IBM Smarter Business 2013
Tee mahdollisesta totta

Tuning TM1 to get the most out of your investment

Ronnie Rich TM1 Product Management

Please Note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Problems Solved by TM1 Multi-Threaded Query

Present Customer Complaints

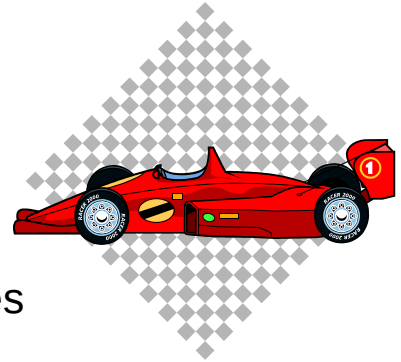
CPU Utilization: "I've got 16 cores and my CPU utilization is at 15%"

Server PVU Value: "More cores do not make my queries faster"

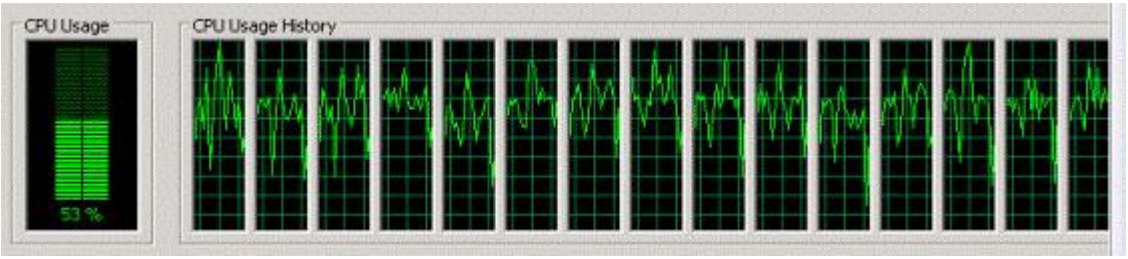
Data Scale: "TM1 Solutions have a data volume ceiling"

Rule Caution: "Rules slow down my queries to an unacceptable performance level"

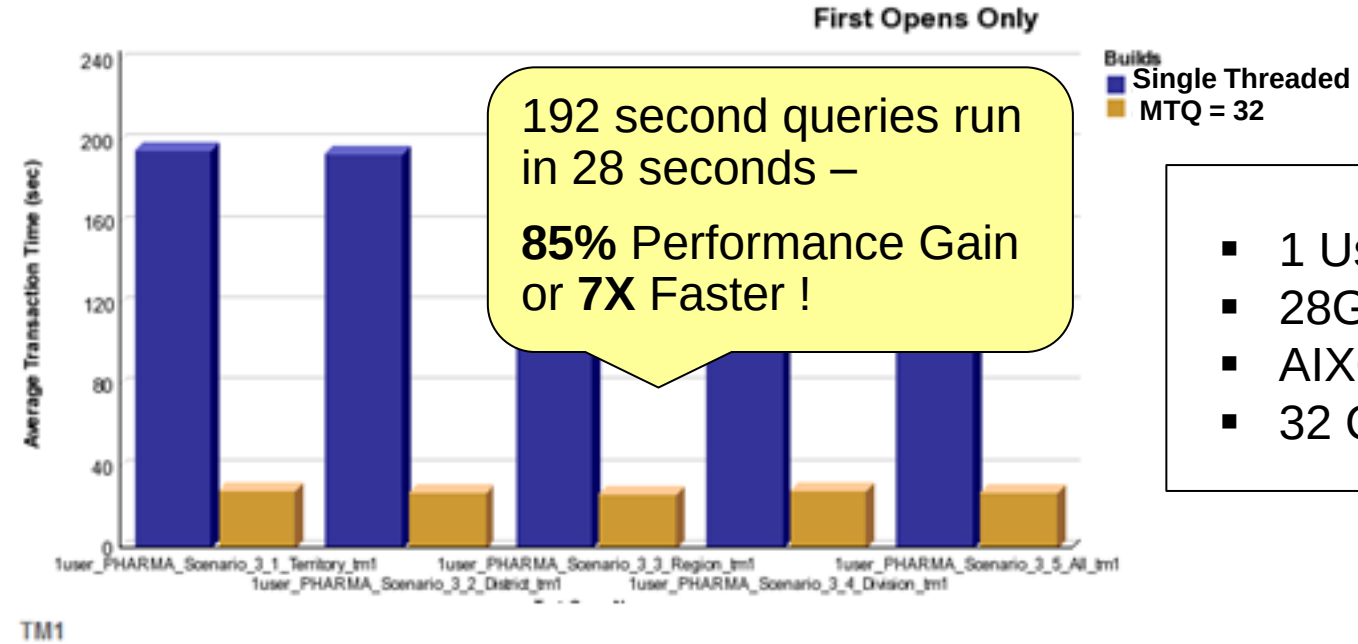
New Multi-Threaded Query Approach



- Simple Configuration: tm1s.cfg -> MTQ = <total number of server cores>
- All UIs can leverage MTQ: TM1 multi-threads stargate cache creation
- High Performance: Query speed improves relative to available cores
- Manages Concurrency: Available cores are load balanced across queries



