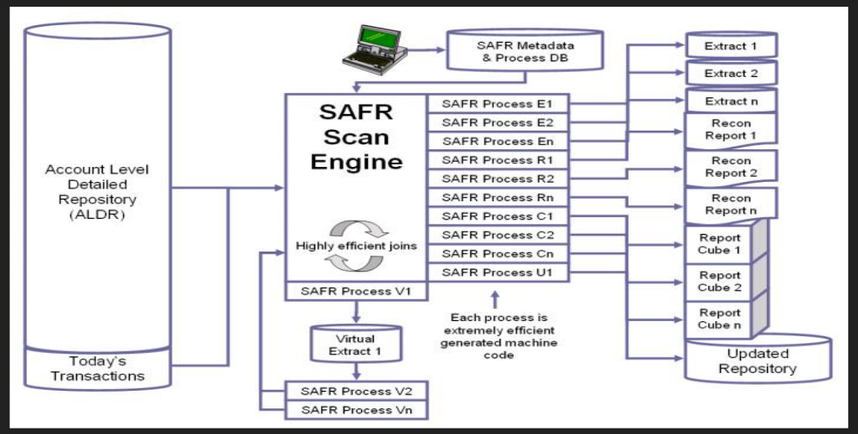
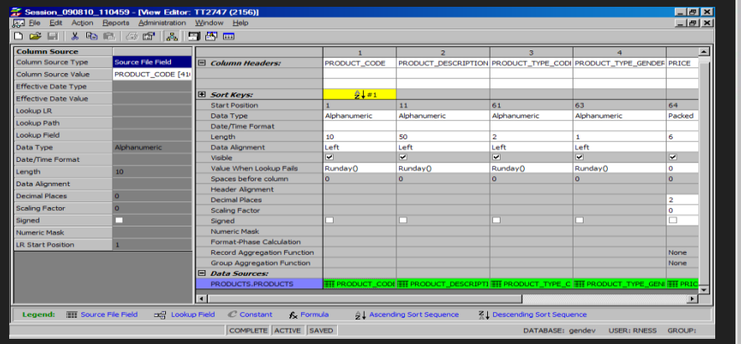
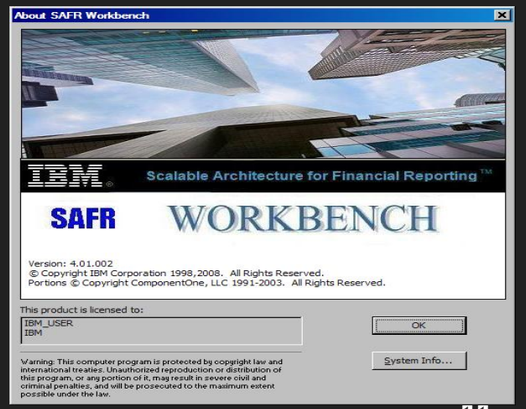
Outline

- **What is GenevaERS**
  - Services asset composed of 100k lines of z/OS assembler code, C++, and Eclipse
  - Generates highly efficient machine code for tremendous scalability
  - Can be used to simply ETL processes all the way to a platform for developing a complex financial and reporting application
  - Created for reporting processes for world's the largest financial organization
  - Has had 20 customers over its 30 years history; currently used by two major FIs as heart of financial, statistical, and KPI processes
- **What it is Not**
  - A database or database centric technology
  - An end-user query tool

- **What it Does**
  Closes the financial books of record on a daily basis – no other application in the marketplace can do this
- **What are the Business Benefits**
  - Allows data analysis at very low levels of detail
  - And of many more attributes – product; geography; channels
  - With greater history
  - Supporting many more financial, regulatory, management and even risk related analysis
  - Which all reconcile and have the same cut off
  - Allowing for faster analytics and insight into business trends by not having to wait for monthly or quarterly closes
  - Saves institutions hundreds of millions of dollars in custom development trying to replicate this asset



