



IN-MEMORY COMPUTING BEST PRACTICES

Adding Speed and Scale to Existing Applications

Rob Meyer
Outbound Product Management

In-Memory Computing Best Practices Series

1. Build an in-memory computing foundation and roadmap
 0. Focus on, measure and Improve the customer experience
 1. Design around the customer experience
 2. Build a new real-time layer on top (of batch)
 3. You need real-time data, business processes, events, tasks
 4. Design for speed, scale, and real-time HTAP at scale
 5. Prioritize real-time projects based on a greater plan

In-Memory Computing Best Practices Series

1. Build an in-memory computing foundation and roadmap
2. Add speed and scale to existing applications
3. Develop new applications, channels and APIs
4. Make the bridge to the cloud
5. Implement HTAP and real-time data management
6. Become event- and stream-driven
7. Think different with machine and deep learning



The Biggest Challenge with Existing Applications

**“ACCELERATING THE WRONG ARCHITECTURE
WITH HARDWARE”**

The Speed and Scale Challenge with Digital Business

Existing Infrastructure Data Growth Challenges in the Last Decade



Existing Applications



New Applications, Analytics



Streaming, Machine Learning

**10-100x
Queries and
Transactions**

**50x
Data
(Big Data)**

**10-1000x
Faster Speed
(Hours to Sec)**



RDBMS



NoSQL



Hadoop

The Wrong Architecture

Accelerating Existing Architectures with Hardware



Existing Applications



New Applications, Analytics

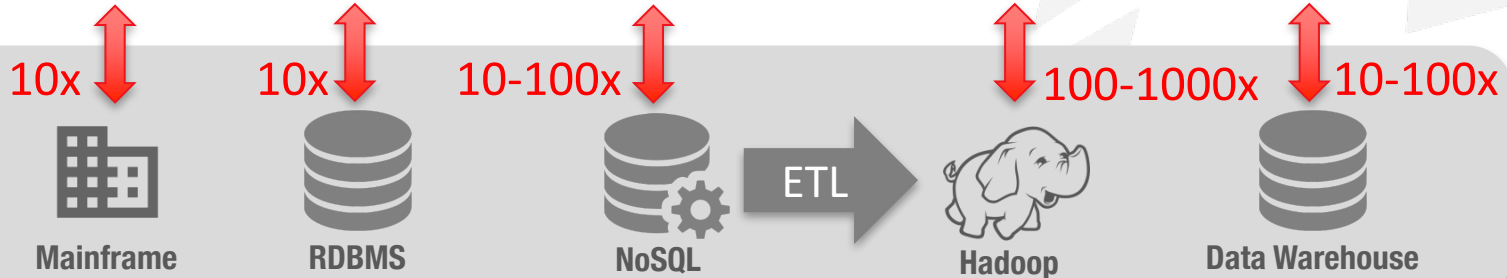


Streaming, Machine Learning

2. Doesn't lower the latency (to improve responsiveness)

3. IT'S VERY EXPENSIVE, AND IS ONLY A SHORT TERM FIX!!!

1. Doesn't merge the data (for APIs, Real-time analytics)





Best Practice 2.0

DON'T DO ONE-OFF INVESTMENTS IN HARDWARE

**USE THE MONEY FROM SCALE-UP HARDWARE TO
INVEST IN THE RIGHT ARCHITECTURE**

The Right Architecture

In-Memory Computing In-Between Data and Apps/APIs



Existing Applications



New Applications, Analytics



Streaming, Machine Learning

2. In-memory speed close to app to lower latency

3. APIs + HTAP collocated with data for speed and scale

1. 1000x horizontal scale to merge all the data and lower costs

10x



Mainframe

10x



RDBMS

10-100x



NoSQL

ETL



Hadoop

100-1000x



Data Warehouse

10-100x



Best Practice 2.1

**BUILD A HORIZONTAL-SCALE IN-MEMORY DATA GRID
TO ADD SPEED AND SCALE
TO EXISTING APPS, NEW APIS, AND NEW ANALYTICS**

New Omnichannel Experience

New Technologies and Data
(cloud, social/streaming, ML/DL)

Security, Identity Management

Business Services (APIs)

