**Q.) Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify objects, morphisms, overloading in functions (if any), and functional relations and any other.**

### Ans.Parallel programming

Parallel programming is a technique in which many computations are performed concurrently. Parallel computation divides a big task into smaller sub-tasks to be executed simultaneously. Parallelism can utilize multi-core processors in a single machine or multi-processors in a cluster of machines.

The parallel execution on a multi-core or a cluster can take many forms. It can be categorized into bit-parallelism, data parallelism, or task/function parallelism. The focus of bit-parallelism is to minimize the count of instructions to execute an operation. This can be done by increasing the processor word size. In data parallelism, the data is split into many pieces and is distributed to multiple cores or processes. All processors run the same code simultaneously but on different data piece. This is also known as single instruction multiple data (SIMD) approach. In contrast to data parallelism, task parallelism is a form of parallelism where multi-processors run different codes or tasks on the same piece of data simultaneously.

#### MPI

MPI is a standard interface that contains a set of libraries and routines to write a parallel program and distributes it over a cluster of machines or a multi-core processor. There are two basic components which are implemented in each MPI library. The first one deals with the compilation such as mpicc [[24](https://link.springer.com/article/10.1186/s13635-017-0062-7#CR24)]. It is a wrapper that links the MPI library and provides an easy operation to set the appropriate paths of both the library and the included files. mpicc also passes its argument to the C compiler which is required to run the program. The second tool is an agent which is responsible for executing the code in a distributed environment such as the mpirun or mpiexe.

#### Open multi-processing (OpenMP)

MPI is just a standard. It has several implementations such as Intel MPI, and OpenMP. OpenMP [[25](https://link.springer.com/article/10.1186/s13635-017-0062-7#CR25)] is a high-quality open-source implementation of the latest MPI standard with a superior performance compared to other implementations [[26](https://link.springer.com/article/10.1186/s13635-017-0062-7#CR26)]. It provides a set of application programming interfaces (APIs) that is easy to use. OpenMP supports shared memory in a multi-processor environment, which gives more flexibility to programmers to develop their distributed applications. It is comprised of a set of library routines and compiler directives that allow the master processor to distribute the data and tasks among the processing units.

Q.) Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfiability issues using modern algebra and/or relevant mathematical models.

Ans.

**Computational complexity:**

Study of how resources required to solve a problem (e.g., CPU time, memory) scale with the size of the problem-e.g., polynomial time algorithm (t ~ N log N, t ~ N2) vs. exponential time algorithm (t ~ 2N ,t ~ eN)

**Complexity classes:**

P: set of problems solvable in time polynomial in problem size on a deterministic sequential machine.

NP (non-deterministic polynomial): set of problems for which a solution can be verified in polynomial time.

NP-Complete: set of problems that are in NP, and are NP-hard (i.e., that every other problem in NP is reducible to it in polynomial time)

‣a polynomial time algorithm to solve one NP-complete problem would constitute a polynomial time algorithm to solve all of them

‣no known polynomial time algorithms for NP-complete problems

‣exponential runtimes consider worst case scenario; increasing interest in typical case complexity

Our project is comes in NP Complete problem. Since it includes Pattern Matching which comes in same.

**Q.)Develop the problem under consideration and justify feasibility using concepts of knowledge canvas and IDEA Matrix.**

|  |  |  |  |
| --- | --- | --- | --- |
| PROPOSITION  host based intrusion detection system for log files | AUDIENCE  company  people  student | VISION  hids  nids | GOALS  suspicious activity  detection |
| CHALLENGES  xml conversation  signature extraction  pattern matching  unknown attacks | STAKEHOLDERS  N/A | APPROACH  pattern matching algorithm | OUTCOMES  pop up alert message  suspicious activity  detection |
| SCOPE  signature based to  anomaly based detection | TEAM  Sachin,shatruanjay  Vikas,surjit  Yogesh | ACTIVITIES  extraction  comparison | METRICS  signature |