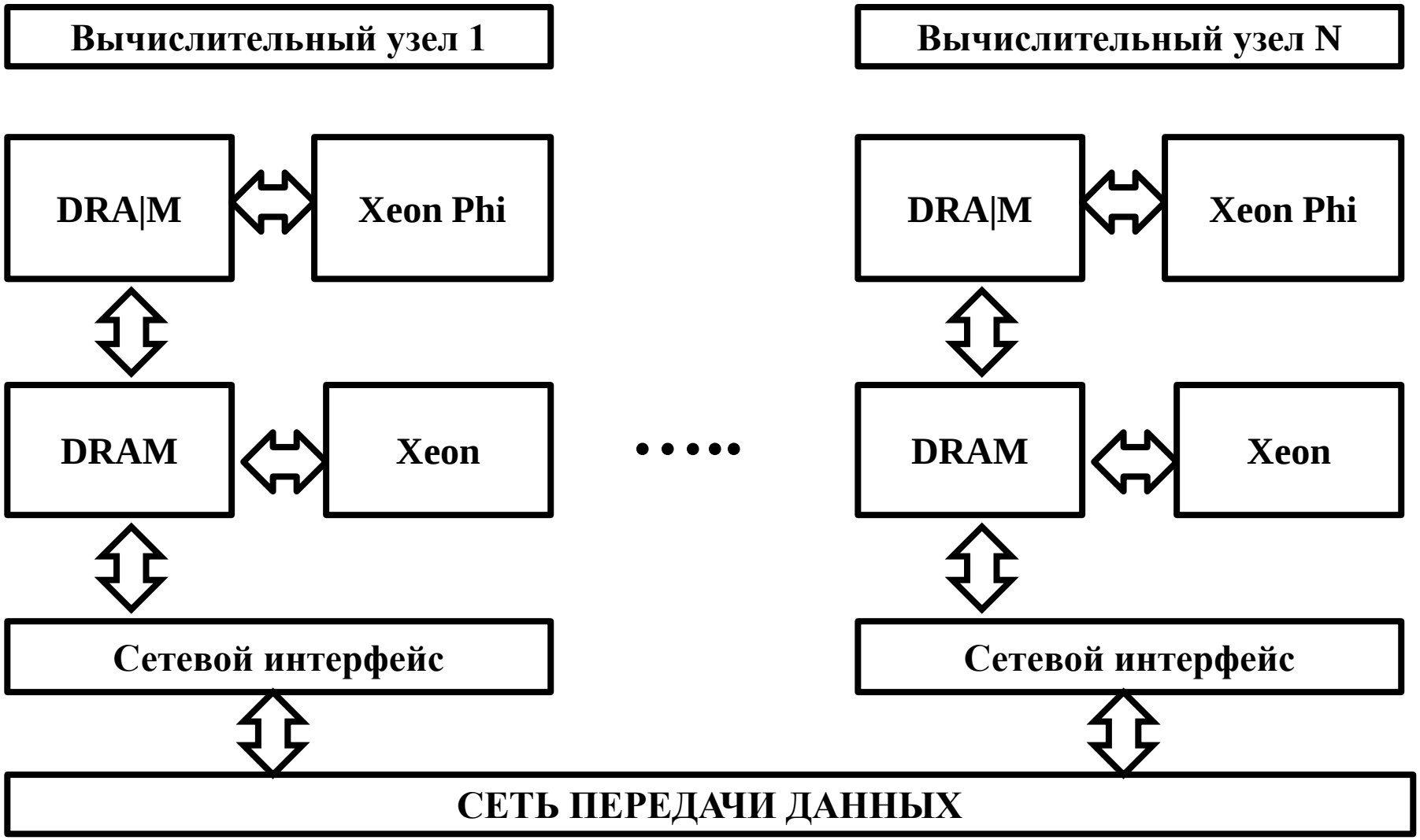
Распределение вычислительной нагрузки в гетерогенных вычислительных сетях

Singularis Lab, Ltd.

