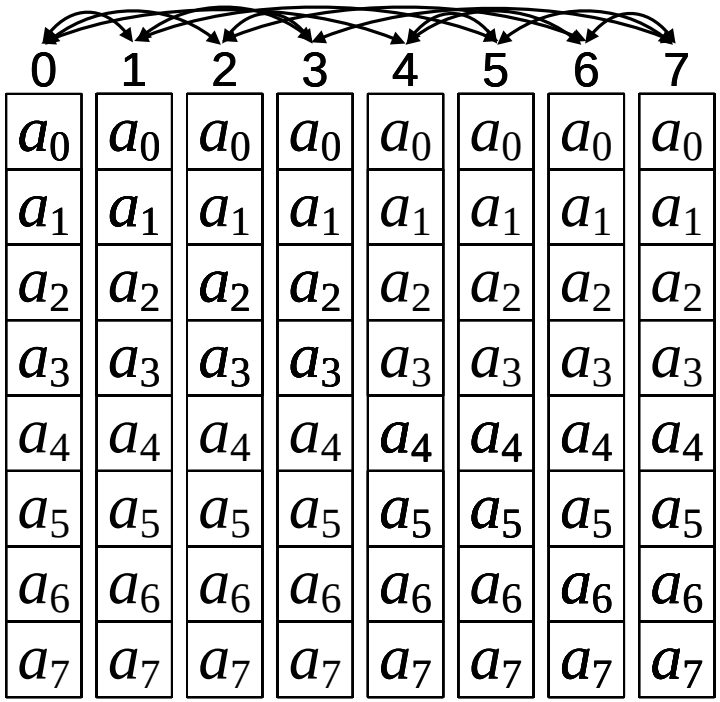
Кольцевой алгоритм (ring)



Каждая ветвь выполняет $2(n-1)$ обменов

Алгоритмы реализации трансляционно-циклической передачи данных в распределённых ВС

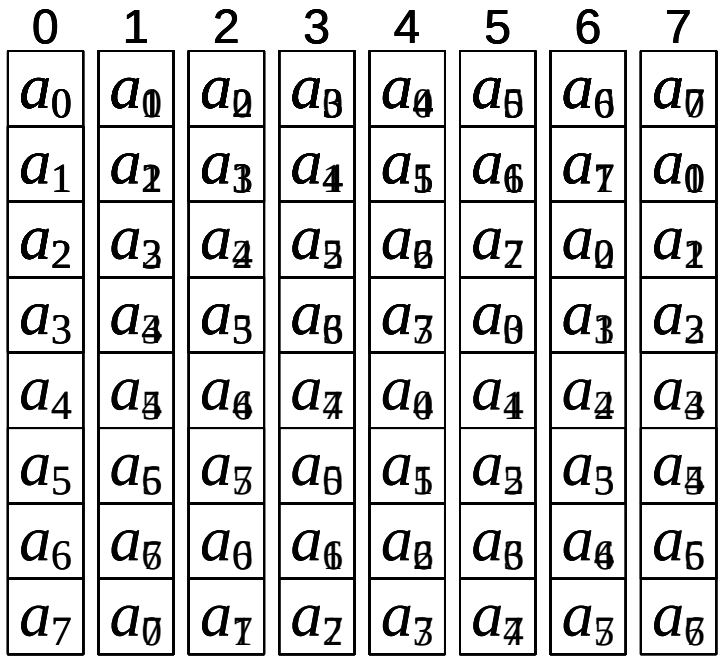
Алгоритм рекурсивного сдваивания (recursive doubling)



Количество обменов: $2\log_2 n$

Только для n равного степени двойки

Алгоритм Дж. Брука (J. Bruck et al., 1997)