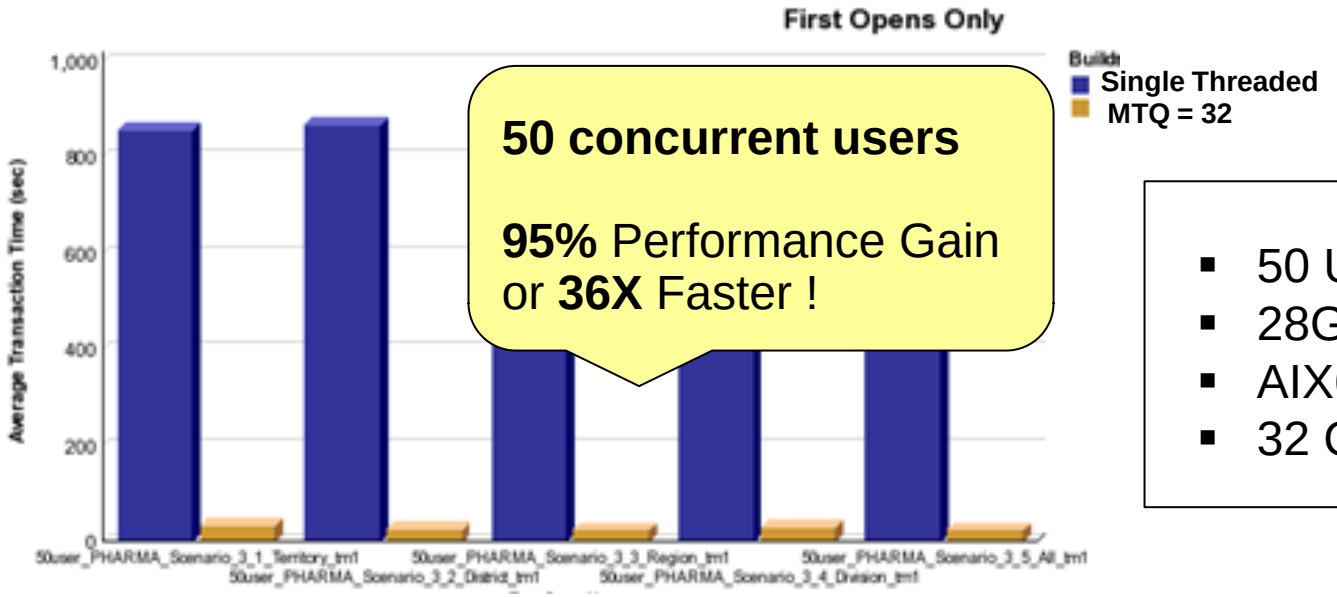
MTQ Benchmark Test #1 Results



192 second queries run in 28 seconds –

85% Performance Gain or **7X** Faster !

- 1 User
- 28G Model
- AIX64 / WAS
- 32 Cores



50 concurrent users

95% Performance Gain or **36X** Faster !

- 50 Users
- 28G Model
- AIX64 / WAS
- 32 Cores

Easier

Faster

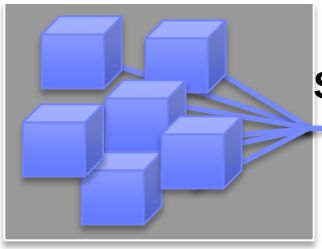
State of the art Turbo Integrator Per Cube SaveData operation

IBM

Background

Customers frequently use the TI Function SaveDataAll() to serialize in-memory data to disk when they have updated a cube with logging off …. To make the TI data update run faster. The SaveDataAll is an expensive operation as it basically locks the entire server for the duration of the serialization process.

Before: TM1 (in memory)



SaveDataAll()

Disk

Cube1.cub
Cube2.cub
Cube3.cub
Cube4.cub
Cube5.cub
Cube6.cub

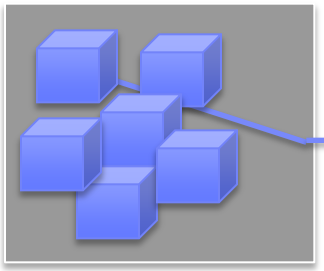
CubeSaveData

- The CubeSaveData('CubeX') TI Function allows customers to designate selective cube objects to serialize, to prevent unnecessary serilization of all cubes on the server, and therefore reducing contention.