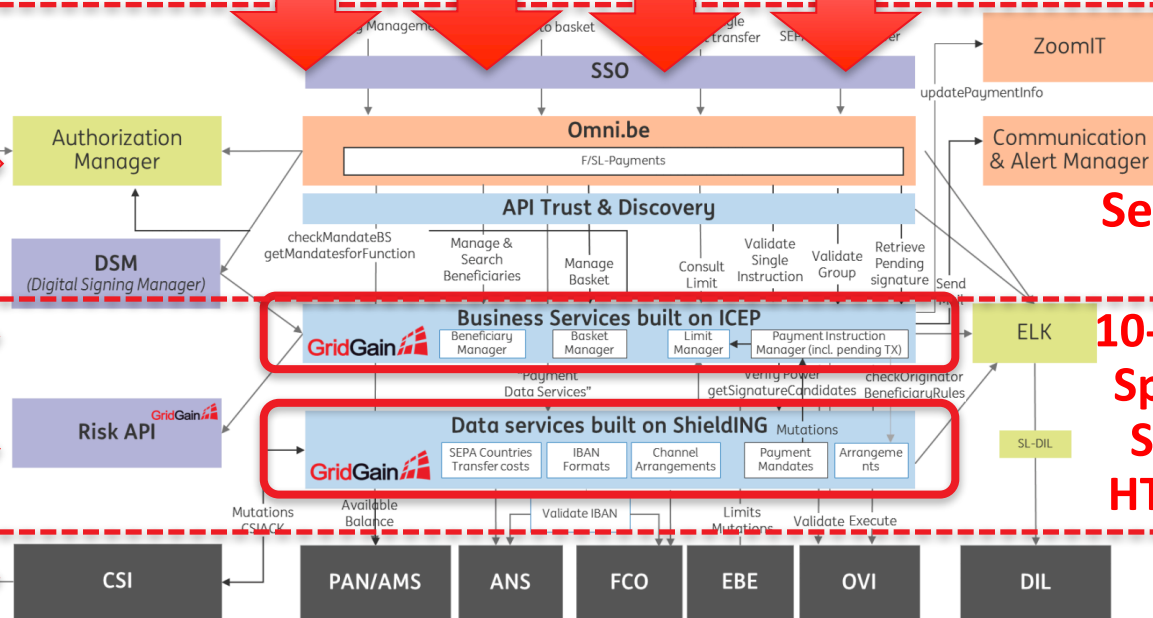
Data Services (APIs)

Existing channels

New Processes

New Data, Apps,

How does everything fit together?



Security

10-1000x
Speed,
Scale,
HTAP, ...



Existing Applications



New Applications, Analytics



Streaming, Machine Learning

**In-Memory
Data Grid**

**In-Memory
Database**

**Streaming
Analytics**

**Continuous
Learning**

GridGain In-Memory Computing Platform



RDBMS



NoSQL



Hadoop

How an In-Memory Data Grid (IMDG) Works



Existing Applications



New Applications, Analytics



Streaming, Machine Learning

ODBC/JDBC

ANSI-99 SQL

Key-Value

ACID Transactions

Compute (and Service) Grid

In-Memory Data Store

ODBC/JDBC



RDBMS



NoSQL



Hadoop

Best Practice 2.1 - Corollary 2.1.1

“Don't accelerate the wrong architecture with hardware”

- Scaling vertically won't work
 - Doesn't scale enough
 - Doesn't lower latency
 - Doesn't support new data
 - Doesn't support APIs or analytics speed and scale
- The only approach is an in-memory data grid architecture
 - Moves the data into memory memory, closer to the customer
 - Delivers horizontal scale with collocated computing

Best Practice 2.1 - Corollary 2.1.2

It's not just about speed and scale.

It's about the customer experience.

- Look ahead (to be) and plan ahead to build your new data layer
- If you don't know the right architecture, ask the right questions
 - Do the math on how much scale you need in the longer term
 - Who else is using the data you have?
 - Which analytics need to be real-time?
 - What new data do you need to improve the experience?
 - What analytics do you need to run to improve the experience?



From Part 1 of the Series ...

