

MPI

Режимы передачи данных:

MPI_Send – стандартный режим.

MPI_Ssend – синхронный режим (ждет ответ о приёме сообщения)

MPI_Rsend – режим передачи по готовности (сработает только при запущенном MPI_Recv)

MPI_Bsend – буферизированный режим (использует дополнительный буфер для отправки)

MPI

Блокирующие функции приостанавливают выполнение процессов до момента завершения работы. (MPI_*Send, MPI_Recv)

Неблокирующие функции обмена данными выполняются без блокировки. (MPI_I*send, MPI_Irecv)

```
int MPI_I* (... , MPI_Request *request);
```

Проверка состояния неблокирующей функции:

```
int MPI_Test( MPI_Request *request, int *flag,  
MPI_status *status)
```

– **request** – дескриптор операции, определенный при вызове неблокирующей функции

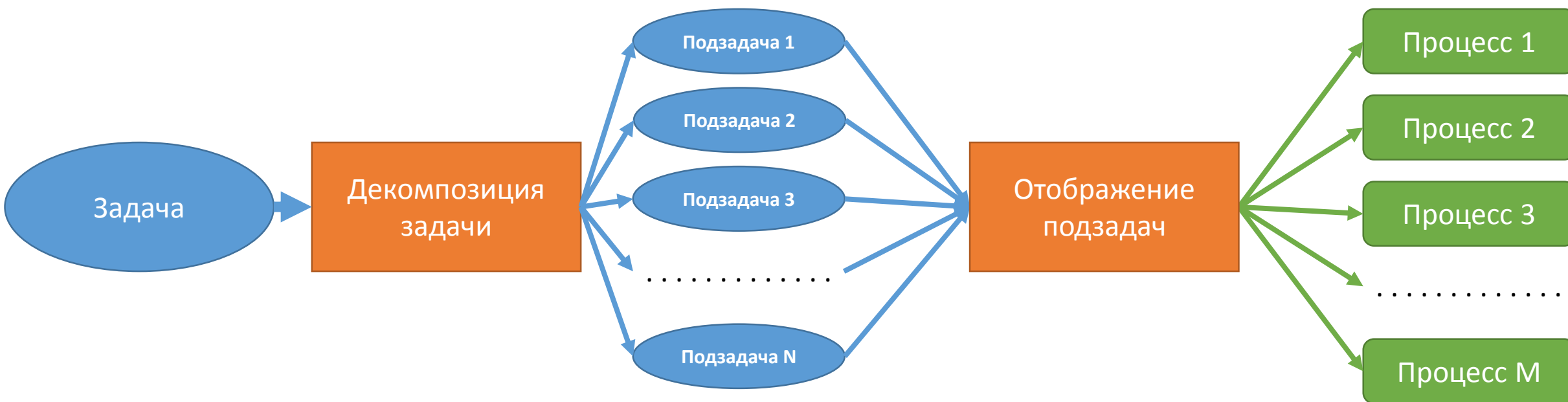
– **flag** – результат проверки (=true, если операция завершена)

– **status** – результат выполнения операции обмена (только для завершенной операции)