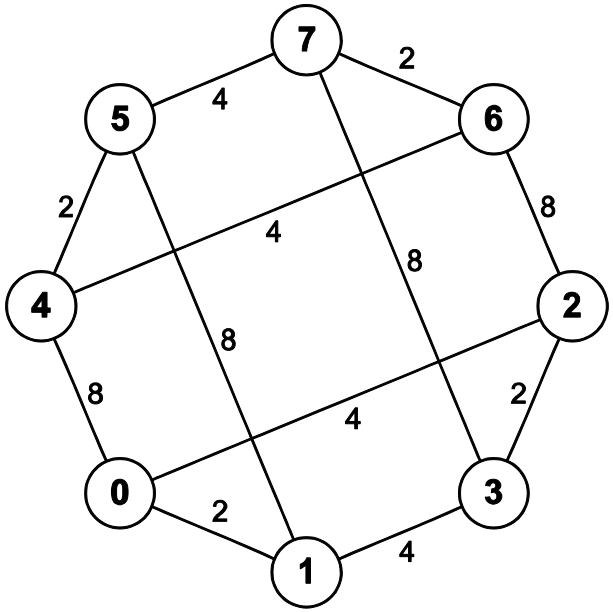


Количество обменов: $2\lceil \log_2 n \rceil$

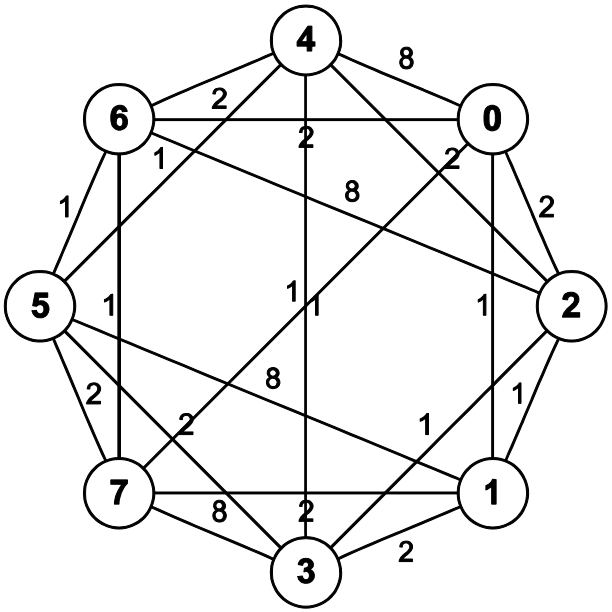
На шаге k ветвь i взаимодействует с ветвями $(i - 2^k + n) \bmod n$ и $(i + 2^k) \bmod n$

На каждом шаге размер передаваемого блока удваивается: $m, 2m, 4m$

Алгоритмы реализации трансляционно-циклической передачи данных в распределённых ВС



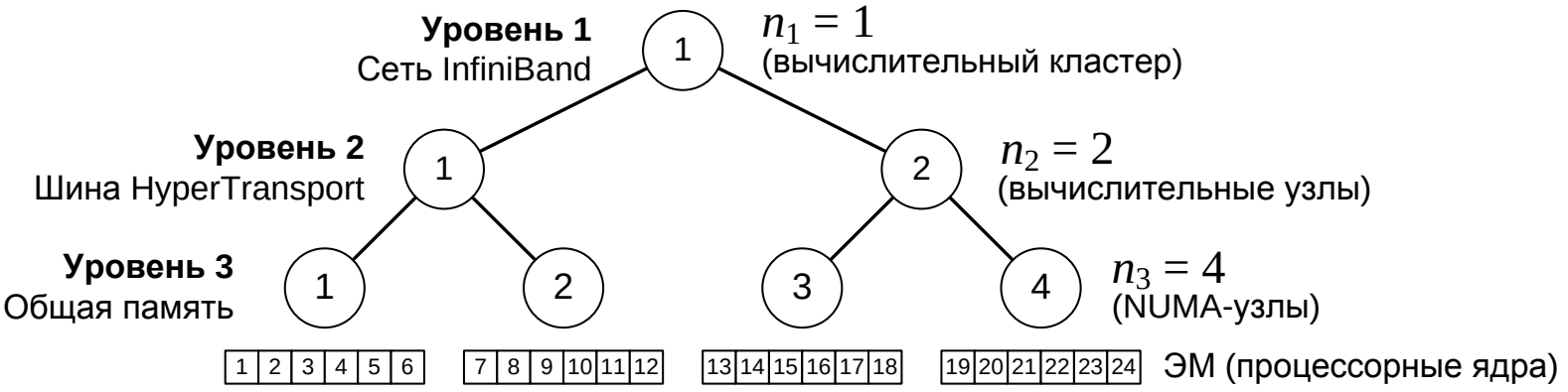
Граф алгоритма
рекурсивного сдваивания для m = 1



Граф алгоритм Дж. Брука
для m = 1

Вес d_{ij} ребра отражает объем данных переданных
по нему при реализации алгоритма