Best Practice 1.2

YOU CAN'T VIOLATE THE LAWS OF PHYSICS

BATCH CAN'T DO REAL-TIME

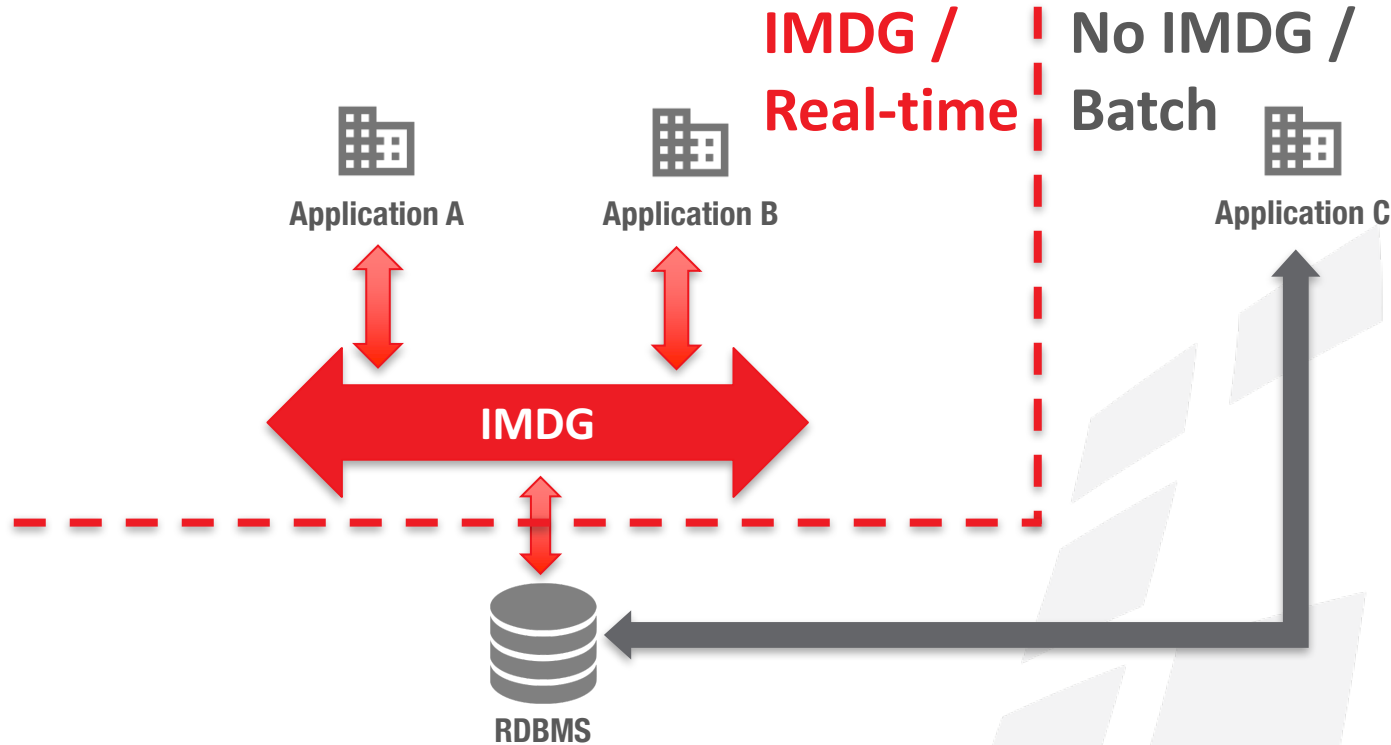
SO BUILD A NEW REAL-TIME LAYER ON TOP



Best Practice 2.2

**CHOOSE THE BEST INTEGRATION ARCHITECTURE
BETWEEN THE NEW (REAL-TIME) LAYER
AND THE EXISTING (NON-REAL-TIME) APPS**

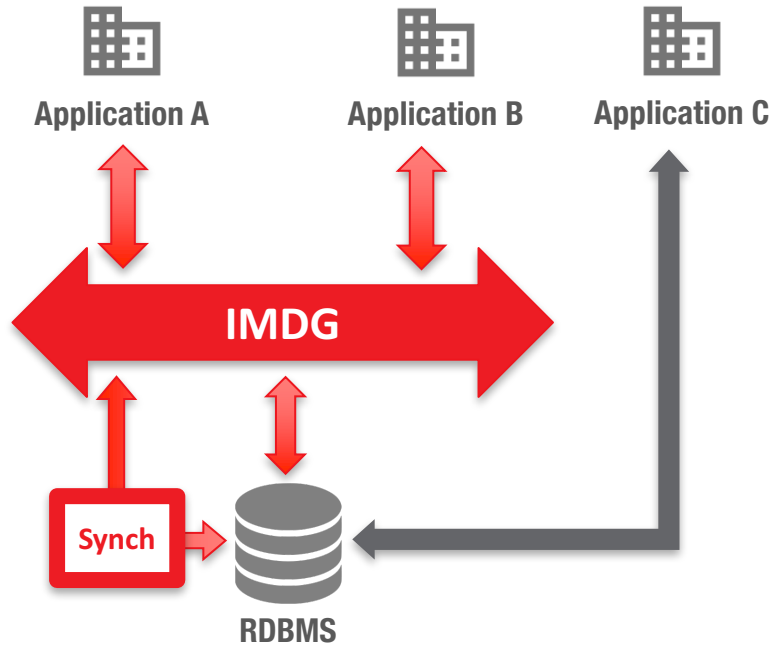
The Challenge - Keeping Two Worlds in Synch



Synch or CDC?

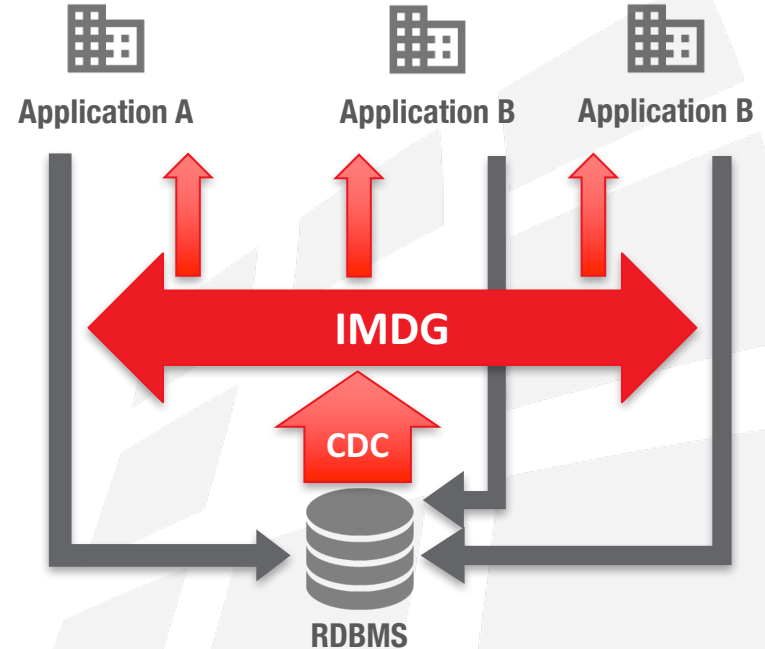
Alternative Approaches for Keeping Existing and New Apps in Synch

Synch



OR

CDC



Synch or CDC?

Alternative Approaches for Synchronization

	Synch	CDC
Does Org own apps?	Yes	No
Immediate consistency needed?	Yes	No
Cross-app write collisions?	No	Yes
IMDG Pattern	Read/Write	Read-only
Integration Pattern	Write-to-IMDG	CDC (read) to IMDG
Suitable for new Apps/APIs	Yes	Not as suitable ...

Best Practice 2.2 - Corollary 2.2.1

Synch When Immediate Consistency Is Needed.

- Need for bi-directional read-write (requires some changes for scale)
- Requires integration from non-participating apps to IMDG
- Allows new apps/APIs to directly write to the data
- Once all apps use the IMDG, you can just turn off the integration

CDC is Simpler, But Doesn't Support New Projects.

- Change all the reads, leave writes the same
- Not immediately consistent (Latency of database write and CDC read)
- CDC doesn't allow new apps or APIs to write to the data
- CDC might require write changes together at the end ...



Best Practice 2.3



FREE YOUR DATA! NOW!