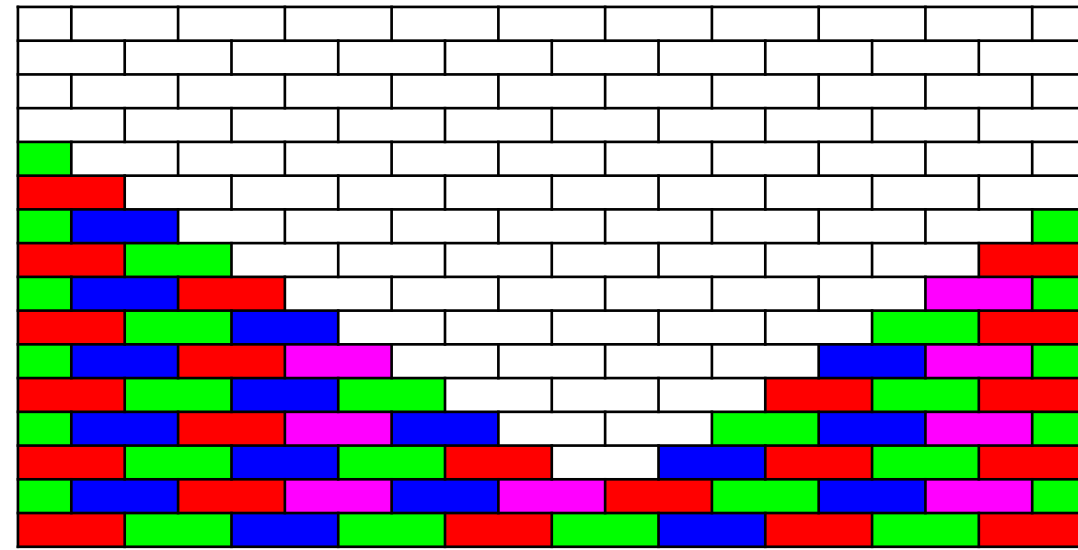
MPI

- MPI_Testall** – проверка завершения всех перечисленных операций обмена
- MPI_Waitall** – ожидание завершения всех операций обмена
- MPI_Testany** – проверка завершения хотя бы одной из перечисленных операций обмена
- MPI_Waitany** – ожидание завершения любой из перечисленных операций обмена
