Work 17 Кириленко Константин 351



void ParallelBitReversing(complex<double>\* inputSignal,

complex<double>\* outputSignal, int size, int NProc, int ProcId) {

int bitsCount = 0;

//bitsCount = log2(size)

for (int tmp\_size = size; tmp\_size > 1; tmp\_size /= 2, bitsCount++);

//ind - index in input array

//revInd - correspondent to ind index in output array

int n1 = size / NProc;

int n2 = (ProcId + 1) \* n1;

if (NProc == ProcId + 1) {

n2 = size;

}

int st = ProcId \* n1;

complex<double>\* outputSignal1 = new complex<double>[size];

for (int i = 0; i < size; i++) outputSignal1[i] = 0;

//#pragma omp parallel for

for (int ind = 0; ind < n2; ind++)

{

//if (i < )

int mask = 1 << (bitsCount - 1);

int revInd = 0;

for (int i = 0; i < bitsCount; i++) //bit-reversing

{

bool val = ind & mask;

revInd |= val << i;

mask = mask >> 1;

}

outputSignal1[revInd] = inputSignal[ind];

}

MPI\_Reduce(&(outputSignal1[0]), &(outputSignal[0]), size, MPI\_DOUBLE\_COMPLEX, MPI\_SUM, 0, MPI\_COMM\_WORLD);

MPI\_Bcast(&(outputSignal[0]), size, MPI\_DOUBLE\_COMPLEX, 0, MPI\_COMM\_WORLD);

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Размер входного сигнала | Мин. время работы последовательного приложения (сек) | Мин. время работы параллельного приложения (сек) | Ускорение |
| 1 | 32768 | 0.095 | 0.065 | 1,46 |
| 2 | 65536 | 0.204 | 0.129 | 1,58 |
| 3 | 131072 | 0.416 | 0.261 | 1,59 |
| 4 | 262144 | 0.938 | 0.646 | 1,45 |
| 5 | 524288 | 2.293 | 1.232 | 1,86 |

Использование параллельной версии функции BitReversing не дает заметного прироста.