When to Redesign and Open Up Your Data

- For new APIs
 - Enables speed and scale without impacting existing apps
- For HTAP
 - Enables new analytics to be performed against data
- For future projects
 - Make sure to plan ahead to support future projects
 - May provide a better ROI
- To simplify integration



Best Practice 2.4

REDESIGN FOR SPEED AND SCALE

Best Practices for Data Redesign

- Redesign for speed and scale
 - Partition data for horizontal scale
 - Optimize SQL for collocation (distributed joins)
 - Remove PL/SQL (which is focused on scale-up)
 - Collocate logic with data for speed (including PL/SQL)
 - Design for scale for other uses as well (HTAP, APIs)
- Design with future projects in mind to keep speed and scale
 - Don't make the same changes twice ... try to redesign now for future data merges
 - Look at what your APIs will require



Best Practice 2.5

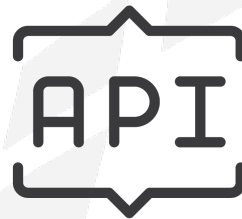
DESIGN WITH APIS IN MIND



Mobile



IoT



API



SaaS

Gartner Strategic Planning Assumptions

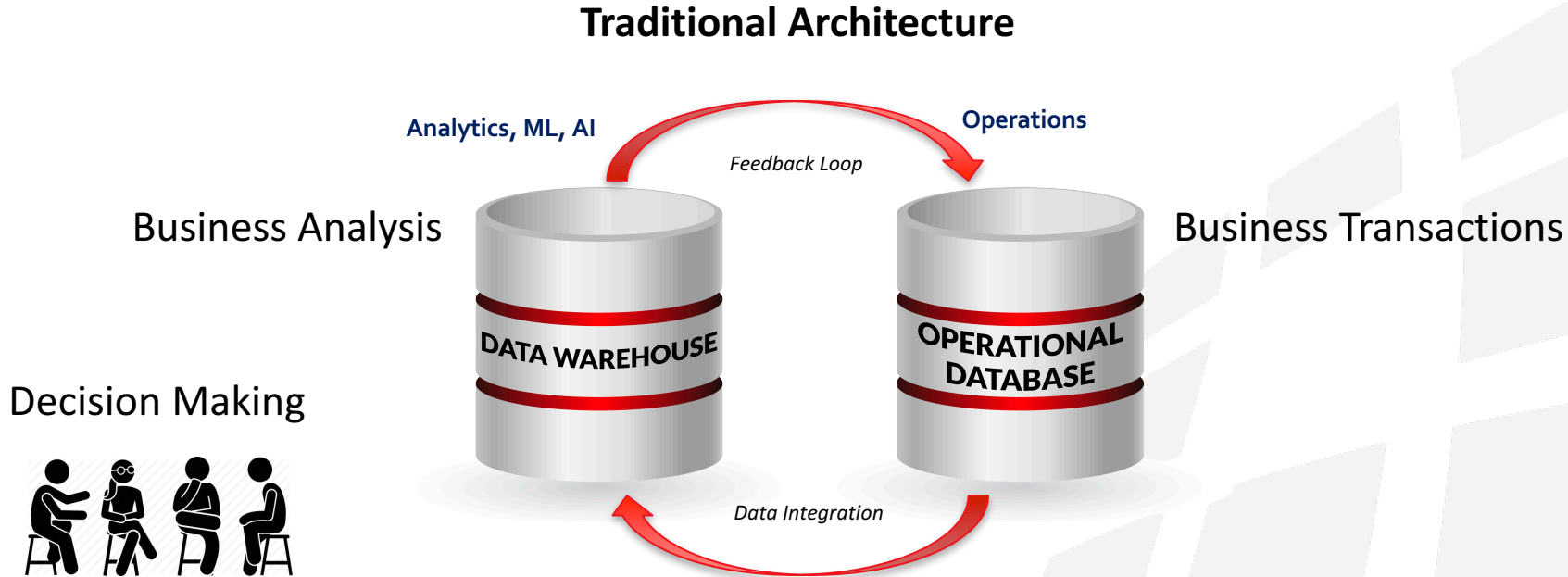
- By 2019, **75% of cloud-native application development will utilize IMC or services that use IMC**, thus enabling mainstream developers to implement high-scale/high-performance applications
- By 2021, at least **25% of large and global organizations will adopt platforms combining multiple in-memory technologies** to reduce their IMC infrastructure complexity
- By 2022, **40% of large and global enterprises will be using IMDBMSs** to reduce the proliferation of physical data stores in the publication of data

Best Practice 2.5: Design with APIs in Mind

- Many new projects use APIs
 - B2B, Social, Mobile, Cloud integration
- IMDGs are the ideal data layer for APIs
 - Easily merge data needed by an API
 - Provide low latency
 - Support key-value (JSON) and SQL
 - Support Docker/Kubernetes (and microservices)
 - Support API security (by supporting APIs)
 - Collocate code with data for speed and HTAP

When is the Best Time to Have Data About the Customer?