Keep In Mind

- **SaveData operations are blocking events (we're working on this…ETA Q4 2013)**
- **TM1 Data files must NOT be accessed externally during SaveData (backups, Copys, Scans…**

Now: TM1 (in memory)



CubeSaveData (Cube3)

Disk

Cube3.cub

Faster

Bigger

State of the art Turbo Integrator TM1RunTI.exe

TM1RunTI.exe is a command line interface tool that can initiate a TI process from within any application capable of issuing operating system commands.

```
OS> tm1runti –server MyTM1Server –username John –pwd "my secret"
    -process "Run Sales Processing" OverwriteParam=yes UpdateParam=32.5
```

TM1RunTI.exe can be deployed to enable

Better user responsiveness for long running, Action Button driven TI Processes.

And

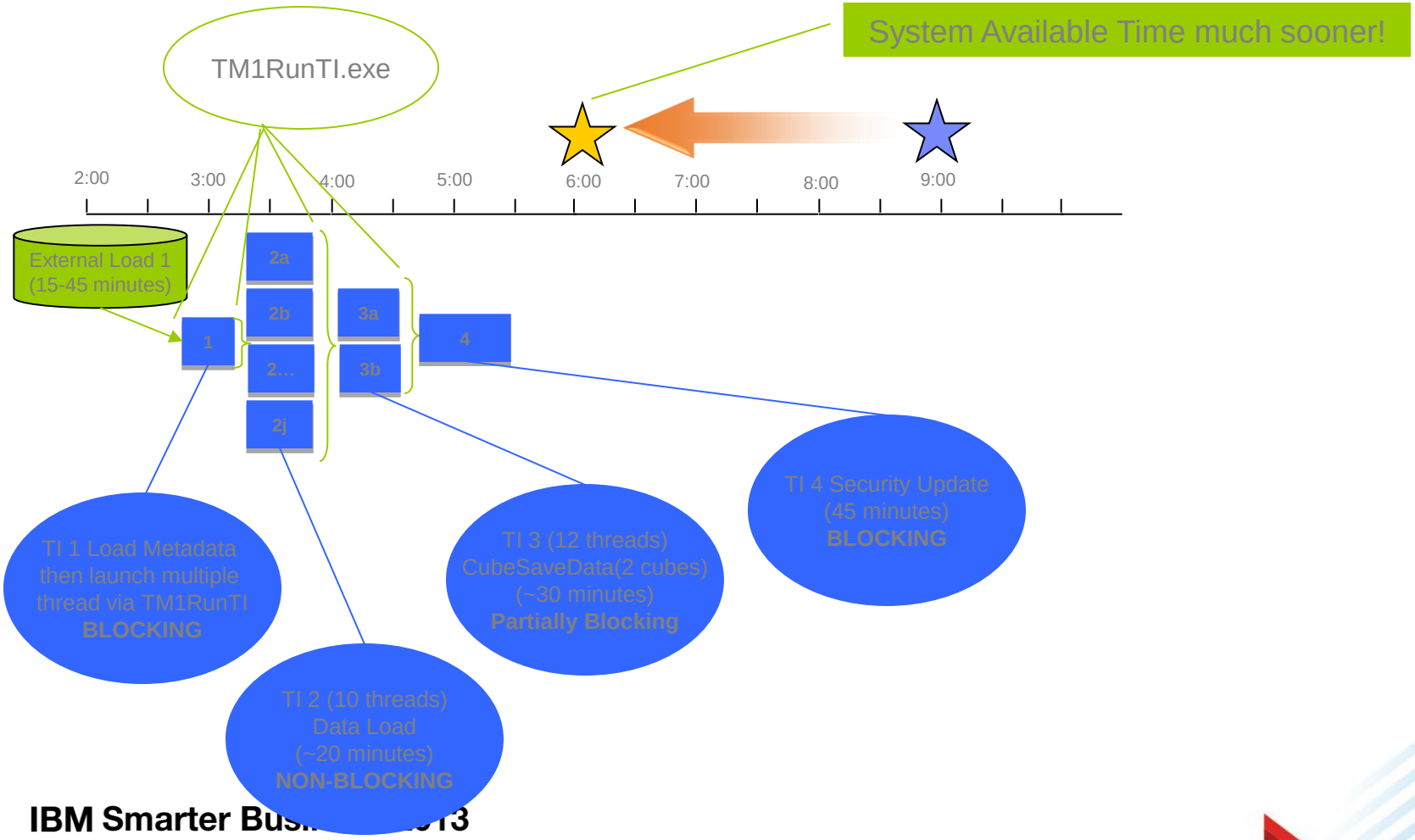
More efficient, controlled synchronization of overnight processing.

