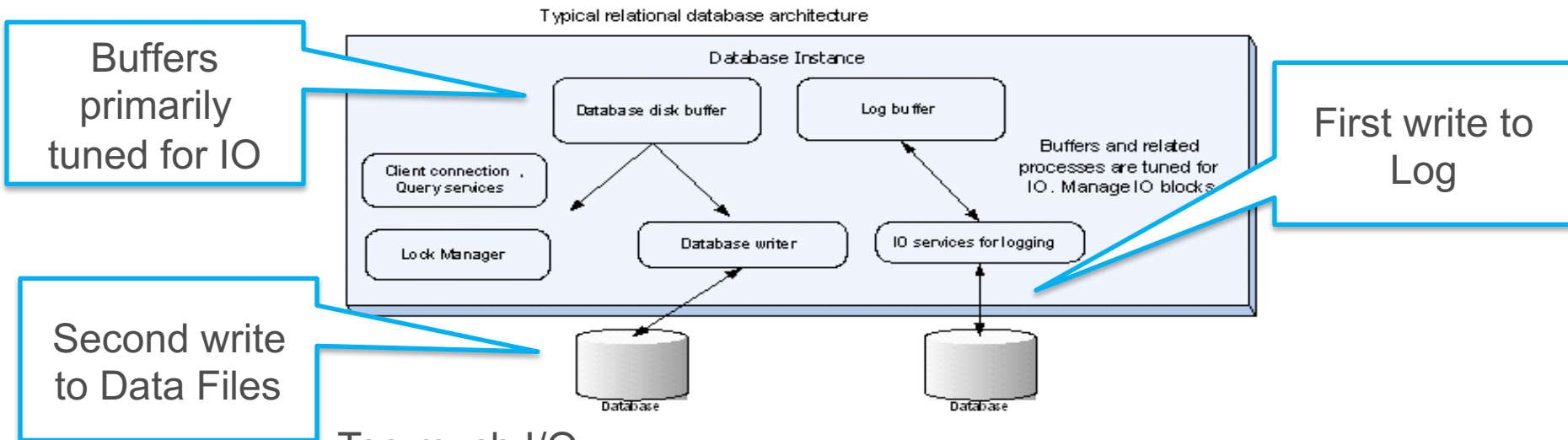


# Building scalable applications using Pivotal GemFire/Apache Geode

**Yogesh Mahajan**  
ymahajan@apache.org

# Eliminate disk access in the real time path

## We Challenge the traditional RDBMS design NOT SQL



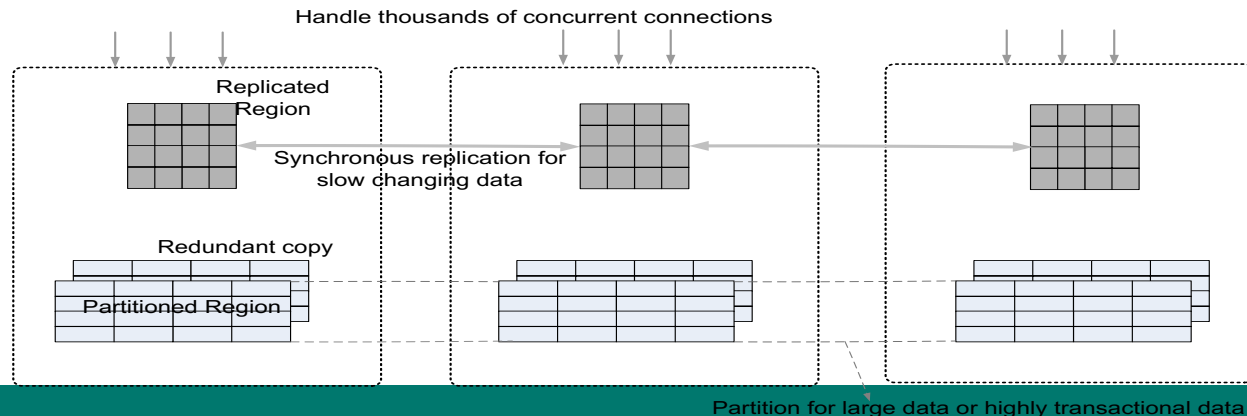
Too much I/O

Design roots don't necessarily apply today

- Too much focus on ACID
- Disk synchronization bottlenecks

# IMDG basic concepts

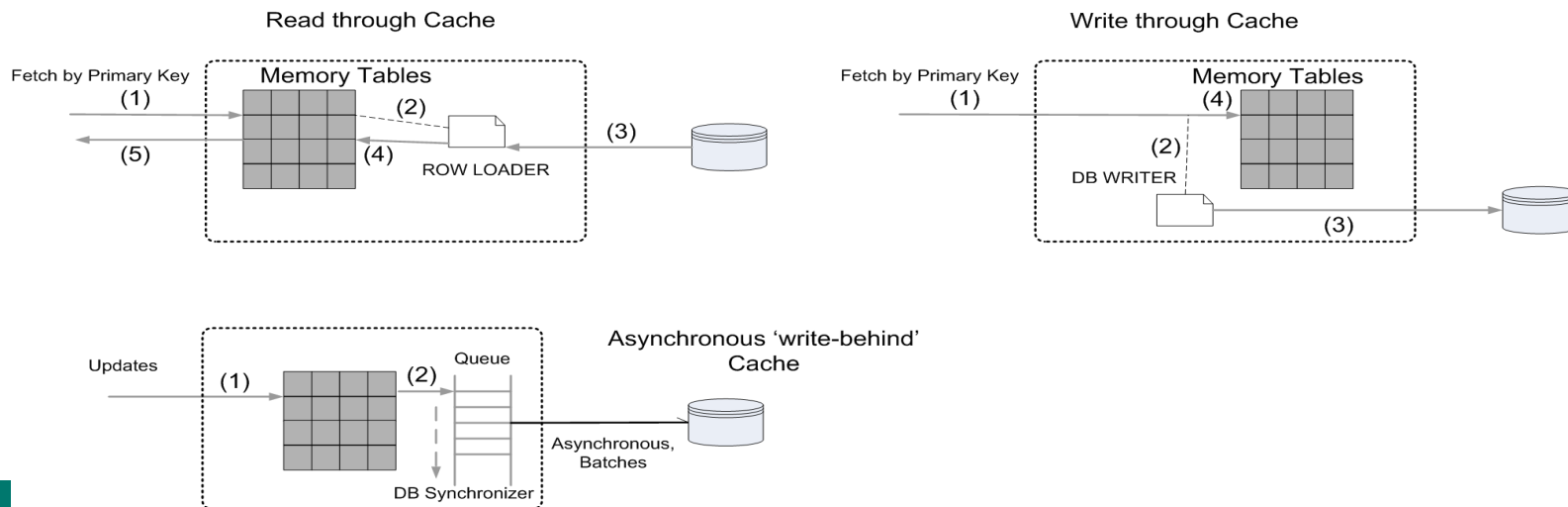