Traditional IT Architectures Will Struggle to Meet the Needs of the Digital Enterprise





Best Practice 2.6

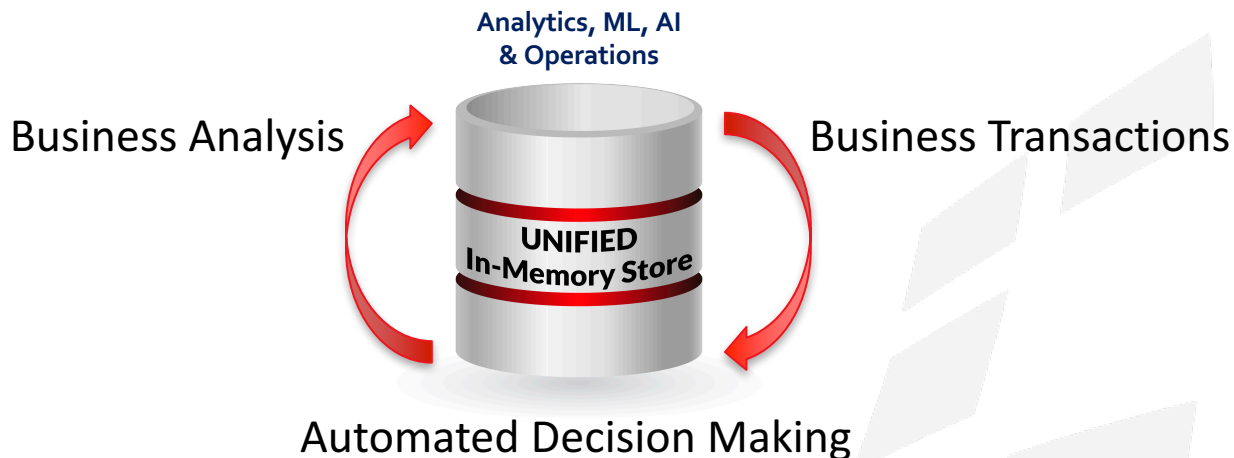
DESIGN WITH HTAP IN MIND

IMC Platforms will Power the Digital Enterprise

(IMC supports HTAP - Hybrid Transaction/Analytical Processing)

"IMC-enabled HTAP can have a transformational impact on the business." — Gartner 2/17

Unified IMC HTAP Architecture



IMC-enabled HTAP enables situational awareness on live operational data as opposed to after-the-fact analysis on stale data (in traditional architectures)

Best Practice 2.6: Design with HTAP in Mind

- Identify where real-time analytics are needed
 - Real-time analytics during transactions (e.g. profiling, promotions)
 - Decision automation (including machine/deep learning)
- Identify what data is needed
- Identify the old analytics, and the projects
 - Start to consolidate efforts

In-Memory Computing Best Practices Series

2. Add speed and scale to existing applications
 0. Use money from scale-up hardware to invest in the right architecture
 1. Build a horizontal-scale IMDG to add speed and scale
 2. Choose the right integration architecture (Synch or CDC)
 3. Free your data! Now!
 4. Redesign for speed and scale
 5. Design with APIs in mind
 6. Design with HTAP in mind

In-Memory Computing Best Practices Series

1. Build an in-memory computing foundation and roadmap
2. Add speed and scale to existing applications
3. Develop new applications, channels and APIs
4. Make the bridge to the cloud
5. Implement HTAP and real-time data management
6. Become event- and stream-driven
7. Think different with machine and deep learning





THANK YOU!

IN-MEMORY COMPUTING BEST PRACTICES

Adding Speed and Scale to Existing Applications

Rob Meyer
Outbound product management

rob.meyer@gridgain.com

GridGain Resources

- Webinars
 - Visit <https://www.gridgain.com/resources/webinars>
 - Visit <https://www.imcsummit.org/>
- White Papers
 - Visit <https://www.gridgain.com/resources/papers>
- Videos
 - Visit <https://www.gridgain.com/resources/videos>
- Free 30-Day Ultimate, Enterprise or Professional Edition Trial
 - Visit <https://www.gridgain.com/resources/download>