Иерархическая организация распределённых ВС

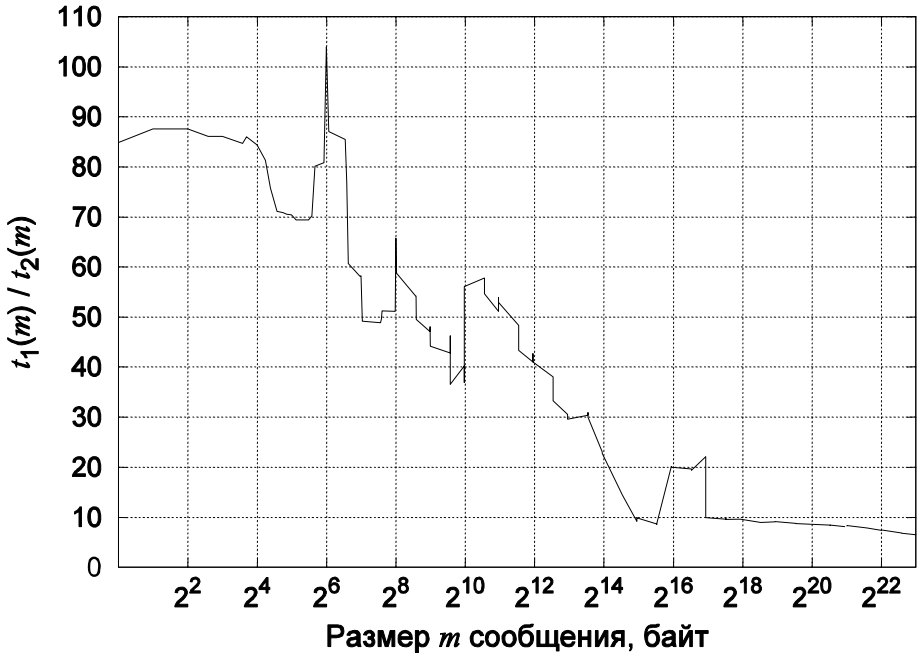


Основные обозначения:

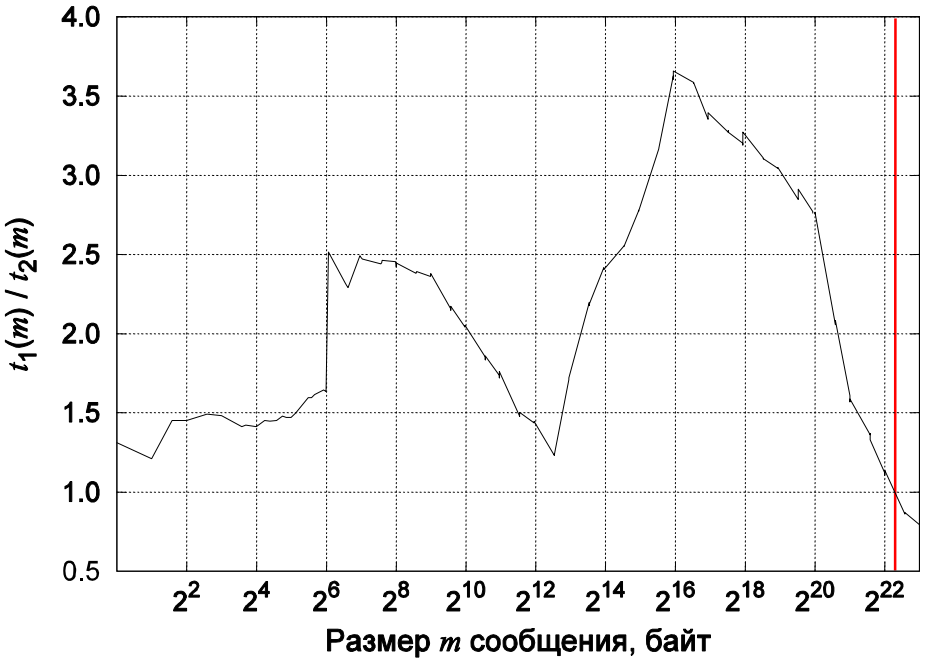
- b_l – максимальное значение пропускной способности каналов связи на уровне l ($[b_l]$ = байт/с).
- $t_l(m)$ – время передачи сообщения размером m байт через каналы связи уровня l ($[t_l(m)]$ = с).

Распределенные ВС из списка TOP500 (34 редакция, ноябрь 2009 года) имеют как минимум два уровня в иерархической организации – общая память узлов и сеть межузловых связей.