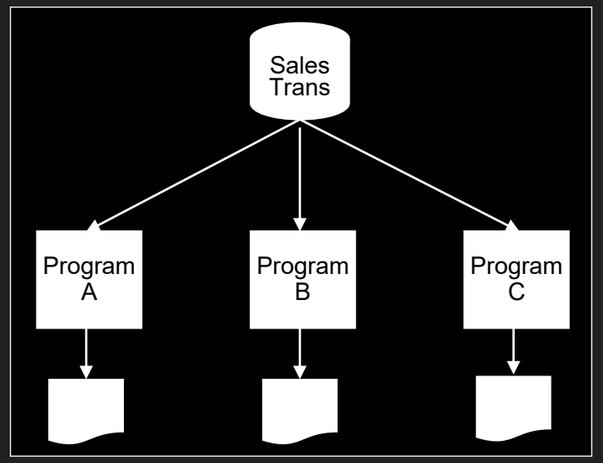




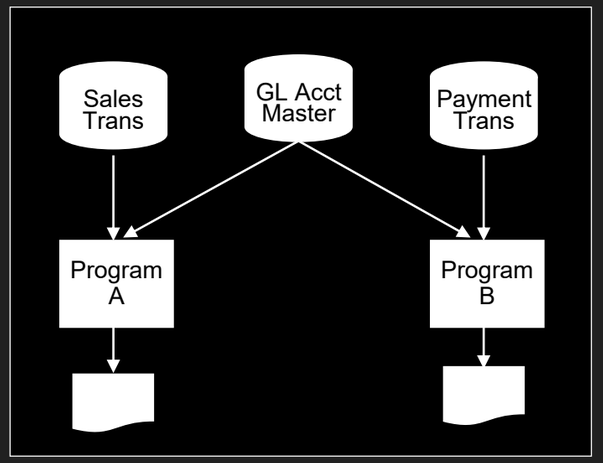
GenevaERS Multiple Aspects of Performance

Problem

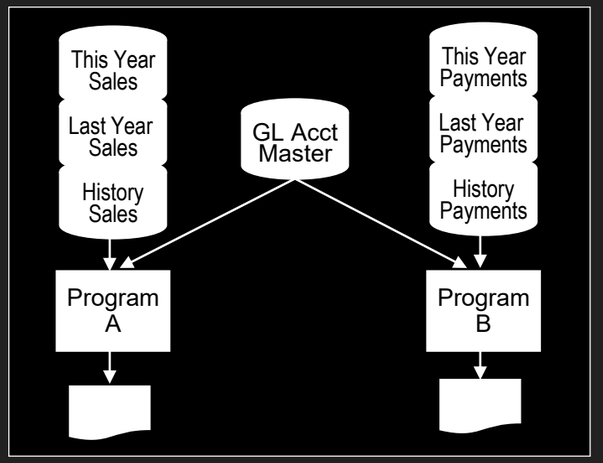
Multiple programs reading same transaction data



Multiple programs reading same reference data

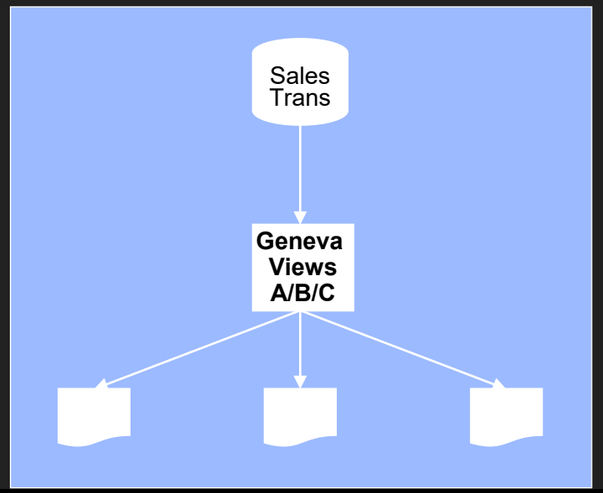


Slow performance due to sequential processing of large data volumes

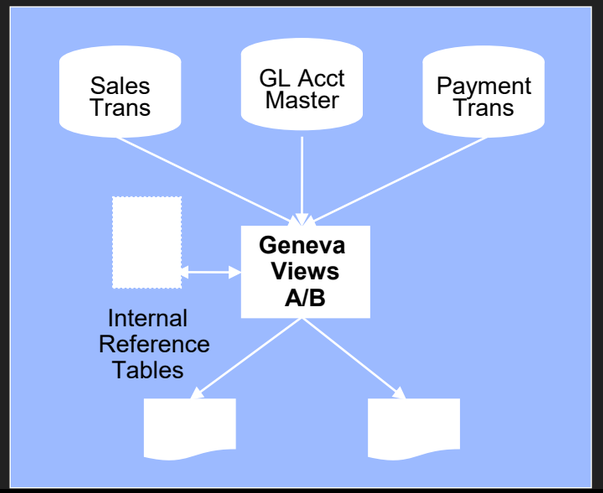


Solution

Multiple report requests produced **simultaneously** in a single scan of the transaction data