- Distributed memory oriented store
  - KV/Objects or JSON
  - Queryable, Indexable and transactional
- Multiple storage models
  - Replication, partitioning in memory
  - With synchronous copies in cluster
  - Overflow to disk and/or RDBMS
- Parallelize Java App logic
- Multiple failure detection schemes
- Dynamic membership (elastic)
- Vendors differentiate on
  - Query support, WAN, events, etc



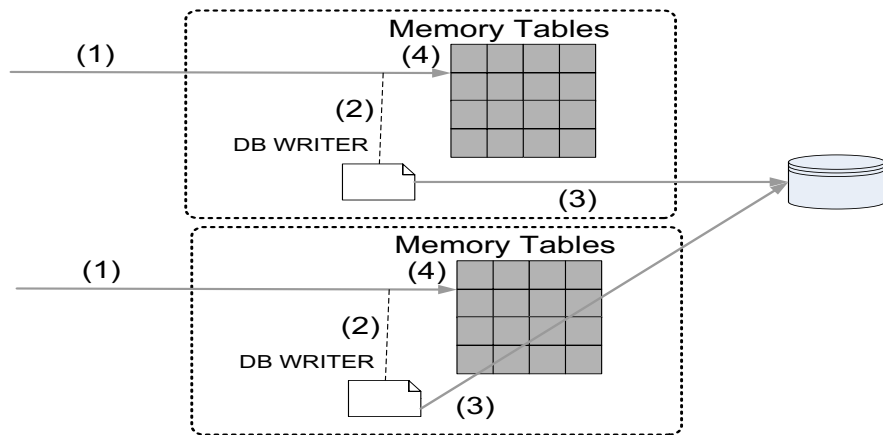
Low latency for  
thousands of  
clients

