LEVELS OF PARALLELISATION

Gritsay I.P.
Tahtamyshev S.S.
Don State Technical University, Rostov-on-Don.

Currently, we see how rapidly developing computer technology. For example, the world's fastest computer is considered to be the Chinese Tianhe-2, or "Milky Way" with capacity 33,86 PFlop, whereas in 2010, is considered to be the most powerful computer complex Oak Ridge National Laboratory (United States) - 2,331 PFlop. However, in many cases, these great computers are not fully loaded and used inefficiently. We have to solve a lot of problems that need to be carried out simultaneously, but the misuse of resources multiprocessor machines shows that execution of such work takes time. One solution to this problem is the method of parallelization. High performance is achieved by deep parallelization of both the program and at the structural level.

Parallelization method has been successfully applied, when we are dealing with a complex and time-consuming task. This method can significantly increase the speed of execution of the task. Solution of the problem can be parallelized on several levels.

Software parallelization is carried out at the level of tasks (multitasking) and at the level of operations and micro-operations (parallelization and execution in time alignment). Structural parallelization is multiprocessor and the availability of specialized units performs arithmetic operations, and other devices that support software parallelization.

Parallelization at the task level is the easiest and yet most effective. This parallelization is possible in cases where the problem is solved by the independent subtasks, each of which can be solved separately. Parallelization in the task level, we demonstrate the operating system which runs programs on a multicore machine on different cores. For example, we can in first program to edit video in second - to surf the Internet, in the third - to download the files and the operating system will be able to easily organize their parallel operation. But if we are dealing with a homogeneous problem, this kind of parallelization is not applicable. The operating system can’t

accelerate a program that uses only one processor, no matter how many cores it would be available at the same time.

To parallelize the homogeneous problems, you need to use parallelism at the data level. Parallelism is to apply the same operation to a plurality of data elements. Parallelism data shows archiver, which used for packaging multiple processor cores. The data are divided into blocks, which are treated (packed) in a uniform way on different nodes. Data parallel model is extremely convenient, as it allows to upload each core, by selecting his specific set of cells. Accounts area is divided into geometric objects, such as parallelepipeds, and cells that are included in this area, given to the processing of a specific kernel. This type of parallelism is widely used in solving the problems of numerical modeling.

«The next level is the parallelization of individual procedures and algorithms. This could include the parallel sorting algorithms, matrix multiplication, and the solution of a system of linear equations. At this level of abstraction is convenient to use a parallel programming technology such as OpenMP. »[4.15]

Parallelism at the level of instruction - the lowest level of parallelism at the level of parallel processing of multiple processor instructions. On the same level is batch processing of multiple data elements a command processor. This is the technology MMX, SSE, SSE2, and other. This type of parallelism is sometimes singled out in an even deeper level parallelization - bits level parallelism. The program is a stream of instructions executed by the processor. You can change the order of the instructions, distribute them to groups, which will be executed in parallel without changing the result of the work of the program. To implement this type of parallelism in microprocessors using multiple instruction pipelines, such as the prediction technology instruction, register renaming.

Also, when parallelizing most important - is to organize the work with memory because processors are much faster than memory and processor communicate with memory takes a lot of time. This will give us an increase in productivity and time to wait for the tasks.

«When using this method, the programmer can use the following tools:

- **OpenMP** – a standard application interface for parallel systems with shared memory.
- **POSIX Threads** – standard implementation of solving flows.
- **Windows API** – threaded applications to C ++.
- **PVM (Parallel Virtual Machine)** – it connects computers to the general computational resource.
- **MPI (Message Passing Interface)** – standard messaging systems between the parallel working processes. »[1.44]

Parallelization gives an important boost in performance and speed of execution of tasks at the correct and appropriate use of it.

References

1. Gritsay I.P. Increasing the speed of tasks solving by parallelization of tasks / Gritsay I.P., Brook A.S. // European journal of natural history. – 2015 – №5. – p. 44
2. P. Pacheco. Parallel Programming With MPI / Peter Pacheco – An introduction to Parallel Programming. – 2011 – 392 p.
3. В. Е. Карпов Введение в распараллеливание алгоритмов и программ / В. Е. Карпов – Математические основы и численные методы моделирования. – 2010 – С. 231 – 272
4. Program Parallelization Methods Implemented in Optimizing Compiler / V. Volkonskiy [et al.] – 2015 – С. 1 – 28
5. Знакомство с уровнями распараллеливания – Режим доступа: https://habrahabr.ru/company/intel/blog/80342/ (дата обращения 27. 04. 16)