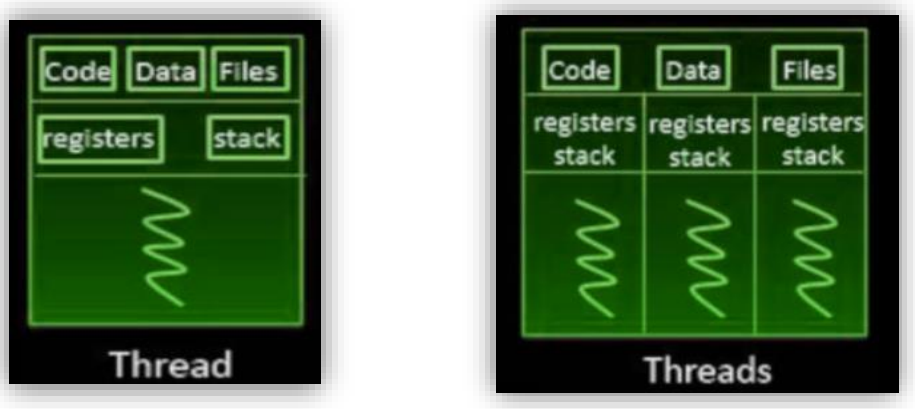
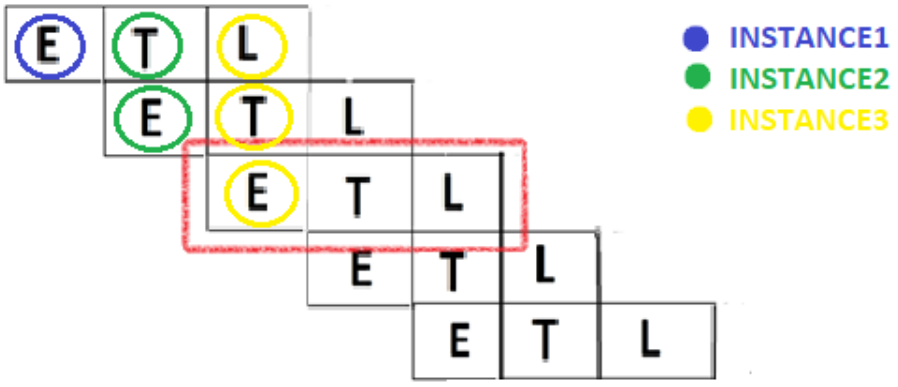
What is Multi-Threading?

Multithreading is a model of program execution that allows for multiple threads to be created within a process, executing independently but concurrently sharing process resources.



Depending on the hardware, threads can run fully parallel if they are distributed to their own CPU core.

What is ETL?



Exclusively, we implemented ETL aka Extract-Transform-Load using multi-threading where a large number of files are processed.

Besides entire processing of the file at once, ETL allows us to run extract, transform and load simultaneously using three threads independently.

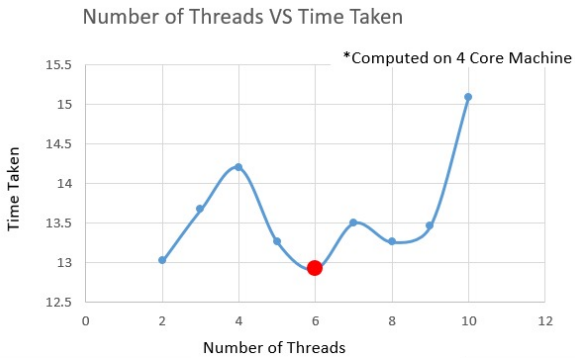
Above Diagram illustrates the sequence of threads executing at a single instance

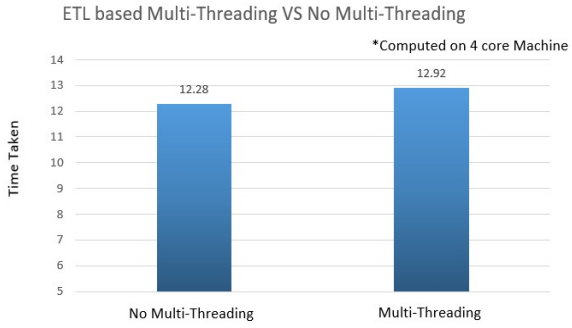
At instance 3,
1st file is loading...→ 2nd file is transforming…→ 3rd file is extracting

Summary of Time Taken:

File Size	No Multi-Threading	2 Threads	3 Threads	4 Threads	5 Threads	6 Threads	7 Threads	8 Threads	9 Threads	10 Threads
25MB	12.28	13.02	13.67	14.2	13.26	**12.92**	13.5	13.26	13.46	15.1
50MB	26.36	24.15	26.9	24.16	**23.2**	24.52	24.4	24.5	24.74	25.2
250MB	131.69	190.22	185.4	132.54	**127.14**	129.43	135.57	154.23	162.44	168.24
500MB	245.22	243.58	249.04	244.8	240.64	**238.64**	243.2	241.15	243.42	244.01
1GB	615.86	521.02	556.01	505.12	510.86	**486.22**	490.23	497.45	501.2	512.34