- MPI_Testsome** – проверка завершения каждой из перечисленных операций обмена
- MPI_Waitsome** – ожидание завершения хотя бы одной из перечисленных операций обмена и оценка состояния по всем

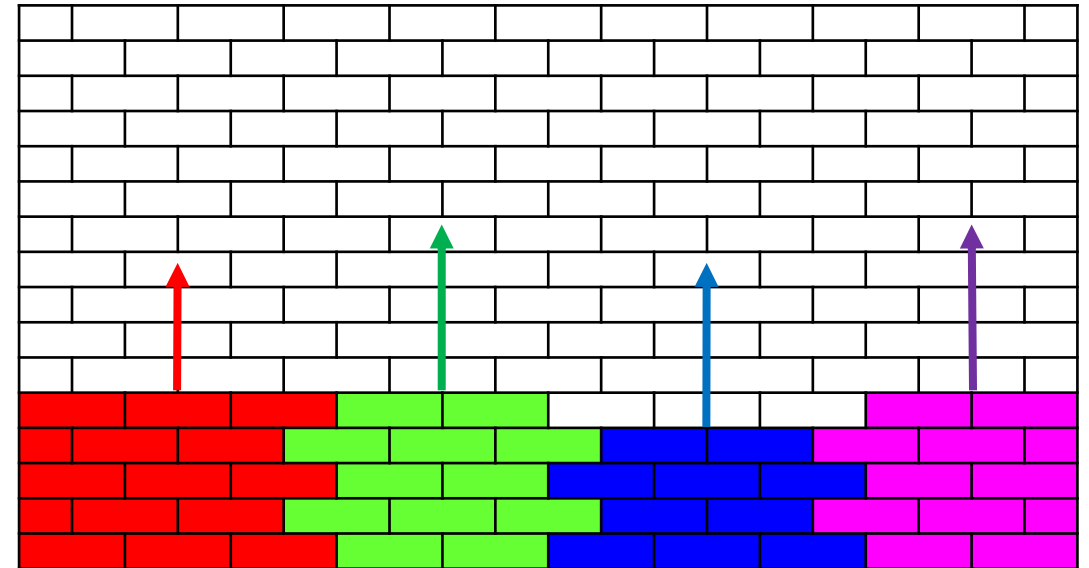
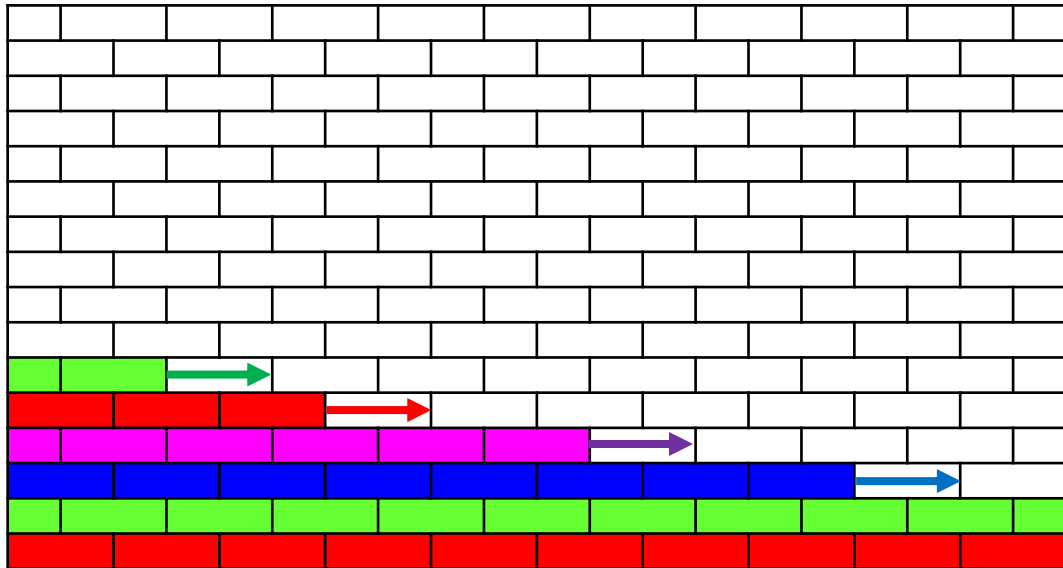
Балансировка нагрузки