Load file into memory to avoid redundant reads

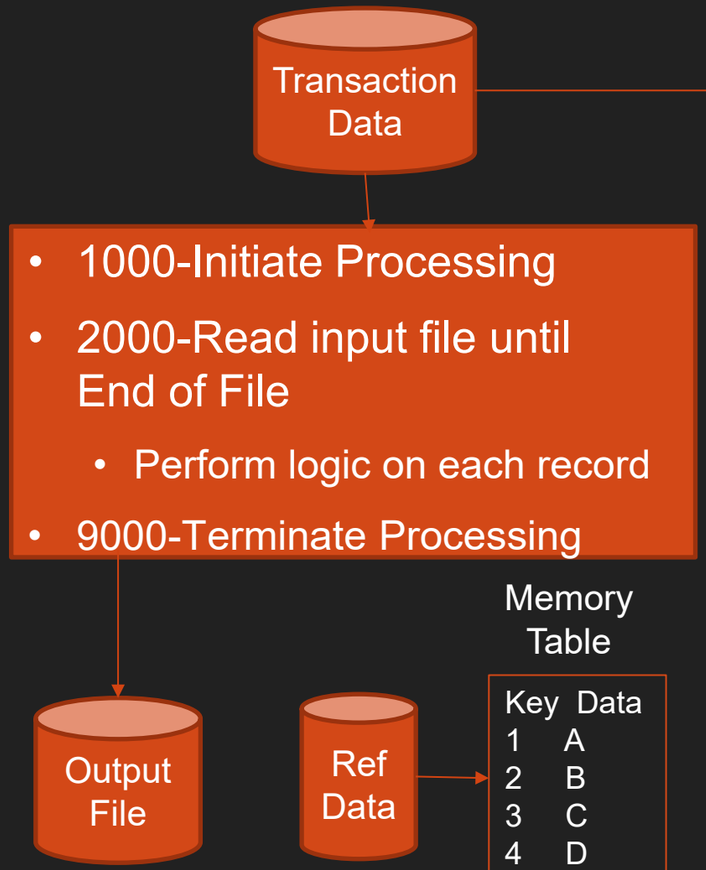


Process files in parallel **and combine the output**

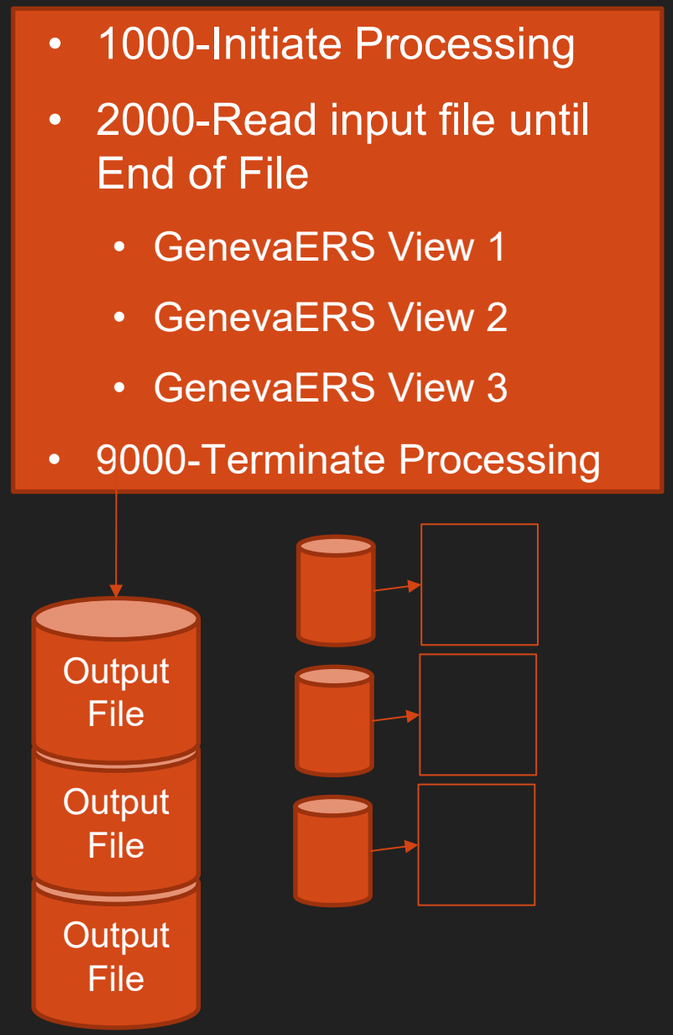
A Sample Customer Application

GenevaERS is fundamentally a compiler, for high volume single-pass, scan processes