Volgograd State Technical University

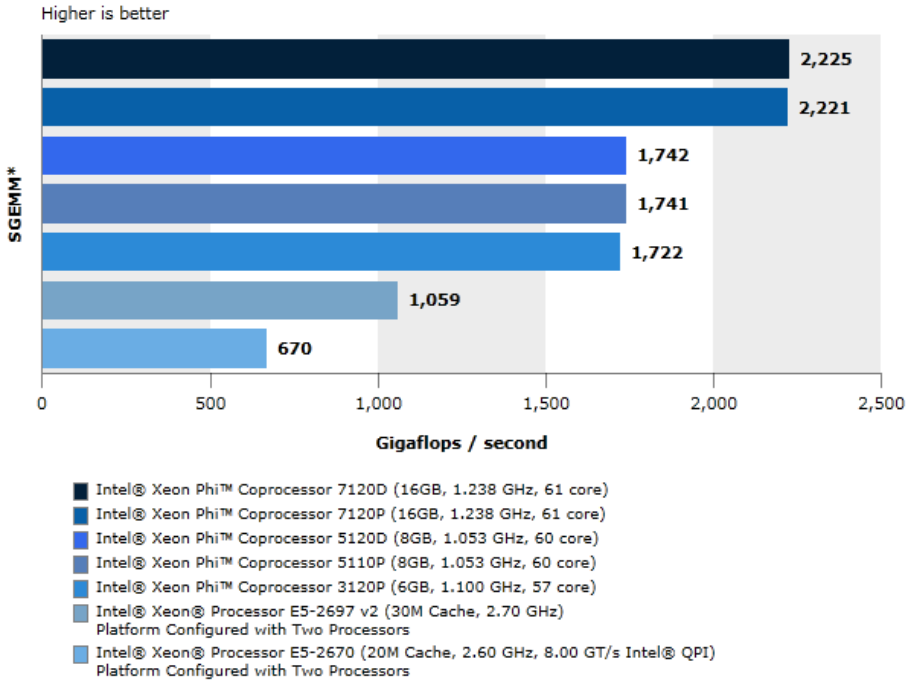
Кластерная ВС на базе узлов с Intel Xeon Phi

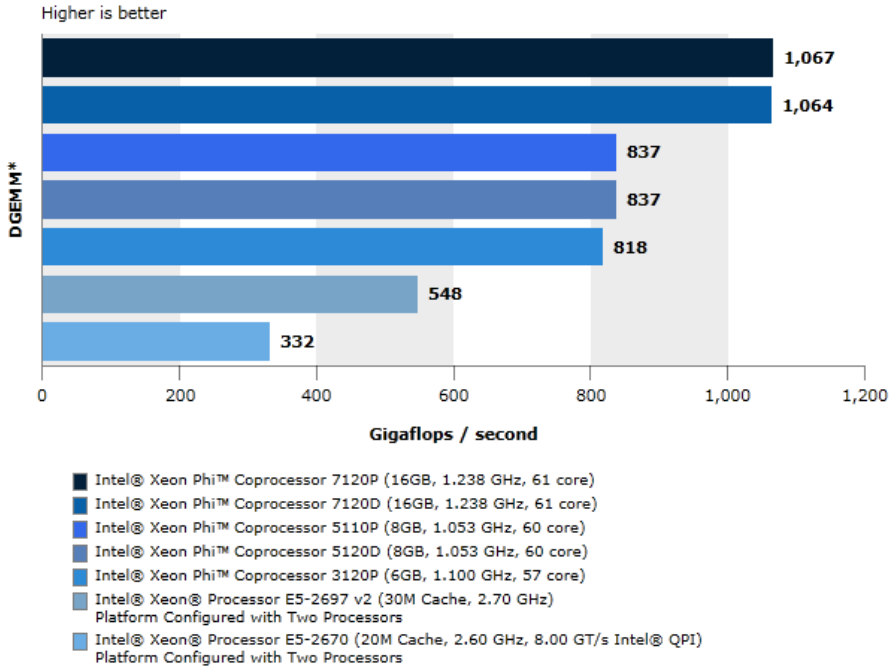


Производительность процессоров

	Intel Xeon processor 5600 series	Intel Xeon processor Sandy Bridge series	Intel Xeon Phi co-processor
Ядер	6	8	61
Потоков	12	16	244
Частота ядра, GHz	3.6	3.6	1.1
Пиковая производительность GFlops	~80	~80	~1000

Производительность процессоров





Производительность вычислительных систем

Загруженность устройства, это величина равная отношению количества реально выполненных операций к максимально возможному количеству операций за время выполнения программы.