Faster

Bigger

State of the art Turbo Integrator
TM1RunTI.exe – Improved Overnight Processing

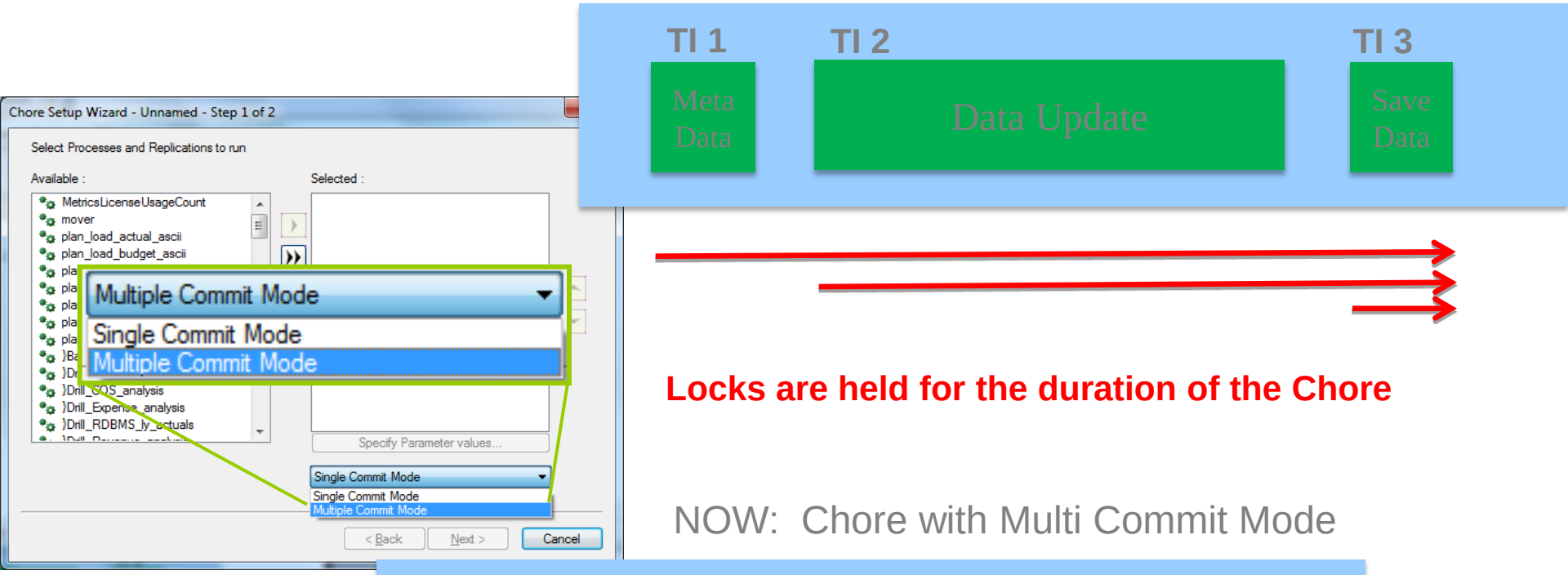
NOW: Use TM1RunTI.exe for efficient process triggering and multi-threading. Resulting in earlier system availability.



Easier

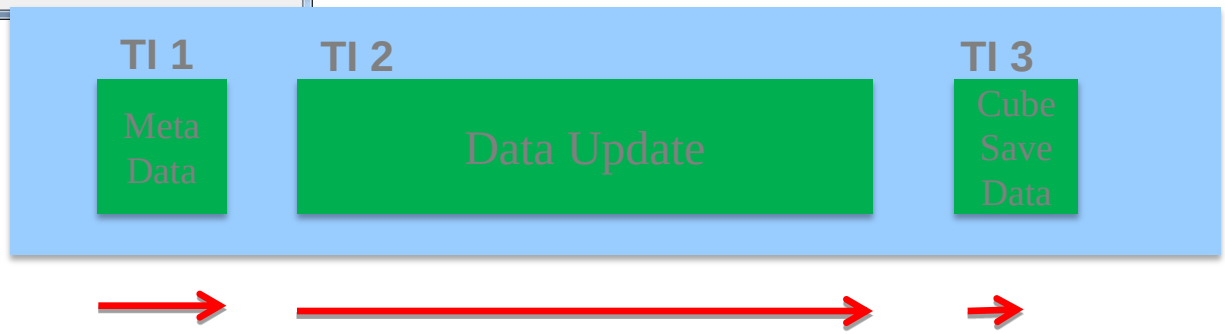
State of the art Turbo Integrator Multi-Commit Chores