File-Size(25MB) – 5Lakh Entries:



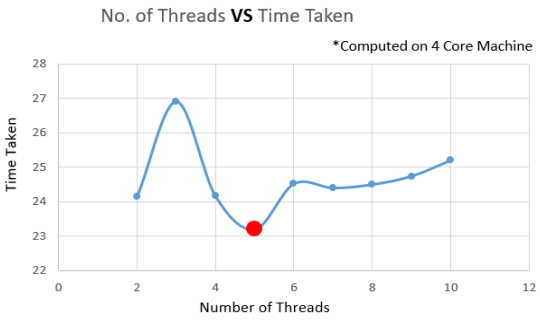


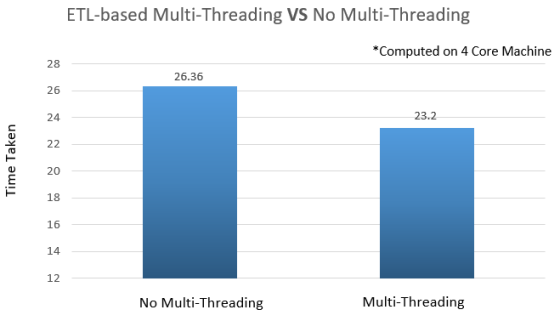
- **Observation:**

  No. of Threads=N*No. of Cores

- When the file size taken was 25MB then the optimum count of Treads comes out to be 6 i.e. the multiplying factor[N] was 1.5.
- When we performed transformation on the same file then time taken when there is no Multi-Threading comes about to be 12.28 seconds and time taken using Multi-Threading was 12.92 seconds.

File-Size(50MB) – 10Lakh Entries:



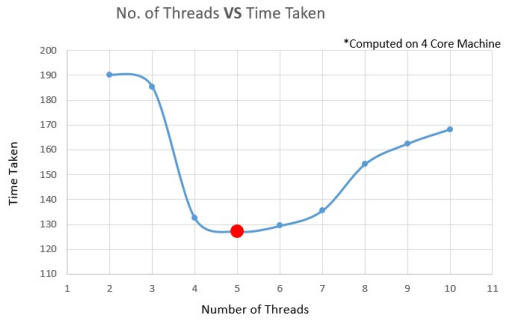


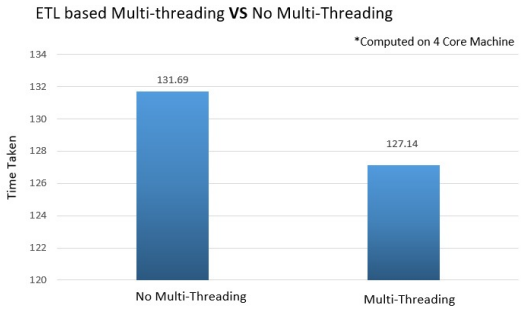
- **Observation:**

  No. of Threads=1.5*No. of Cores

- When the file size taken was 50MB then the optimum count of Treads comes out to be 5 i.e. the multiplying factor[N] was 1.4.
- When we performed transformation on the same file then time taken when there is no Multi-Threading comes about to be 26.36 seconds and time taken using Multi-Threading was 23.20 seconds.

File-Size(250MB) – 50Lakh Entries:



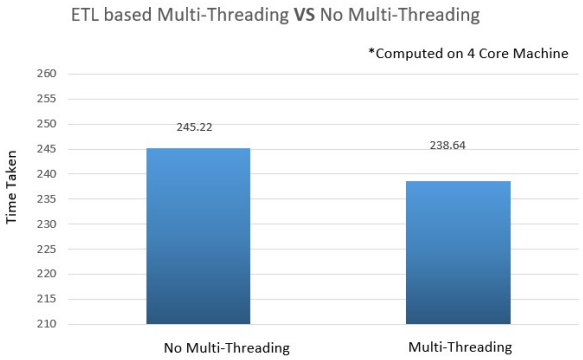


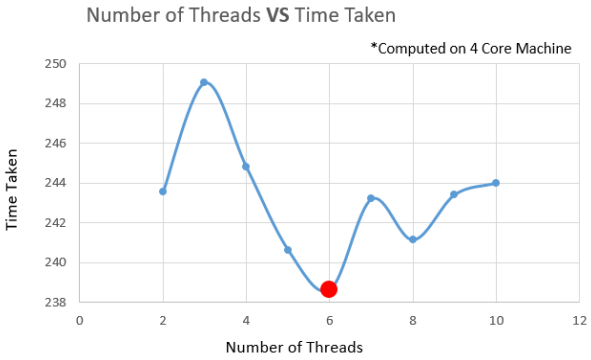
- **Observation:**

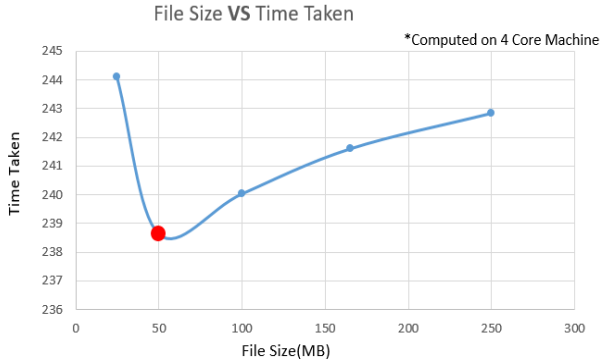
  No. of Threads=1.5*No. of Cores

- When the file size taken was 250MB then the optimum count of Treads comes out to be 5 i.e. the multiplying factor[N] was 1.4.
- When we performed transformation on the same file then time taken when there is no Multi-Threading comes about to be 131.69 seconds and time taken using Multi-Threading was 127.14 seconds.

File-Size(500MB) – 1Crore Entries:





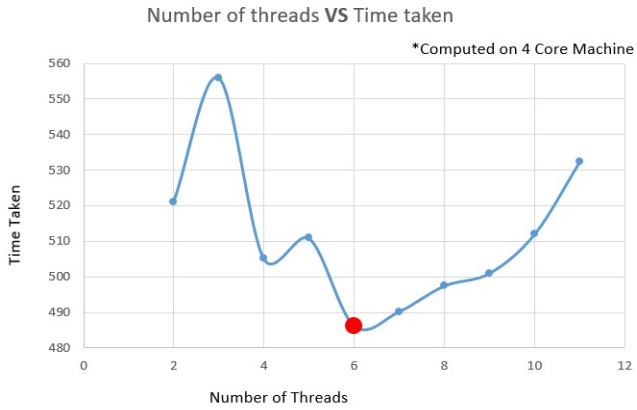


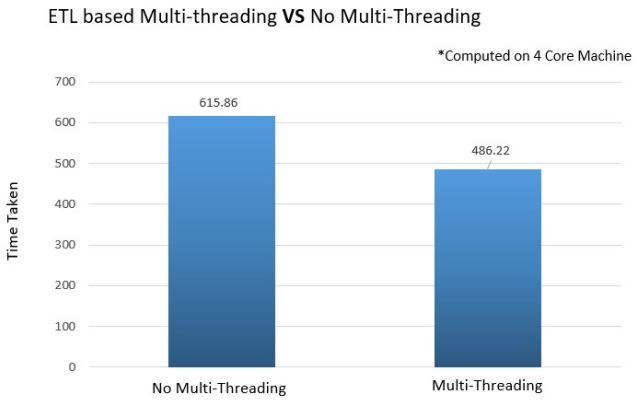
- **Observation:**

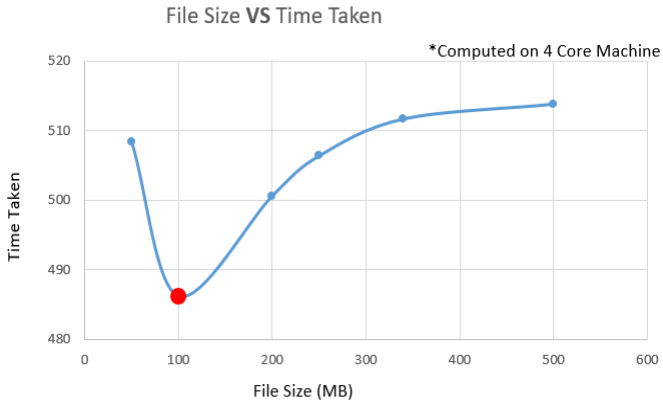
  No. of Threads=1.5*No. of Cores

- When the file size taken was 500MB then the optimum count of Treads comes out to be 6 i.e. the multiplying factor[N] was 1.5.
- When we performed transformation, we found Multi-Threading 2.78% Efficient as compared to traditional Computation.
- On Splitting this 500MB file into files of variable sizes, we found that time taken to completer the same task using Multi-Threading was minimum when the file sizes were 50MB each.

File-Size(1GB) – 2Crore Entries:







- **Observation:**
  No. of Threads=1.5*No. of Cores
- When the file size taken was 1GB then the optimum count of Treads comes out to be 6 i.e. the multiplying factor[N] was 1.5.
- When we performed transformation, we found Multi-Threading 26% Efficient as compared to traditional Computation.
- On Splitting this 1GB file into files of variable sizes, we found that time taken to complete the same task using Multi-Threading was minimum when the file size was 100MB each.

- **Conclusion:**
  - Factors favoring Multi-Threading includes:
    - Large Files.
    - Number of Cores.
    - Optimum Number of Threads.
  - Splitting the Larger files into Smaller chunks of Specific size to Multi-Threaded Program comes up with Efficient results.

Concluded By:

- Ayush Pushkarna
- Sukhbir Singh