$$p = \frac{r}{\pi}$$

Если ВС состоит из **S** вычислительных элементов, имеющих производительности $\pi_1, \ldots \pi_s$, работающих с загруженностями $p_1, \ldots p_s$, тогда загруженность системы:

$$p = \sum_{i=1}^{S} \alpha_i p_i; \quad \alpha_i = \frac{\pi_i}{\sum_{j=1}^{S} \pi_j}; \quad \sum_{i=1}^{S} \alpha_i = 1, \quad \alpha \geq 0, \quad 1 \leq i \leq S$$

Где α_i весовой коэффициент характеризующий вклад вычислительного элемента (сопроцессора) в пиковую производительность ВС. Для того чтобы загруженность ВС равнялась 1, необходимо чтобы равнялись 1 загруженности каждого вычислительного элемента ВС

Производительность вычислительных систем

Вычислительный кластер содержит более одного вида вычислительных устройств (процессоров/сопроцессоров):

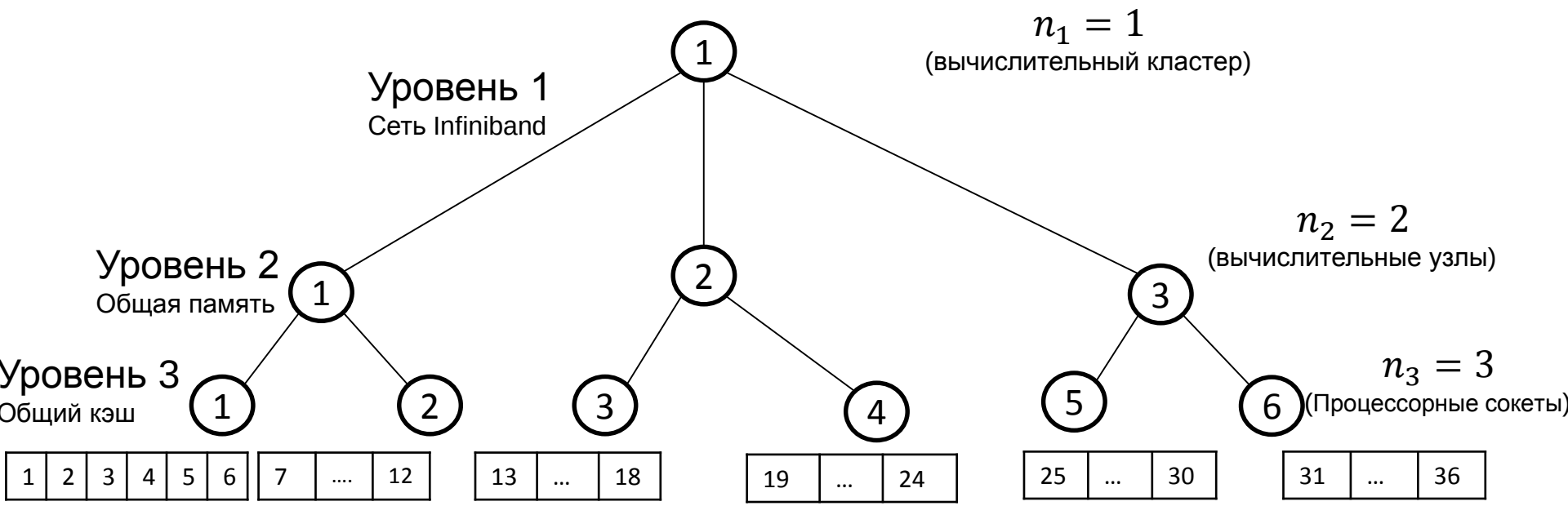
- Intel(R) Xeon(R) X5680
- Intel(R) Xeon(R) Phi

Оптимальная параллельная программа должна использовать параллелизм не уровне различных процессоров. Для достижения максимальной производительности на ВС, количество операций **w** которые необходимо выполнить для выполнения программы необходимо пропорционально распределить между **S** вычислительными элементами:

$$w_i = w\alpha_i \qquad i = 1,2,\dots,S$$

Модель ВС с иерархической организацией

- ВС укомплектована $\boldsymbol{N}$ однородными процессорами
- Коммуникационная среда системы дерево $\boldsymbol{L}$ уровней
- Каждый уровень образован отдельным видом структурных элементов
- На уровне l размещено n_l элементов