BEFORE: Chore with Single Commit Mode



Locks are held for the duration of the Chore

NOW: Chore with Multi Commit Mode



Locks are committed when each TI in the Chore is complete resulting in shorter locks and less opportunity for contention

Calculations & Cache – The Basics

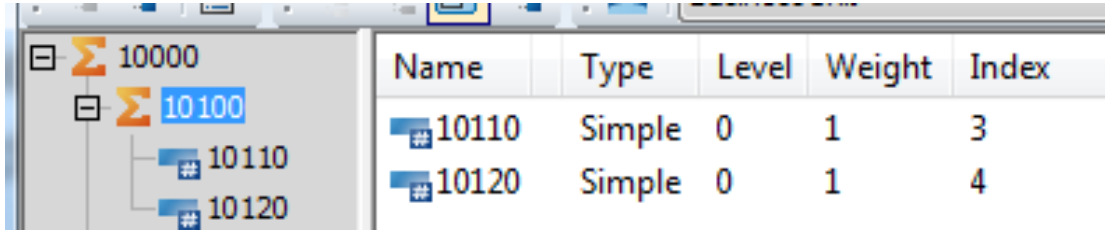
IBM Cognos TM1 uses two types of calculations:

Cube Rule Calculations FAST

```
['Cost']=N:['Price']*['Quantity'];
```

Rules are critical to real time calculations/modeling based on dynamically changing data and attributes

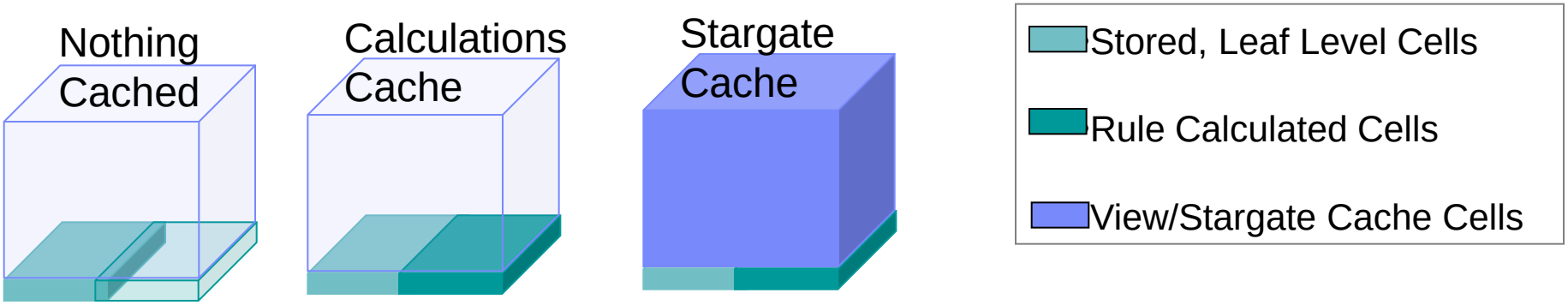
Dimension Aggregation Calculations VERY FAST



- And two types of caching:
  1. Calculation Cache for cube rule populated cells
  2. View (Stargate) Cache for aggregations

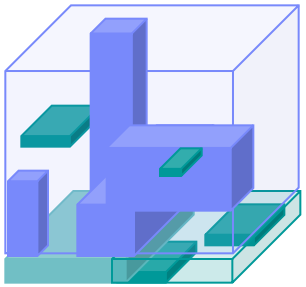
Calculations & Cache

- Caching made simple



It's not really that simple…

Real World Mixed Caches



Calculations & Cache - Differences

Rule Calculation Cache is pretty simple. The first time a rule calculated cell is evaluated the resulting answer is stored in calculation cache and reused until it is invalidated.

View/Stargate Cache is a bit trickier:

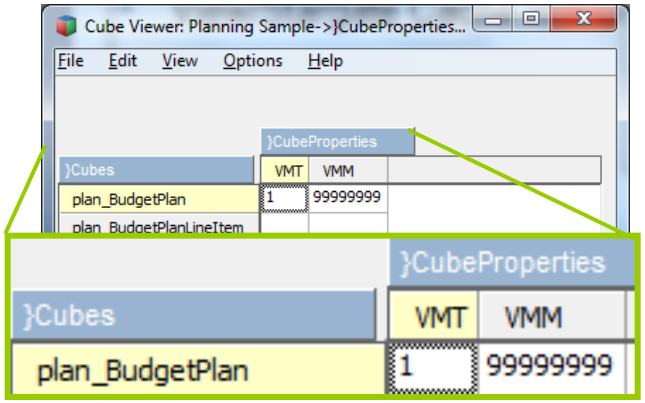
Created When Queries are Run

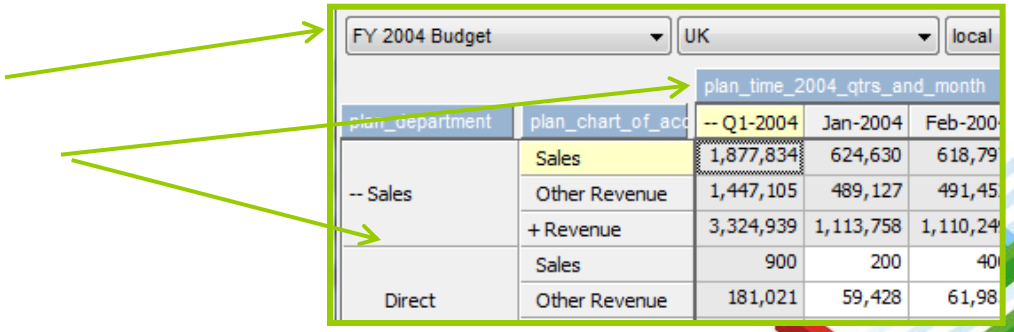
Dependent on:

- VMT (minimum time to create)
- VMM (maximum RAM to use)

Defined by Query

- Title selections
- Axis dimensions

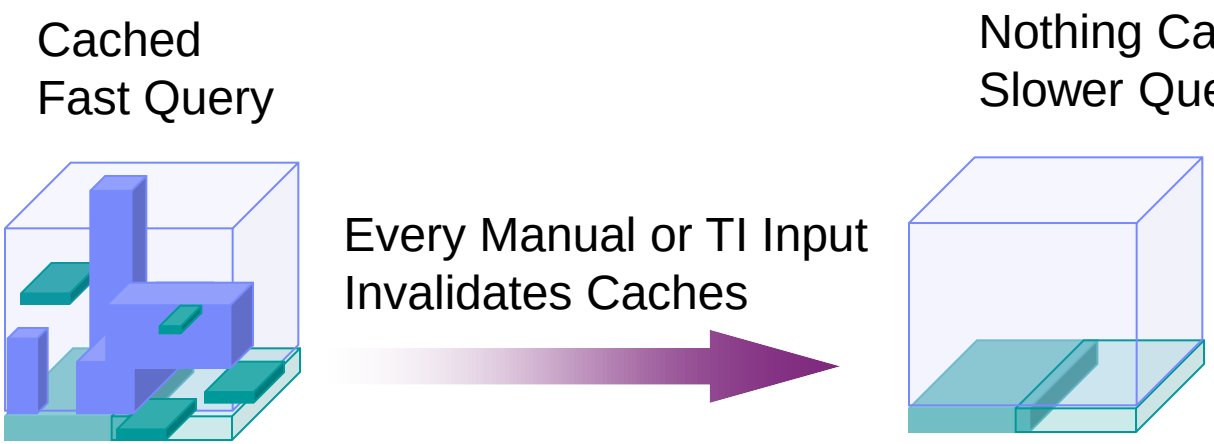




Calculations & Cache - Similarities

Both Cache types are automatically used by virtually all clients (Excel, Insight, Web, BI, TI*…)

Both Cache types are Invalidated by any input to the cube. So what is “invalidation” to TM1?

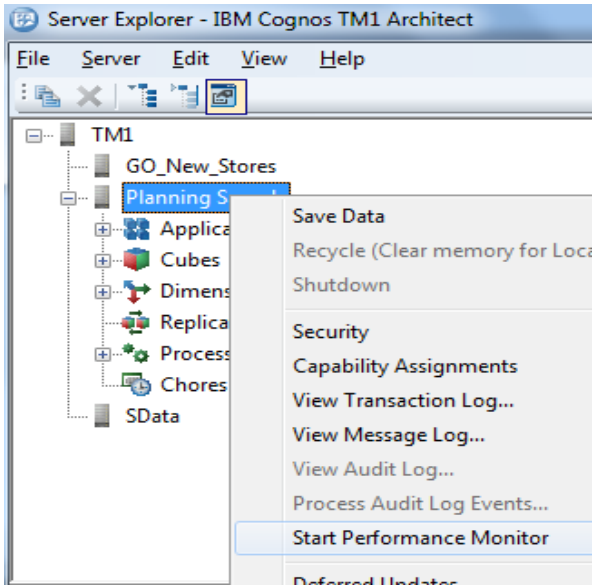


- And there’s more, Cube Dependencies relationships cause multi-cube cache invalidation on cell update…

