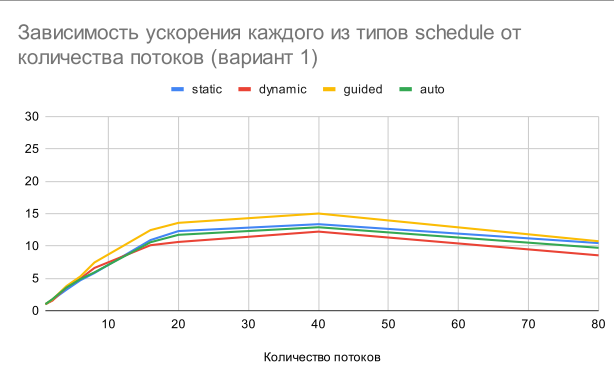


CPU model: Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz  
Architecture: x86\_64  
Threads/core: 2  
Cores/socket: 20  
Sockets: 2  
Server name: ProLiant XL270d Gen10  
Nodes: 2  
Node 0 size: 385636 MB  
Node 1 size: 387008 MB  
OS: Ubuntu 22.04.3 LTS

omp schedule		Количество потоков																	
		1		2		4		6		8		16		20		40		80	
		T1	T2	S2	T4	S4	T6	S6	T8	S8	T16	S16	T20	S20	T40	S40	T80	S80	
static		42,309	25,295	1,673	13,097	3,23	9,011	4,695	7,282	5,81	3,883	10,896	3,443	12,288	3,169	13,351	4,055	10,434	
dynamic		42,436	27,335	1,552	11,792	3,599	8,55	4,963	6,434	6,596	4,2	10,104	4	10,609	3,477	12,205	4,969	8,54	
guided		43,96	25,67	1,713	11,585	3,795	8,3	5,296	5,91	7,438	3,535	12,436	3,242	13,56	2,931	14,998	4,102	10,717	
auto		42,712	24,059	1,775	11,984	3,564	8,771	4,87	7,224	5,913	4,044	10,562	3,647	11,712	3,317	12,877	4,402	9,703	
single region		45,397	12,907	3,517	7,98	5,689	4,607	9,854	3,782	12,003	1,827	24,848	1,727	26,287	1,496	30,346	2,125	21,363	



Вывод: при решении данной задачи наиболее эффективно себя показал тип schedule guided, при котором итерации распределяются среди потоков в зависимости от их нагрузки. Однако второй вариант программы с предварительным созданием параллельной секции и выполнением всего алгоритма в ней показал себя в 1.5 - 2 раза эффективнее, поэтому рекомендуется использовать именно его. До 16 потоков скорость растёт практически линейно, после 16 потоков скорость роста значительно падает. Оптимальное количество потоков: 16-20. Использовать более 40 потоков не рекомендуется, скорость выполнения программы падает из-за накладных расходов для переключения контекста и/или ограничений шины процессора.