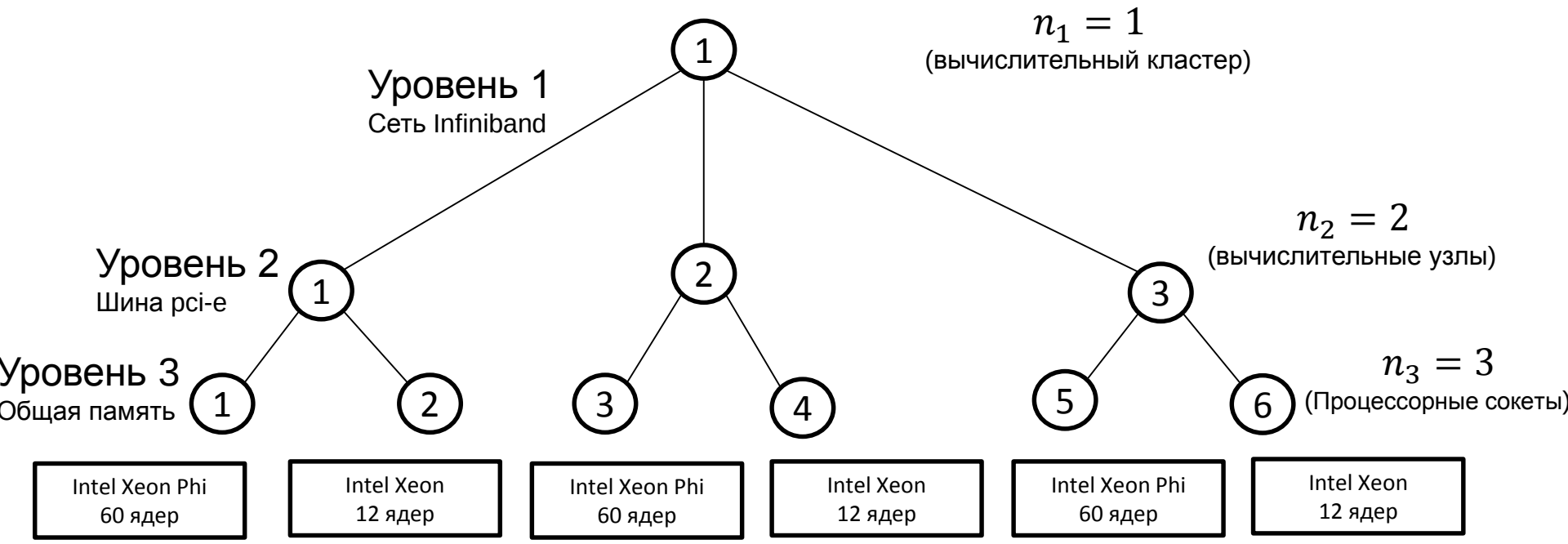
Intel Xeon
6 ядер
12 потоков

Однородные процессорные ядра

Однородные вычислительные узлы

Модель ВС с иерархической организацией

- ВС укомплектована $\boldsymbol{N}$ неоднородными процессорами
- Коммуникационная среда системы дерево $\boldsymbol{L}$ уровней
- Каждый уровень образован отдельным видом структурных элементов
- На уровне l размещено n_l элементов



