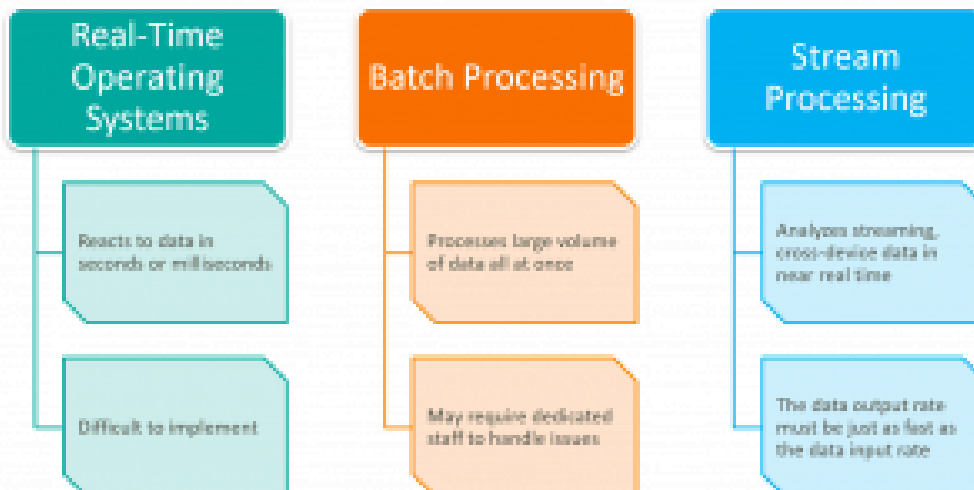


REAL TIME VS BATCH PROCESSING VS STREAM PROCESSING



With the constant rate of current innovations, developers can expect to analyze terabytes and even petabytes of data in any given period of time. While this allows advantages far beyond what we can see, it can be difficult to know the best way to accelerate and speed up these technologies, especially when reactions must occur quickly. For digital-first companies, a growing question has become how best to use real-time processing, batch processing, and stream processing. This post will explain the basic differences between these data processing types.



Benefits and drawbacks of common data processing types

Real-Time Operating Systems

Real-time operating systems typically refer to the *reactions* to data. A system can be categorized as real-time if it can guarantee that the reaction will be within a tight real-world deadline, usually in a matter of seconds or milliseconds.

One of the best examples of a real-time system are those used in the stock market. If a stock quote should come from the network within 10 milliseconds of being placed, this would be considered a real-time process. Whether this was achieved by using a software architecture that utilized stream processing or just processing in hardware is irrelevant; the guarantee of the tight deadline is what makes it real-time. Some other samples of when using real-time systems would be beneficial are bank ATMs, air traffic control, and anti-lock braking systems.

Challenges

While this type of system sounds like a game changer, the reality is that real-time systems can be extremely hard to implement through the use of common software systems. As these systems take control over the program execution, it brings an entirely new level of abstraction. What this means is that the distinction between the control-flow of your program and the source code is no longer apparent because the real-time system chooses which task to execute at that moment. This is beneficial, as it allows for higher productivity using higher abstraction and can make it easier to design complex systems, but it means less control overall, which can be difficult to debug and validate.

Another common challenge with real-time operating systems is that the tasks are not isolated entities. The system decides which to schedule and sends out higher priority tasks before lower priority ones, thereby delaying their execution until all the higher priority tasks are completed.

More and more, some software systems are starting to go for a flavor of real-time processing where the deadline is not such an absolute as it is a probability. Known as soft real-time systems, they are able to usually or generally meet their deadline, although performance will begin to degrade if too many deadlines are missed.

Batch Processing

Batch processing is the processing of a large volume of data all at once. The data easily consists of millions of records for a day and can be stored in a variety of ways (file, record, etc). The jobs are typically completed simultaneously in non-stop, sequential order. An example of a batch processing job is all of the transactions a financial firm might submit over the course of a week. It can also be used in payroll processes, line item invoices, and supply chain and fulfillment.

Batch data processing is an extremely efficient way to process large amounts of data that is collected over a period of time. It also helps to reduce the operational costs that businesses might spend on labor as it doesn't require specialized data entry clerks to support its functioning. It can be used offline and gives managers complete control as to when to start the processing, whether it be overnight or at the end of a week or pay period.

Challenges

As with anything, there are a few disadvantages to utilizing batch processing software. One of the biggest issues that businesses see is that debugging these systems can be tricky. If you don't have a dedicated IT team or professional, trying to fix the system when an error occurs could be detrimental, causing the need for an outside consultant to assist.

Another problem with batch processing is that companies usually implement it to save money, but the software and training requires a decent amount of expenses in the beginning. Managers will need to be trained to understand how to schedule a batch, what triggers them, and what certain notifications mean.

Stream Processing

Stream processing is the process of being able to almost instantaneously analyze data that is streaming from one device to another. This method of continuous computation happens as data flows through the system with no compulsory time limitations on the output. With the almost instant flow, systems do not require large amounts of data to be stored.

Stream processing is highly beneficial if the events you wish to track are happening frequently and close together in time. It is also best to utilize if the event needs to be detected right away and responded to quickly. Stream processing, then, is useful for tasks like fraud detection and [cybersecurity](#). If transaction data is stream-processed, fraudulent transactions can be identified and stopped before they are even complete.

Challenges

One of the biggest challenges that organizations face with stream processing is that the system's long-term data output rate must be just as fast or faster than the long-term data input rate otherwise the system will begin to have issues with storage and memory.

Another challenge is trying to figure out the best way to cope with the huge amount of data that is being generated and moved. In order to keep the flow of data through the system operating at the highest optimal level, it is necessary for organizations to create a plan for how to reduce the number of copies, how to target compute kernels, and how to utilize the cache hierarchy in the best way possible.

Conclusion

While all of these systems have advantages, at the end of the day organizations should consider the potential benefits of each to decide which method is best suited for the use-case.

Additional Resources

[Manage SLAs for your batch services](#) [joe goldberg](#) from [BMC Software](#)