

CTSS with IGT: Deep dive into EOD batch Process

Making the case for more interaction
`free method of running software
programs.

SEPTEMBER 2023

InfoGlobalTech

Innovative Solutions for your Enterprise

Executive Summary

Compatible Time-sharing (CTSS) was compatible with batch processing in 1961. CTSS could offer both time sharing and batch processing concurrently, batch processing computer systems were mostly thinking in terms of time sharing.

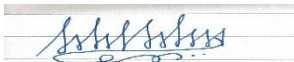
In Job production and computing space, batch processing stays pervasive for both general data processing and for system housekeeping tasks.

Currently, data warehousing are certainly assembled from a variety of data sources and the volume of data that must be processed with service level agreement is unvaried.

Unfortunately, databases may perform slowly because they must take care of concurrency, integrity, maintenance, indices.

But what are the core benefits? How can we work together to derive most out of it? And why does the world need this autonomous data processing unit?

This insightful whitepaper from IGT Infoglobal tech offers a keen analysis of significantly important data source - batch process.



Sahil Daterao,
Associate Business Analyst, IGT Infoglobaltech

History

The ‘Batch Processing’ originates in the traditional classification method of production as Job Production (one-off production), batch production (production of a ‘batch’ of multiple items at once, one stage at a time) and flow production (mass production, all stages in process at once)

Earlier, computer could run only one program at a time. Each user had sole control of the Machine for a scheduled period. They would arrive at computer with program and data and would load their program, run, and debug it. Popular then were, IBM’s Fortran Monitor System, SOS (Share operating system) and finally IBSYS for IBM’s 709x system in 1960.

Third generation computers capable of multiprogramming began to appear in 1960s. Instead of running one batch job at a time, this system can have multiple running at the same time. To prevent deadlock the job scheduler needs to know job resource’s requirements, so various scripting languages were developed to supply this information in a structured way. IBM system System/360 attached support processor is an example of asymmetric multiprocessing, remote batch.

The first general-purpose time-sharing system, Compatible time-sharing system (CTSS) was compatible with batch processing. This facilitated transitioning from batch processing to interactive computing. Allan Scherr at the 25th anniversary of CTSS “CTSS was a wonderful agglomeration of really useful tools including hierarchical file system, primitive graphics for 2D screens, Interactive text editors, debugging tools, implementation of language MAD (derivative of ALGOL) and Saltzer’s TYPSET and RUNOFF word processing program.

Uses

CTSS had big impact in various application worlds not only because it was the first time-sharing system several reasons why the time was right for the system's development.

- The MIT Computation Center was swamped with demand for computing capability and somethings needed to be done.
- Some users wanted faster (i.e. Interactive) access to the computer without the delays inherent in batch processing in their edit-compile-run-cycle
- There were some inherently interactive activities that could not be done in batch mode, Joe Weizenbaum's ELIZA program that simulated the interaction, was first implemented once time sharing was available.
- The idea of time sharing was 'in the air' someone was going to do it soon and MIT computer researchers naturally were at forefront of such development

[this section we list a few specific, and representative, projects that used CTSS Under the leadership of Professor Charles Miller, MIT's Civil Engineering department developed the Integrated Civil Engineering System (ICES), with its COGO (COordinate GeOmetry) and STRESS (STRuctural Engineering System Solver) programming languages that allowed non-computer-programmer engineers to make effective use of the computer.]

Doug Ross's work with Automated Engineering Design (AED) and Computer Aided Design (CAD) resulted in the AED-0 programming language and changed the way computing was done and what was thought of as computable. In particular, it addressed interactivity and data abstraction.

Joe Wiezenbaum's most CTSS famous project was ELIZA, but he also developed the OPL-1 system on CTSS, an incremental programming system providing an early model for how a user to augment both his program and his data base.

MIT's Project INTREX 14 was another use of CTSS. This early, innovation library information system project, led by Professor Frank Reintjes, was in some sense the earliest search engine.