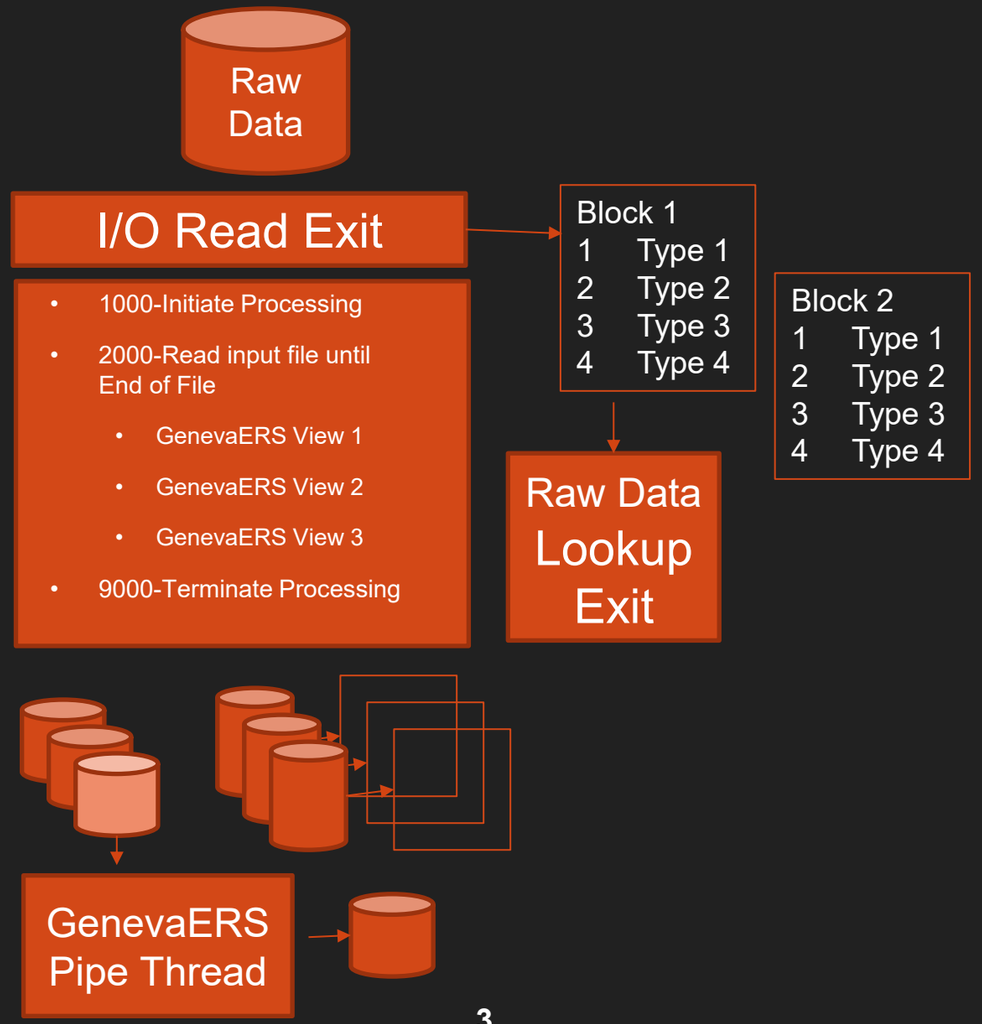
Traditional Batch Program:



Basic GenevaERS Structure:

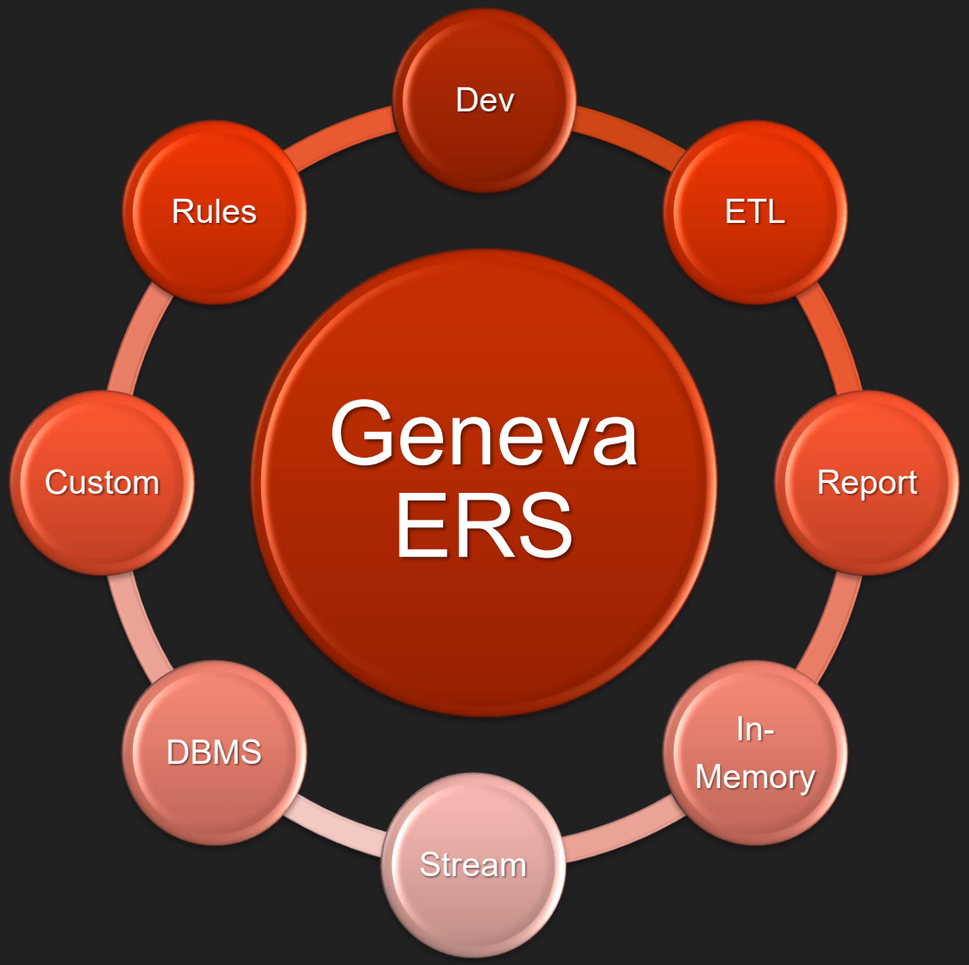


Specialized Customer Structure



GenevaERS Functionality

- High Performance Development Tool
  - Code converted to machine code at run time
- ETL Tool
  - Transforms inputs from various formats and prepares load-ready outputs
- Extracts for Reporting
  - Summary files, formatting flexibility
- In-Memory Processing
  - Windowed joins of large files
  - Read-once process-many flow with multiple views and feedback of spawned records into same flow
- Streaming Batch Processor
  - Processes pre-sorted batch sequential files in a continuous stream
- DBMS-like Functionality
  - Views / Lookups / Select / Join / Group-By
- Customizable Service
  - Customer-directed vendor enhancements
- Rules Engine
  - Consumes business rules via XML
  - Metadata driven layouts, conditions and flows



Code and Configuration Overview

	Leveraged Asset Development
Tooling	The SAFR tool is made up of the following code base: • Performance Engine (C++) 221,040 • Performance Engine (HLASM) 137,802 • Work Bench (Java) 308,822 • SQL 18,635 • Grand Total 686,299 This is a significant asset, nearly 30 years in the making for high performance custom development for the world's largest organizations

Configuration	• One Customer's GenevaERS Instrument Ledger includes approximately 15 different executions of GenevaERS, each with different configuration parameters
	• One of the more complex Executions includes:
	  • 32 different SAFR "views" effectively each an individual program
	  • All executed in parallel
	  • In a Hierarchy Twelve Levels deep
	• These programs average 72 new transactions across all the required business views for each business event

- The SAFR Scan Engine is a parallel processing engine, generating very efficient machine code for IBM z/OS
- Each process generates multiple outputs in one pass through the reporting data.
- In-memory piping between processes and summarization eliminates additional IO.
- Its efficient join algorithms operate on the order of 500,000 joins a CPU second, while its processing capabilities consistently achieve a throughput of a million records a minute.
- Although the Scan Engine is a traditional batch process, because it is able to do in minutes what other technologies require hours to accomplish, with more frequent executions it may outperform real-time systems.