Балансировка нагрузки



Стена Фокса (пример с лекций © М.В. Якобовского)



Балансировка нагрузки

Проблемы балансировки вычислительной нагрузки:

- структура распределенной задачи неоднородна
- структура вычислительного комплекса (например, кластера) неоднородна
- структура межузлового взаимодействия неоднородна

Балансировка нагрузки

Статическая балансировка

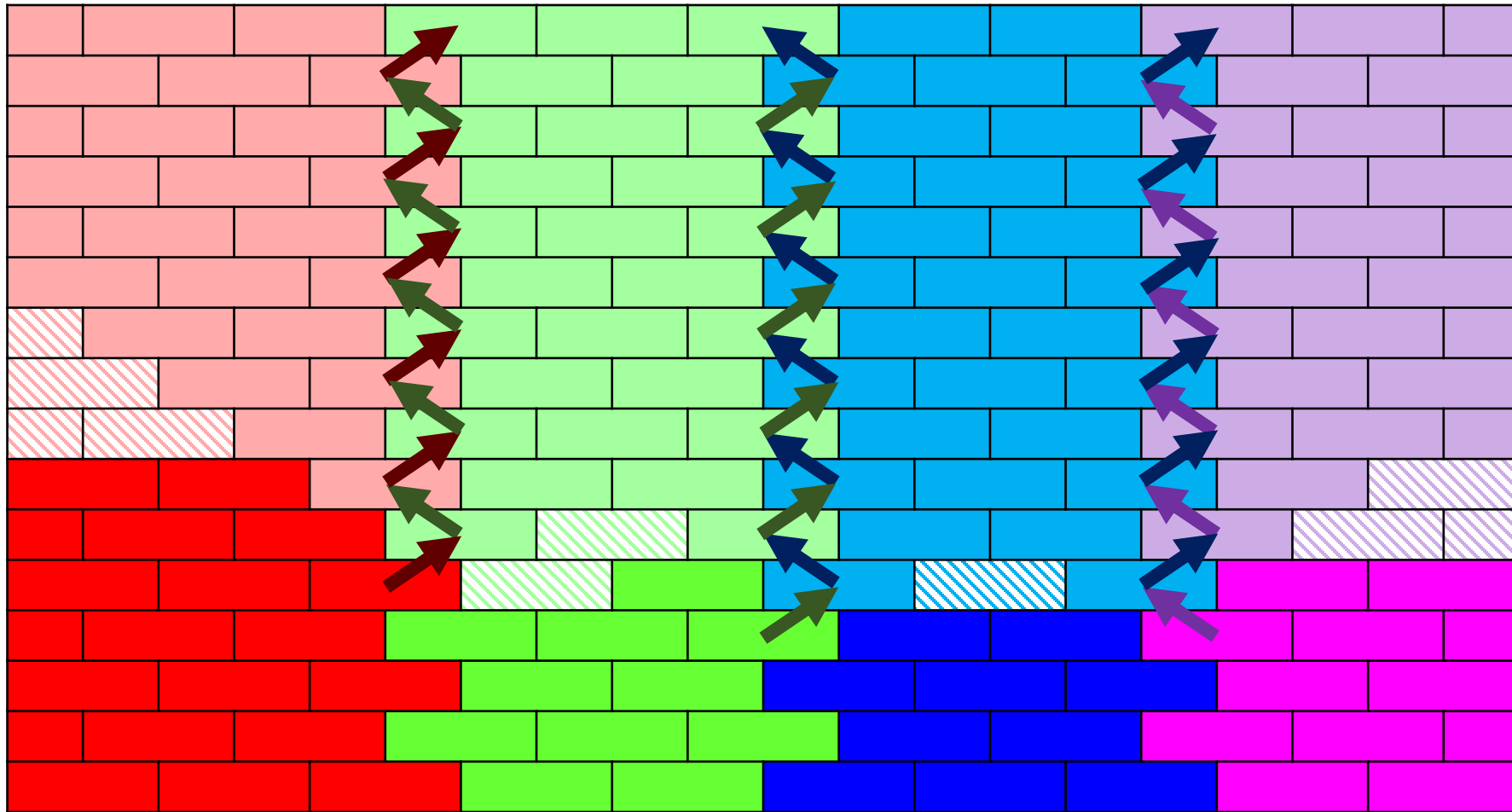
- отображение задач до начала выполнения задачи
- борьба с неоднородностями при помощи эвристик и опыта предыдущих запусков

Динамическая балансировка

- отображение задач происходит до и во время выполнения задачи
- борьба с неоднородностями при помощи постоянного (пере)распределения задач