Иерархическая организация распределённых ВС



а



б

Отношение времени передачи сообщений через сеть межузловых связей

к времени передачи через общую память узла:

а – кластер ЦПВТ ГОУ ВПО “СибГУТИ” (Gigabit Ethernet / Shared Memory PC5300)

б – кластер ИВЦ ГОУ ВПО “НГУ” (InfiniBand 4x DDR / Shared Memory PC5300)

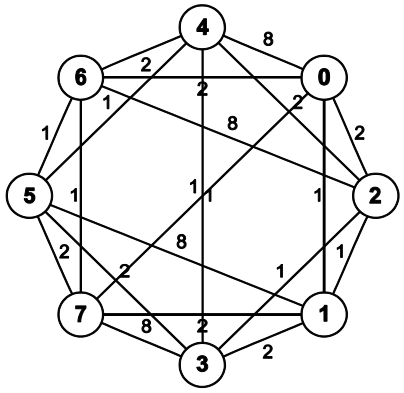
Реализация ТЦП в иерархических распределённых ВС

Реализация ТЦП сообщения размером 2048 байт алгоритмом Дж. Брука на кластерной ВС с иерархической организацией:

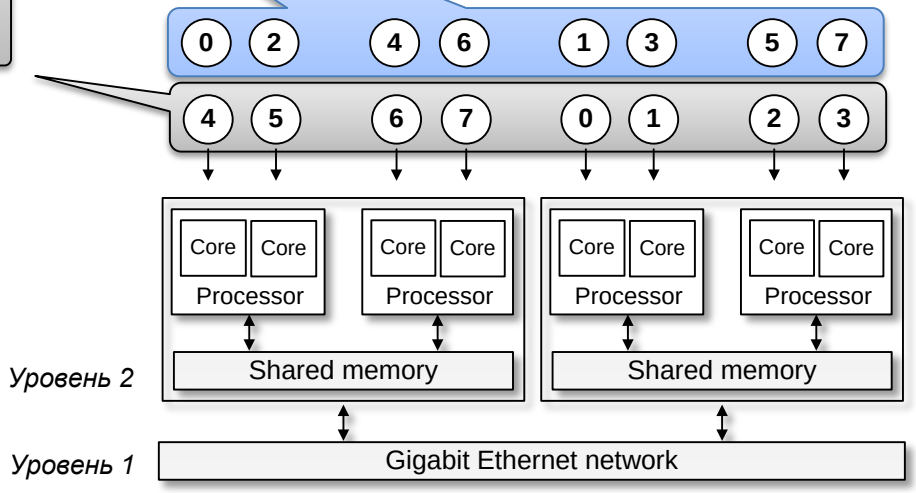
Время реализация ТЦП с учетом иерархической организации распределенной ВС – 324 мкс.

Ускорение в 1.75 раз

Время реализация ТЦП с распределением ветвей стандартными средствами библиотеки MPI (MPICH2 1.2.1) – 567 мкс.



