**In-Memory OLTP Demo**

**Startup Process:**

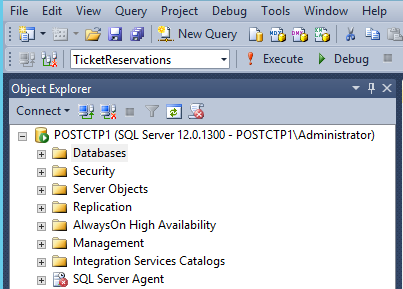
1. When booting up, login as local Administrator because using domain login might not work

**Demo Preparation Process:**

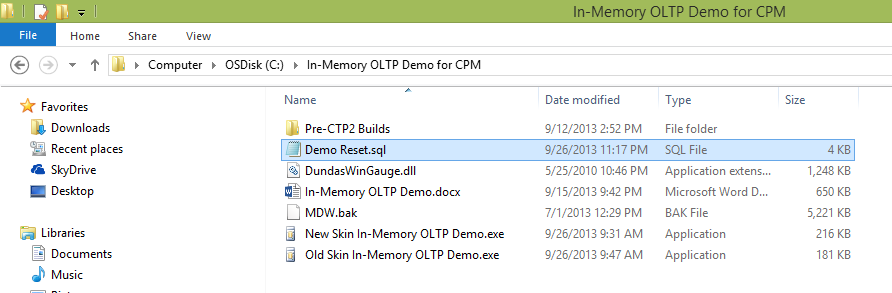
1. Open SQL Server Management Studio in the taskbar



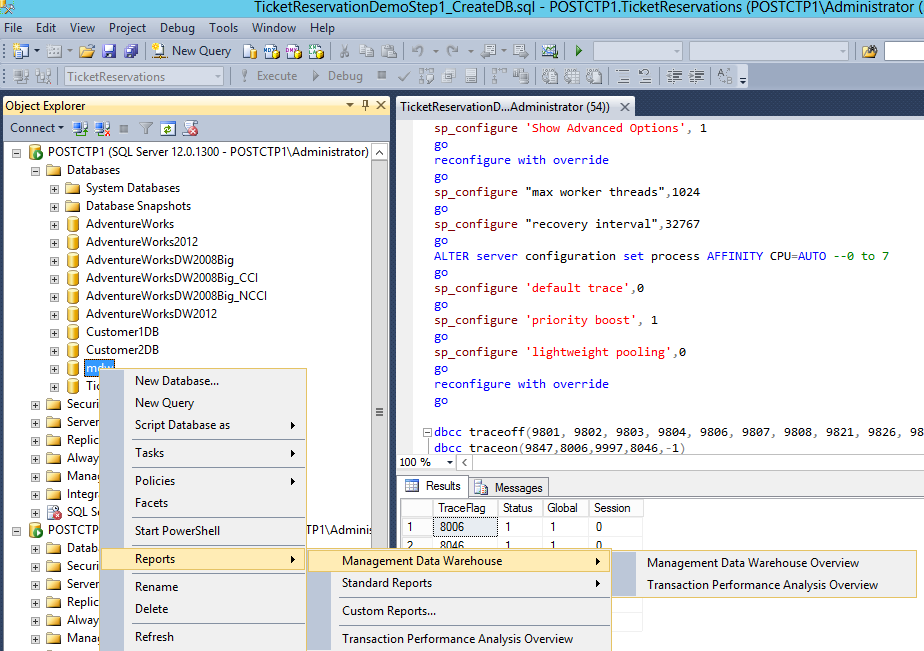
1. Make sure you are connected to the local SQL Server instance using Windows Authentication