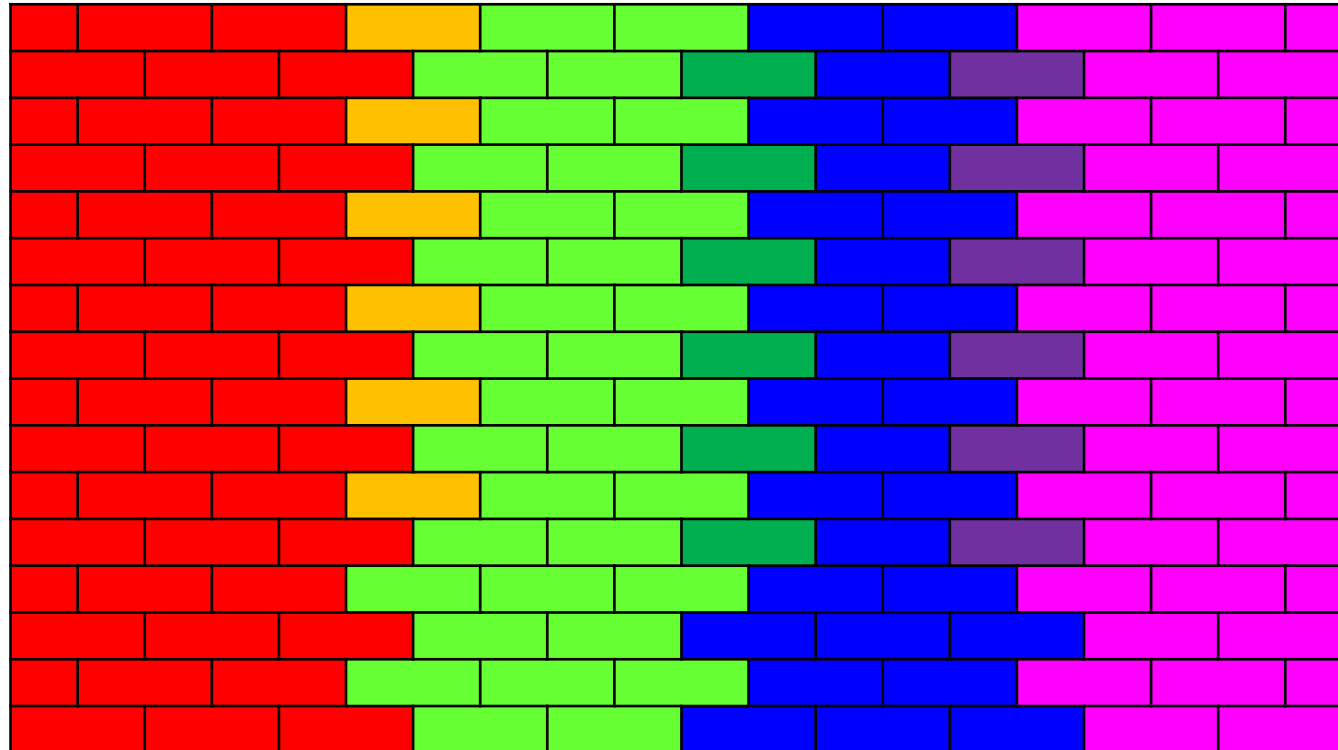
Балансировка нагрузки

Статическая балансировка

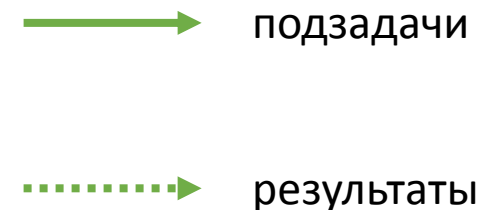


Балансировка нагрузки

Динамическая балансировка



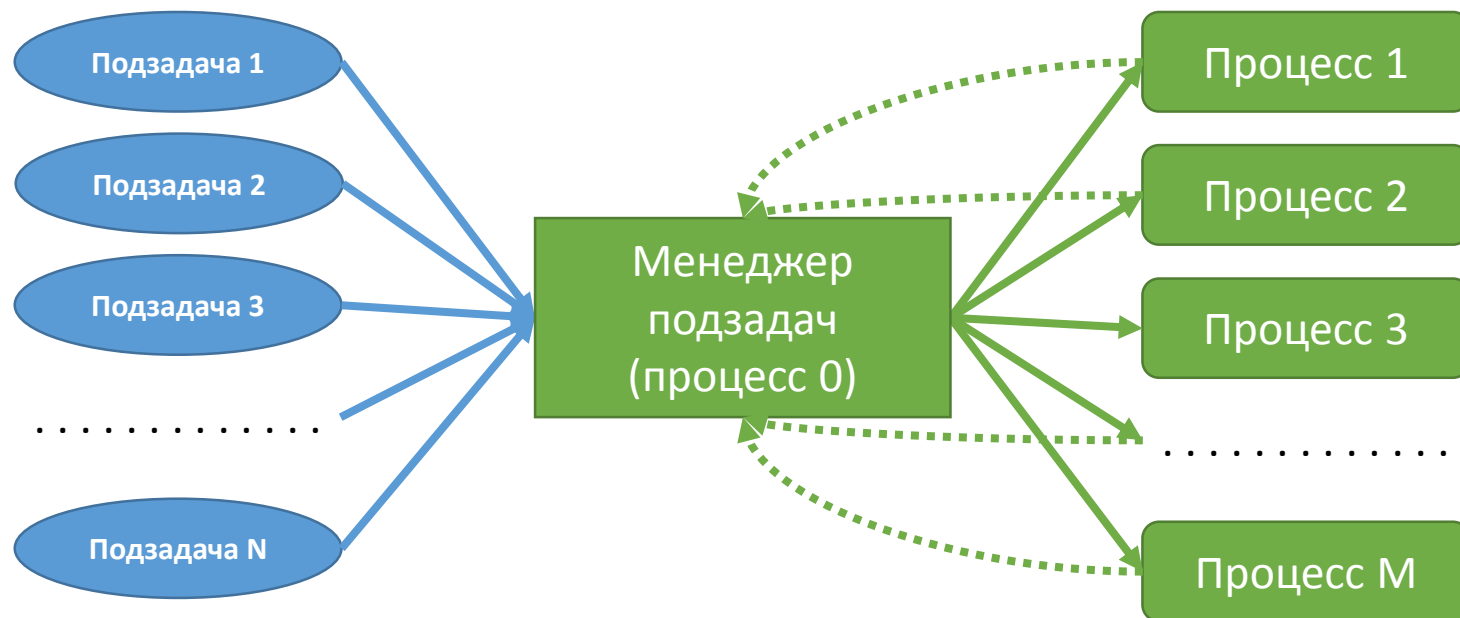
Балансировка нагрузки



Динамическая балансировка

RCL – стратегия переноса нагрузки:

- случайный алгоритм (random, R)
- алгоритм, основанный на коммуникациях (communication, C)
- алгоритм, основанный на вычислении нагрузки (load, L)



Балансировка нагрузки

Длинная арифметика (сложение)

+	21	43	76	54
	4	55	24	02
			0 ←	56
		1 ←	100	
	0 ←	99		
	25	99	00	56

Балансировка нагрузки

Спекулятивные вычисления

+	21	43	76	54	53	09	12	94	11	23	08	05
	4	55	24	45	85	75	25	41	54	25	08	97
			0 ← 99				1 ← 135				1 ← 102	
			1 ← 100				1 ← 136				1 ← 102	
		1 ← 100				0 ← 38				0 ← 17		
		1 ← 101				0 ← 38				0 ← 17		
	0 ← 99				0 ← 84				0 ← 48			
	0 ← 99				0 ← 84				0 ← 48			
	25				138				65			
	25				138				65			
	25	99	00	99	38	84	38	35	65	48	17	02
	25	99	01	00	38	84	38	36	65	48	17	02

Балансировка нагрузки

Суммирование старших разрядов

+	21	43	76	54	53	09	12	94	11	23	08	05
	4	55	24	45	85	75	25	41	54	25	08	97
			0← 99	138			1← 135	65			1← 102	
			1← 100	137			1← 136	66			1← 102	
		1← 100				0← 38				0← 17		
		1← 101				0← 38				0← 17		
	0← 99				0← 84				0← 48			
	0← 99				0← 84				0← 48			
	25				138				65			
	25				138				65			
	25	99	00	99	38	84	38	35	65	48	17	02
	25	99	01	00	38	84	38	36	65	48	17	02

Балансировка нагрузки

Динамическая балансировка

Исполнитель					1	3	3 +	2	1 +
Первое число	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	24 разряда
Второе число	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	32 разряда	24 разряда
Результат (0)							32 разряда		24 разряда
Результат (1)							32 разряда		24 разряда
Остаток (0)							1 разряд		1 разряд
Остаток (1)							1 разряд		1 разряд