Граф алгоритма Дж. Брука



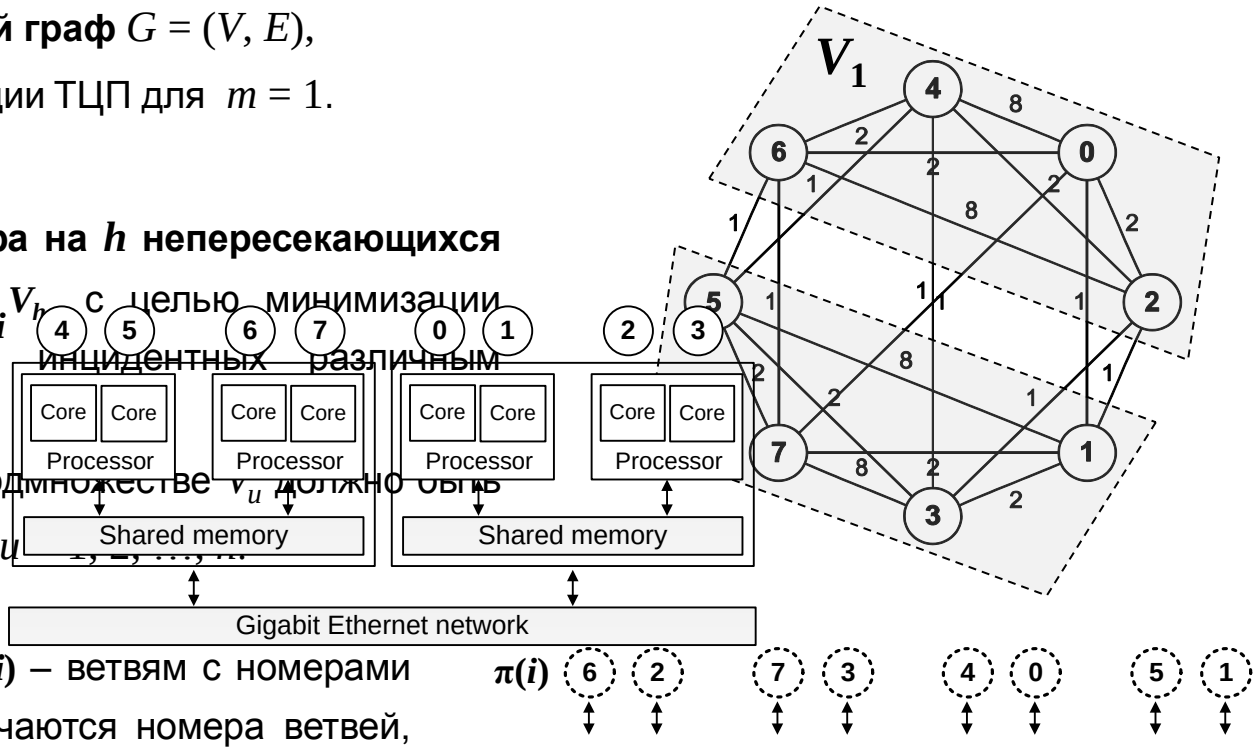
Метод оптимизации трансляционно-циклической передачи данных в иерархических распределённых ВС

1. **Формируется взвешенный граф** $G = (V, E)$, $|V| = n$ алгоритма реализации ТЦП для $m = 1$.

2. **Строится разбиение графа на h непересекающихся подмножеств $V_1, V_2, \ldots, V_h$** с целью минимизации суммарного веса рёбер, инцидентных различным подмножествам разбиения.

   Количество элементов в подмножестве V_u должно быть равно заданному числу s_u, $u = \overline{1, h}$.

3. **Строится отображение $\pi(i)$** – ветвям с номерами из подмножества V_u назначаются номера ветвей, распределенных на элементарные машины вычислительного узла q_u.



Алгоритм recursive doubling exch.

1. Ветвь i передает свое сообщение a_i ветви $\pi^{-1}(i)$, принимает от ветви $\pi(i)$ сообщение и делает его начальным.
2. На шаге $k = 0, 1, \ldots, \log_2 n - 1$ ветви i и $\pi^{-1}(i \oplus 2^k)$ обмениваются ранее принятыми 2^k сообщениями.

Алгоритм recursive doubling reorder

1. Ветвь i сдвигает свое сообщение a_i из позиции i в позицию $\pi(i)$.
2. На шаге $k = 0, 1, \ldots, \log_2 n - 1$ ветви i и $\pi^{-1}(i \oplus 2^k)$ обмениваются ранее принятыми 2^k сообщениями.
3. Сообщение из позиции $j = 0, 1, \ldots, n - 1$ переносится в позицию $\pi^{-1}(j)$.

Алгоритм Bruck exch.

1. Ветвь i передает свое сообщение a_i ветви $\pi^{-1}(i)$, принимает от ветви $\pi(i)$ сообщение и делает его начальным.
2. На шаге $k = 0, 1, \ldots, \lceil \log_2 n \rceil - 1$ ветвь i передает все принятые сообщения ветви $\pi^{-1}((i' - 2^k + n) \bmod n)$ и принимает сообщения от ветви $\pi^{-1}((i' + 2^k) \bmod n)$, где $i' = \pi(i)$.
3. Каждая ветвь циклически сдвигает сообщения вниз на i' позиций.

Алгоритм Bruck reorder

1. На шаге $k = 0, 1, \ldots, \lceil \log_2 n \rceil - 1$ ветвь i передает все принятые сообщения ветви $\pi^{-1}((i' - 2^k + n) \bmod n)$ и принимает сообщения от ветви $\pi^{-1}((i' + 2^k) \bmod n)$, где $i' = \pi(i)$.
2. Сообщение в позиции $j = 0, 1, \ldots, n - 1$ переставляется в позицию $\pi^{-1}((j + i') \bmod n)$.

М.Врубель. Сирень