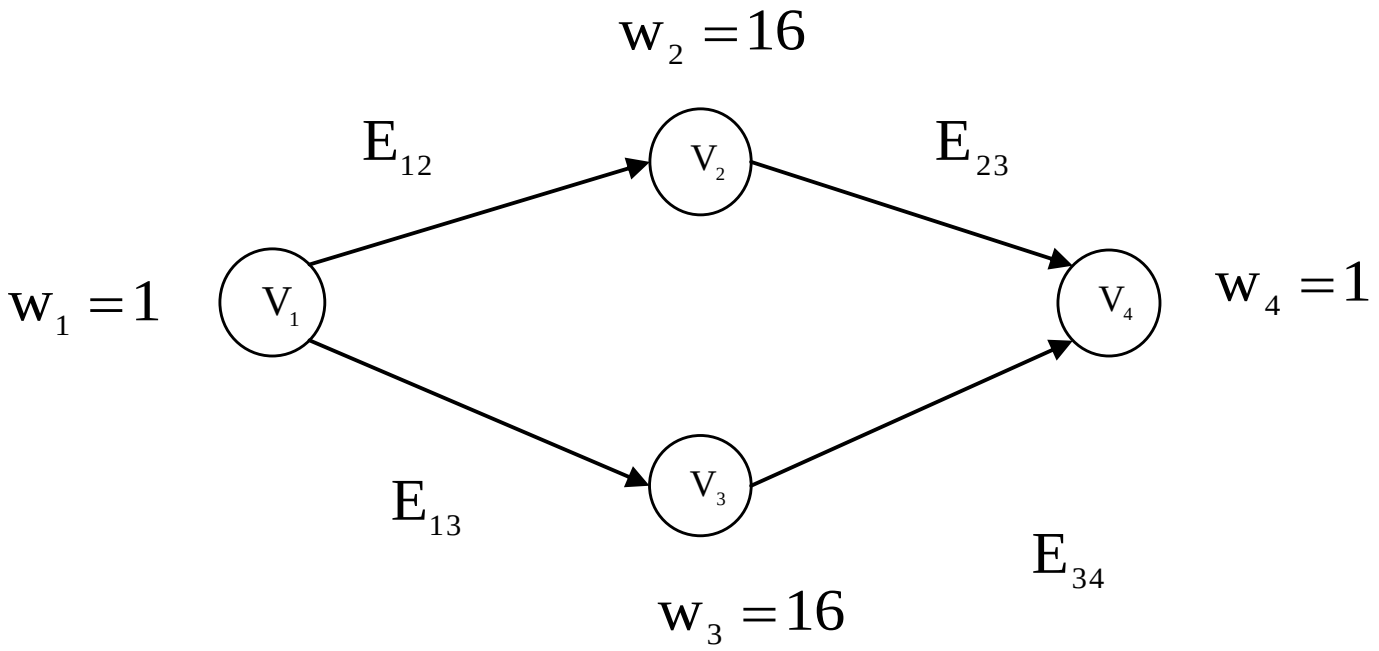
Процессорные ядра разного типа

Однородные вычислительные узлы

Формализация программ

Пусть программа представлена ациклическим направленным (**DAG** - *directed acyclic graph)* или не ориентированным графом $G(V, E)$, где $V = \{1,2, \dots, M\}$ множество ветвей параллельной программы, а $E = V \times V$ – множество информационно логических связей между ее ветвями.

Каждой вершине графа поставлен в соответствие весовой коэффициент w_i характеризующий вычислительную сложность(время).



Вложение параллельных программ в ВС

Пусть ВС включает N процессоров, множество которых обозначим через $P = \{1,2, \dots, N\}$, и задано множество $\mathrm{T} = \{1,2, \dots M\}$ ветвей параллельной программы которые необходимо назначить на процессоры. Пусть ВС состоит из H вычислительных узлов, тогда разобьём множество P на семейства подмножеств, объединяющих процессоры которыми укомплектован вычислительный узел или сопроцессор:

$$\bigcap_{h=1}^{H} P_h = \emptyset$$

Вложение параллельных программ в ВС

Задача оптимального вложения параллельной программы в ВС заключается в отыскании инъективной функции $f: T \to P$, ставящей в соответствие ветвям параллельной программы процессоры системы.

Требуется найти a_{ijk} :

$$A = \{a_{ijk}: i \in T, j \in H, k \in N_j\}$$

$$a_{ijk} = \begin{cases} 1, & \text{если } f(i,j) = k \\ 0, & \text{иначе} \end{cases}$$

$$a_{ijk} = 1, i \in T, j \in H, k \in N_j$$

Вложение параллельных программ в ВС

На множестве T, и семействах множеств P_h, заданы весовые функции:

- $x_{ih} = x(i,h)$ - функция характеризующая время выполнения ветви i на любом процессоре общего назначения вычислительного узла h
- $c_{ij}^{kq} = c(i,j,k,q)$ - время необходимое для осуществления взаимодействия между ветвями i и j распределенными между процессорами хостов k и q соответственно.

Цель вложения ветвей параллельной программы в ВС минимизация суммы времен выполнения арифметических операций и коммуникационного взаимодействия:

$$\sum_{i=1}^{M}\sum_{h=1}^{H}\sum_{k=1}^{N}(a_{ihk}\cdot x_{ih}) + \sum_{(i,j)\in E}\sum_{(p,q)\in H}\sum_{(h\in N_{1p},k\in N_{1q})}(a_{iph}\cdot a_{jqk}\cdot c_{ij}^{pq})$$

Методы решения

- Task Map Round Robin, Task Map Linear
  - Реализованные в утилите mpiexec
- Точные методы
  - Венгерский алгоритм
  - Симплекс метод
- Эвристические методы
  - Многоуровневое методы разбиения графа задачи на слабо связанные подмножества
- Методы динамического программирования