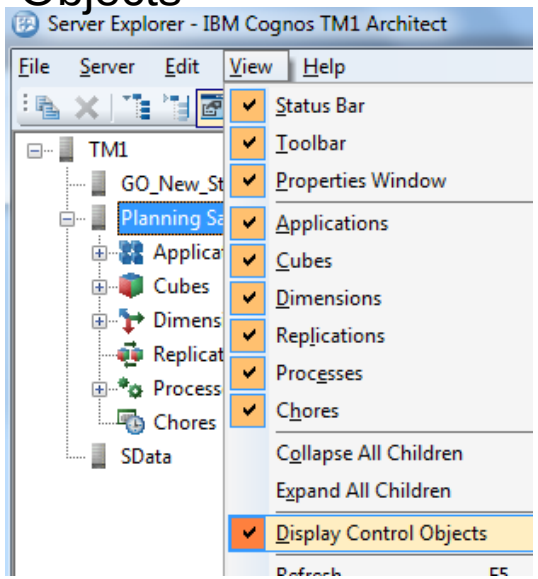
Calculations & Cache Optimization

Monitoring Cache Utilization with Performance Cubes

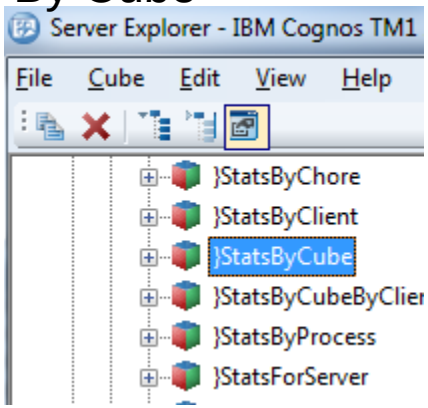
Step 1 – Turn on Performance Monitor



Step 2 – View Control Objects



Step 3 – Open Stats By Cube



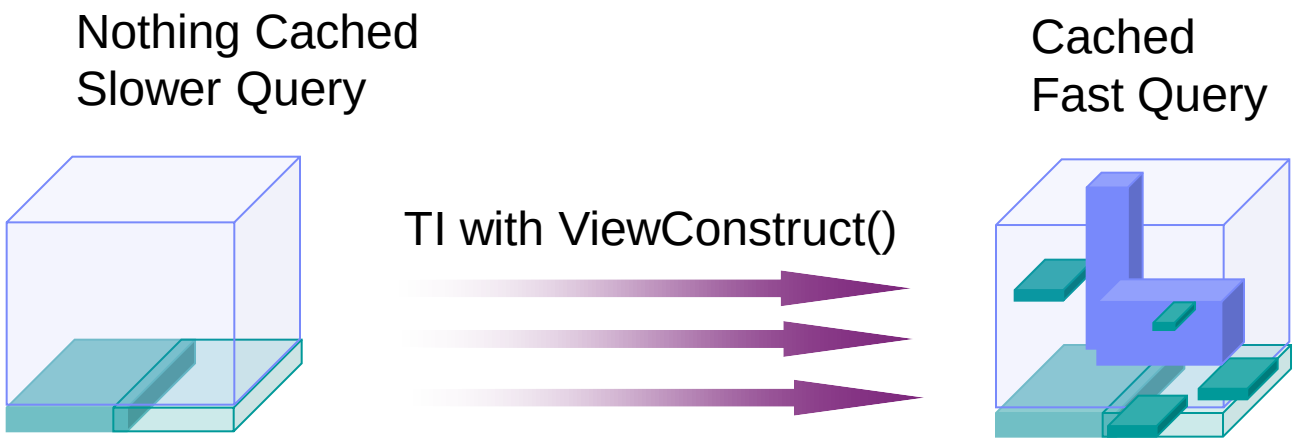
	}StatsStatsByCube							
}TimeIntervals	Memory Used for Views	Number of Stored Views	Number of Stored Calculated Cells	Number of Populated Numeric Cells	Number of Fed Cells	Memory Used for Calculations	Memory Used for Input Data	Total Memory Used
LATEST	527104	2	5905764	779831	8578141	125797120	75469824	201266944
0M22	527104	2	5905764	779831	8578141	125797120	75469824	201266944
0M21	314880	1	5905756	779831	8578141	125797120	75469824	201266944
0M20	314880	1	[illegible]	[illegible]	[illegible]	125797120	75469824	201266944
0M19	314880	1	5905756	779831	8578141	125797120	75469824	201266944
0M18	0	0	0	779831	8578141	0	75469824	75469824