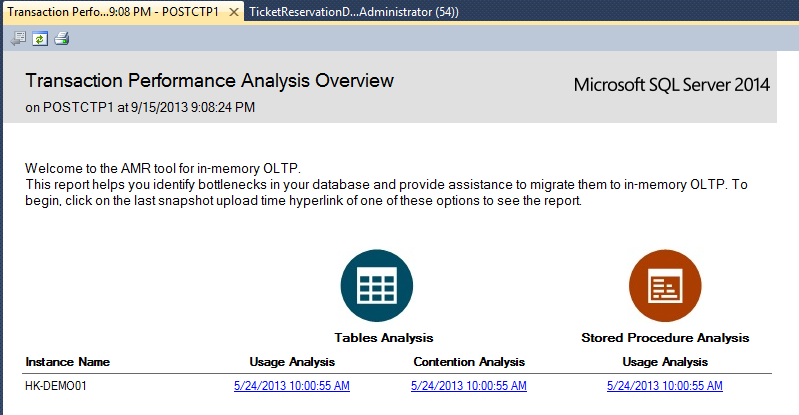
1. Open Windows Explorer, navigate to *C:\In-Memory OLTP Demo for CPM* folder
2. Open the file Demo Reset.sql



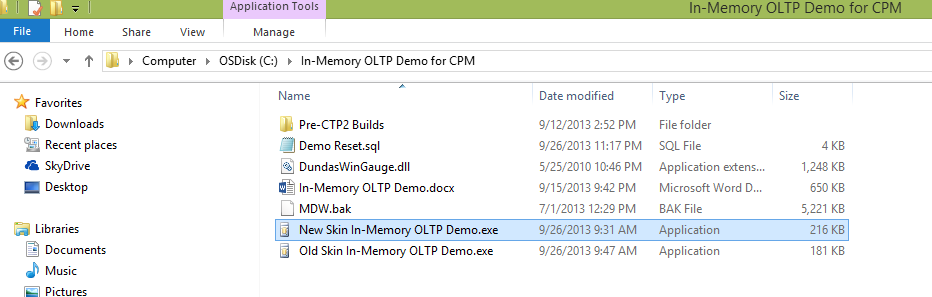
1. Execute the script to reset the demo
2. Expand the Databases tree pane and Right Click on the database mdw, select Reports, select Management Data Warehouse, select Transaction Performance Analysis



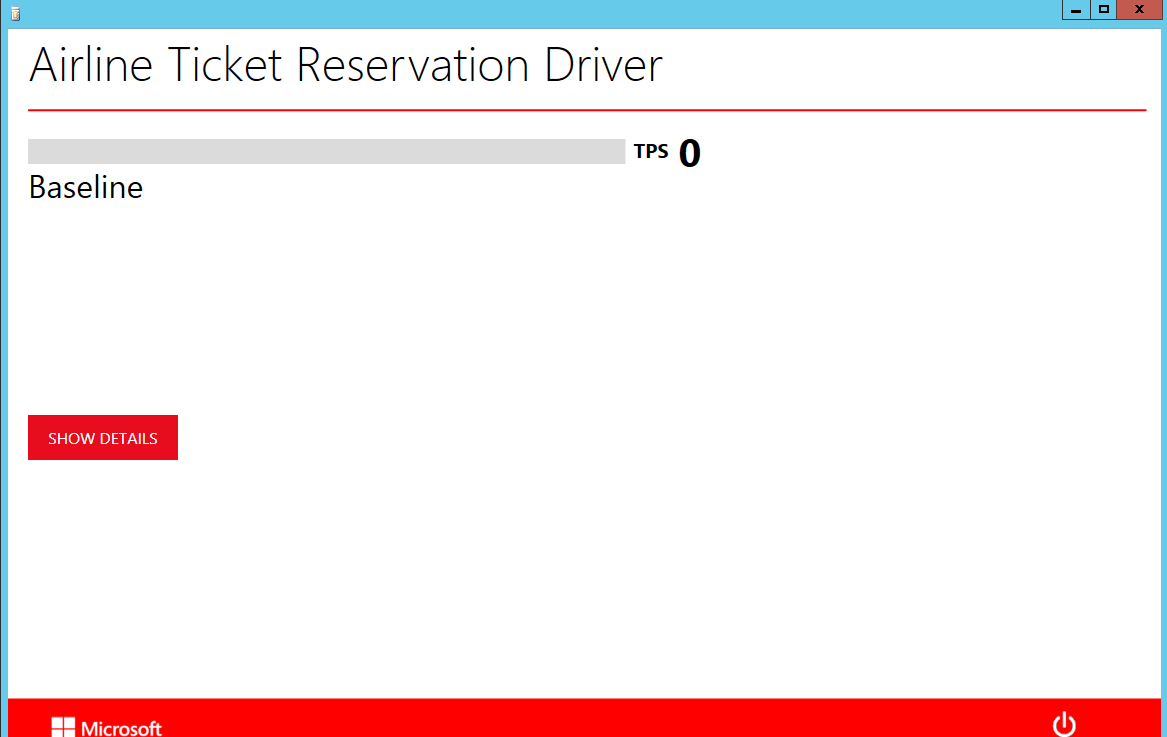
1. You should see the Transaction Performance Analysis Overview window