OPS-3, developed by Professor Martin Greenberger et al.,¹⁶ was an “on-line process synthesizer,” a simulation system depending on CTSS to enable the user to iteratively develop the simulation.

The DYNAMO (DYNAMIC MODELS) system, created by Jay Forester’s team, for doing systems dynamics simulations¹⁸ is one such system that benefited from running on an interactive time-sharing system

Many MIT areas of research benefitted from the availability of CTSS, but CTSS was in the right environment to have its capabilities utilized for many things and to have new capabilities developed on top of it.

CoSaCS is a .net application, used for Unicomer group serving in retail, banking, insurance industry. Initially, CoSaCS was maintained and enhanced by BBSL (**Blue Bridge Solutions PVT Ltd.**, UK) for some time, and currently, it is managed by **Caribbean Licensing Corp.** IT and 3rd party service providers.

Unicomer Group was founded in 2000, in San Salvador, El Salvador. It serves chain of retail brands in the consumer durable sector, specializing in furniture, audio & video, appliances and electronics in over 1000 stores in Central America, the Caribbean, South America, United States.

As part of testing for production environment, equal or more amount of time was required to administer and validate the outcomes as of the batch process time. The run-time accounting to simulations was analysed between the restored database and the application that ran the batch process.

To train over 15000 employees the functional flow of the application, was disastrous. The time was right to adapt to Robotic Process Automation to cover all the scenarios to be tested. **Unicomer**, serving all over the Caribbean market, faces the issue with real life business situations on daily basis, this needs to be interpreted by CoSaCS application so the database is up-to-date. So simulate the scenario on daily basis was quite challenging. There

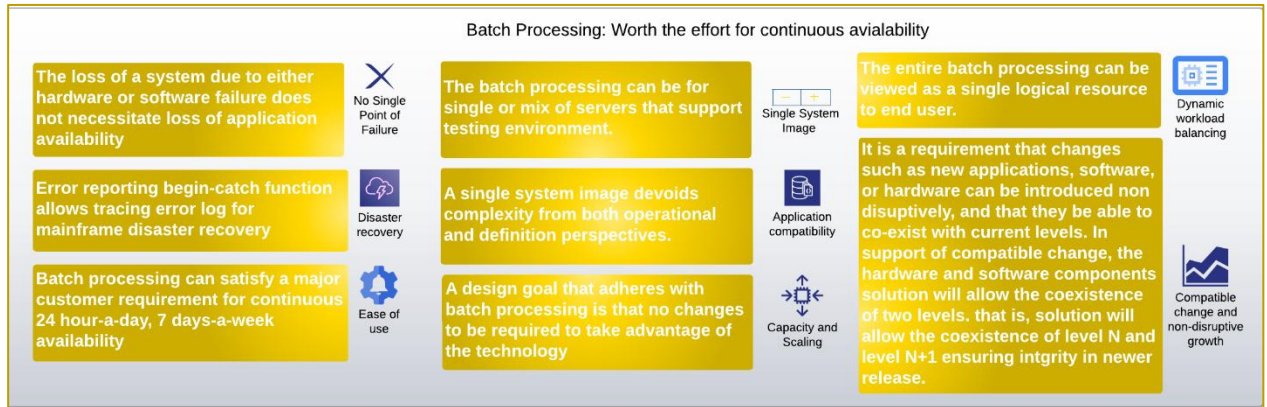
is a special deck in CoSaCS application for batch process if request came in to run it, operator would stop whatever was running on the machine and do the End of Day Job immediately.

With access to batch process and robotic process automation, CoSaCS the .net application could update the database for the client. For issues in daily business process to be updated in database, the quality analyst would conduct requirement engineering, and programmer having more no. of hours worked on the specific domain (Sales/Credit Collection/Accounting) would simulate and run the batch process. The program would be edited, compiled, run and debugged and solution would be delivered to client.