Analyze Caching

Calculations & Cache Optimization
Pre-Caching With Turbo Integrator

Turbo Integrator's ViewConstruct function can by used to pre-populate Cache.

```
VIEWCONSTRUCT(CUBENAME, VIEWNAME);
```



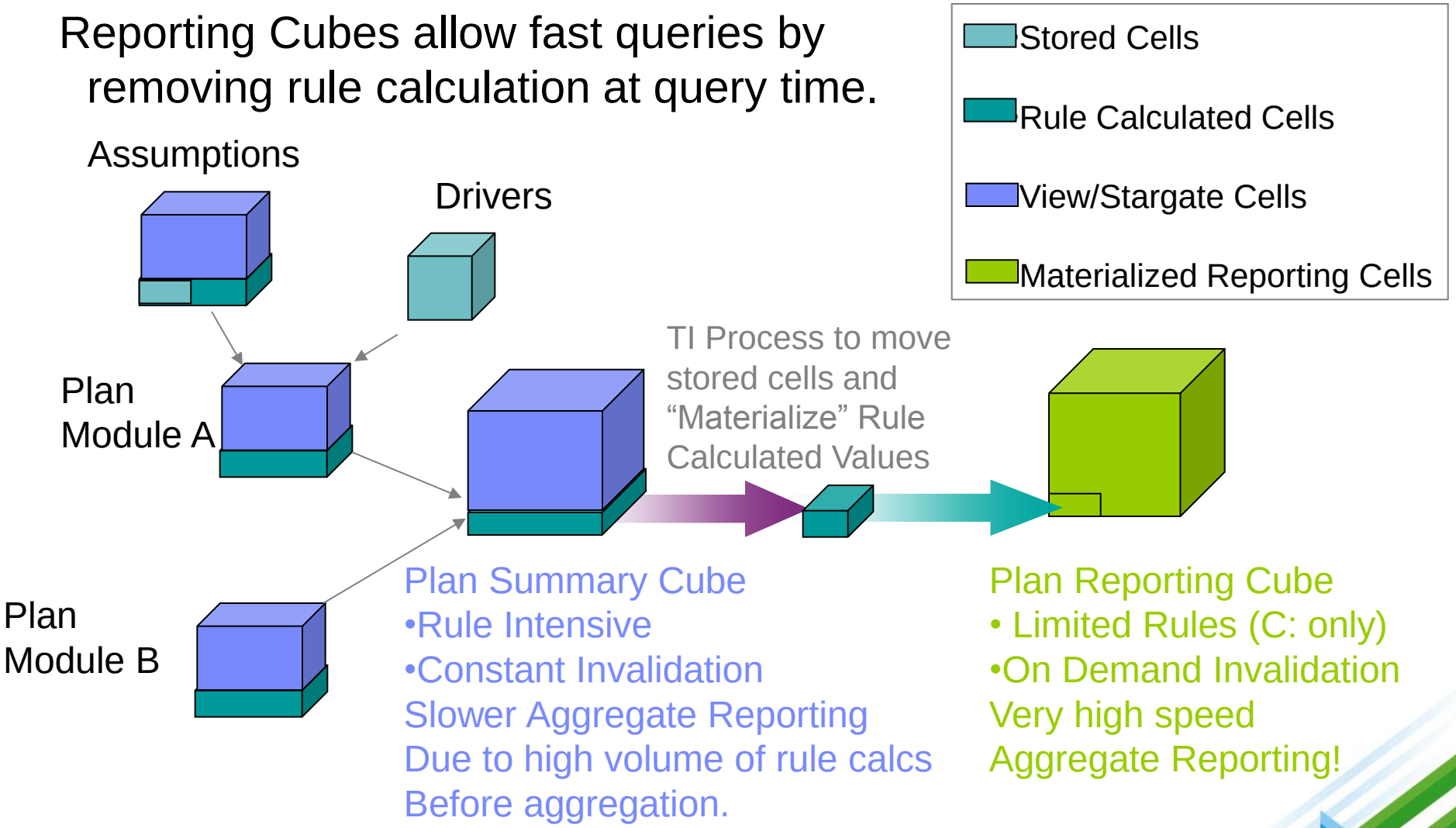
- Execute Pre-Cache TI's at Server Startup or after data/meta-data imports

Keep In Mind

- **VIEWCONSTRUCT is a locking event (we're working on this)**
- **Populating Cache for use as a TI Process source requires calling VIEWCONSTRUCT in a sub-process via EXECUTEPROCESS**

Calculations & Cache Optimization
Reporting Cubes

Reporting Cubes allow fast queries by removing rule calculation at query time.



Keep In Mind

Not Quite Real Time

Thank you