1. Open Windows Explorer, navigate to *C:\In-Memory OLTP Demo for CPM* folder
2. Open the file New Skin In-Memory OLTP.exe to launch the simulator



1. Now you are ready!



|  |  |  |
| --- | --- | --- |
| **Screenshots** | **Do This** | **Say This** |
|  | 1. Click on the Start button in the lower right corner of the simulator 2. Expand the CPU and latches counters by clicking on the SHOW DETAILS button in the middle left of the simulator | 1. We have developed an Airline Ticket Reservation simulator to test drive In-Memory OLTP 2. With my current system, I am getting over 350 Transactions Per Second (TPS) and it is quite good 3. However, I am not using my entire CPU (~65%) because there are many database latches (~22,900) that block the database process |
|  | 1. Switch to SQL Server Management Studio in the Transaction Performance Analysis Overview window, click on the link under Usage Analysis in the Table Analysis area 2. This should give you a new windows called Recommended Table Based on Usage 3. Click the back button to return to the Transaction Performance Analysis Overview window | 1. SQL Server 2014 ships with a tool to suggest tables which is a great candidate to move into memory 2. The best table is on the right hand corner because it is the easiest to migrate (x-axis) and give you the most performance gain (y-axis) |
|  | 1. Switch to the simulator, click on Migrate button on the lower right corner of the simulator | 1. In order to save time, we built in T-SQL scripts to move the table of interest into memory into our simulator |
|  | 1. Click on the Start button on the lower right corner of the simulator | 1. Wow! Now I am getting ~9X performance improvement just by moving the table into memory without changing my hardware and my application 2. The main reason is that I am using 100% of the CPU resources and the latches are gone! |
|  | 1. Switch to SQL Server Management Studio in the Transaction Performance Analysis Overview window, click on the link under Usage Analysis in the Stored Procedure Analysis area 2. This should give you a new windows called Recommended Table Based on Usage | 1. But there is more, I can speed the performance even more by re-compiling my database stored procedures into native machine language to they can execute even faster 2. The best stored procedures are listed on the top |
|  | 1. Switch to the simulator, click on Migrate button on the lower right corner of the simulator | 1. In order to save time, we built in T-SQL scripts to re-compile the stored procedures of interest into native machine language into our simulator |
|  | 1. Click on the Start button on the lower right corner of the simulator | 1. Wow! Now I am getting ~32X performance improvement by re-compiling my most used stored procedures into native machine language. 2. Amazing! This is the same result as adding more than 30 servers to my existing server. 3. (Optional) For those of you still competing with Oracle RAC (Real Application Clusters), this results demonstrates that you don’t really need to do scale out anymore because your database server can scale 30 times faster. Plus, scale out introduce more complexities, and requires additional licensing. SQL Server 2014 really gives you better performance and lower TCO than Oracle! |