The list of pre-requisite for CoSaCs application would help me make you understand the implementation process, here it is - Visual Studio Community 2019, SQL Server 2016, TeamCity, GitHub, AWS, Windows Server 2016/2019, IIS, Testing Tools -Selenium, Winium, Performance Testing - Load Runner. Oracle Cloud having SQL Server, Caribbean ODS, RPA resource (Selenium) and development testing resource has thin layer connection with CoSaCS .net application having data tier for reports, End of day - Batch Process, activity log and with Production Virtual Machine, IIS & Web Services. It is configured with country level parameters. It is supported with CUCM Cisco Unified Communication Manager. The program lead, debuggers, operation specialists, tester well addressed the requirement needs originating from domain of Sales, Finance, Merchandise, Logistics, etc.

The most direct impact of batch processing was with 3rd party service providers, It allowed them to simulate the scenarios based on client's different requirements. A significant subset of users was willing to expend effort to learn something new and potentially beneficial. A need for documentation with widely accepted standards was fulfilled by the 15000 employees.

Unicomer Group now has data warehouse, logistic service, inventory management systems, payroll system etc in place to serve retail industry more sincerely.



(While transactions and data are cut for a particular days batch activity)

CoSaCS was only a Microsoft Windows based Client application and CoSaCS did not have merchandising and procurement features. While transactions and data are cut for particular days batch activity the Engineers would run the batch process to containerize the information in given parameters. In times of sudden loss of power source, application was available to rerun the batch process. Because of Cursors and triggers (automatic event fire) in the scripting language, the process for that particular account/entity would start from begin. All 3rd party service providers have their own team to analyse the scenario, deliver the outcome best of which would be chosen by **Unicomer**, for this End of Day Batch process would be available 24*7 every day of week, to simulate the process for 1 account (case) or for 1000s.

Errors were traceable. This saved a lot of time for analysts, developers to read the message, fix it right away. Batch process, for testing environment was available at one server and many, i.e the results having processed the batch for one server would be available for other servers if required. This allowed larger no. of engineers, developers to analyse the pre-post conditions, outcomes.

CTSS (Compatible Time Sharing) allowed CoSaCS with batch process a guarantee to deliver identical outcome, irrespective of location, time etc given the inputs are one and the same. So the sales, discounts, price update, stock updates at client layer, batch layer, service layer, data storage and reports would be archived at Unicomer's data repository on daily basis.

This resulted to record keeping function for **Unicomer** and CoSaCs. Now with the record keeping facility, **Unicomer** adopted a new business function, Credit Sanctioning and Credit collection (delinquency management).

Conclusion Summary

Batch processing is an exemplary for producing custom work. It is remarkable approach to improve efficiency in the company. It has potential because of its authenticity, speed, small barriers to implementation etc. In most implementations, load stage is fundamentally a batch process, facilitated by system software, glue logic, and scripting languages.

Batch processing can help everyone. Quality analysts can simulate more test scenarios, developers can suffer less friction and ship new products quicker and market can expand to less affluent population.

CTSS was a very visible proof of concept simulation models as well as simple mathematical ones. Implementation of CTSS (Compatible Time Sharing) enabled the scientific aspirant of MIT, especially belonging to non – Computer Science background, to devoir programming language in their field of study. CTSS enabled CAD, AED, FEA Ansys, software to simulate the experiments from field other than computer science and promote theoretical study.

It is fascinating to note that measurements of the CTSS system: number of users, response times, service times, etc concepts apply to many of the client-server systems being created.

About Sponsor

IGT Infoglobal Tech engage closely with your team to develop data-driven frameworks and quality platforms that will enable you to identify real-time predictions of application defects and provide effective Root Cause Analysis (RCA) measures to prevent the issue from re-occurring.

IGT Team has automation engineers, experts, and advisors with a simple commitment of earning your “Customer Affection”. Since IGT’s inception in Atlanta, Georgia dating back to 2001, IGT have grown globally with our presence in the United States, India, Canada, Poland, and Hong Kong.

www.Infoglobaltech.com