# Key IMDG pattern - Distributed Caching

- Designed to work with existing RDBs
  - Read through: Fetch from DB on cache miss
  - Write through: Reflect in cache IFF DB write succeeds
  - Write behind: reliable, in-order queue and batch write to DB

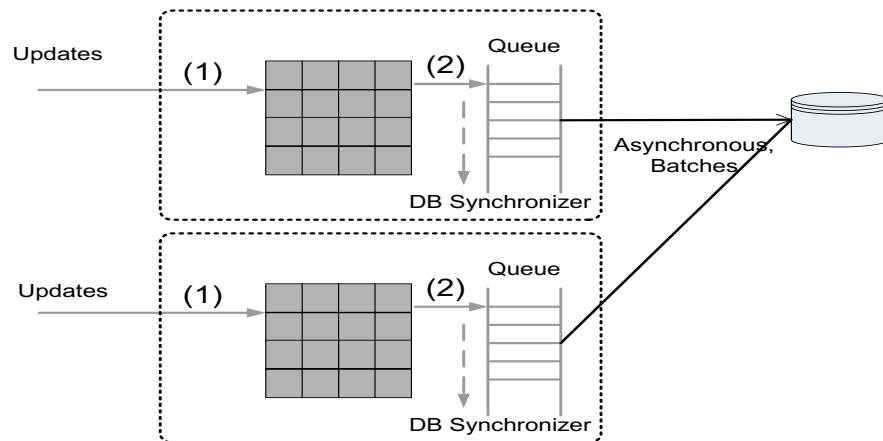


# Traditional RDB integration can be challenging



## Synchronous “Write through”

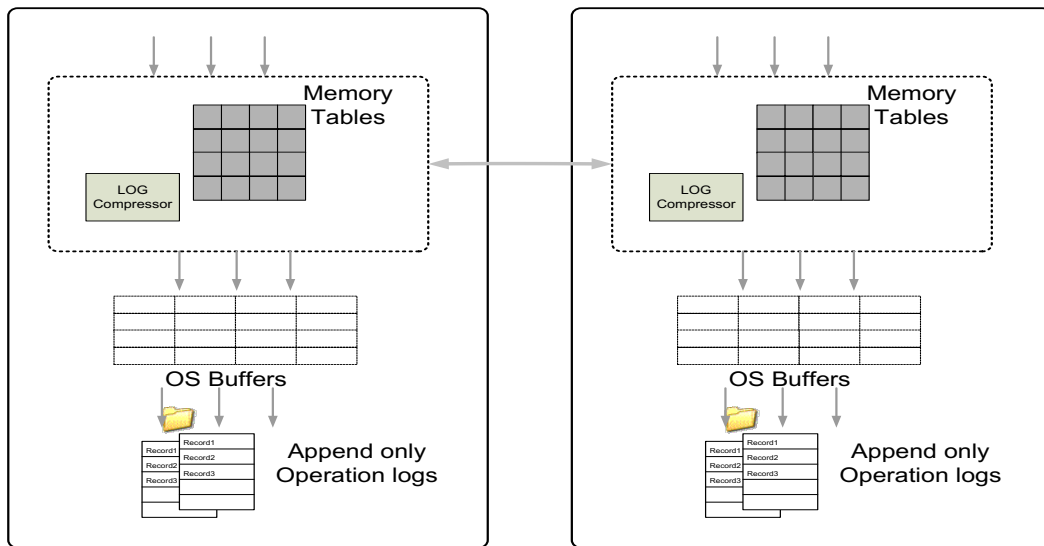
Single point of bottleneck and failure  
Not an option for “Write heavy”  
Complex 2-phase commit protocol  
Parallel recovery is difficult



## Asynchronous “Write behind”

Cannot sustain high “write” rates  
Queue may have to be persistent  
Parallel recovery is difficult

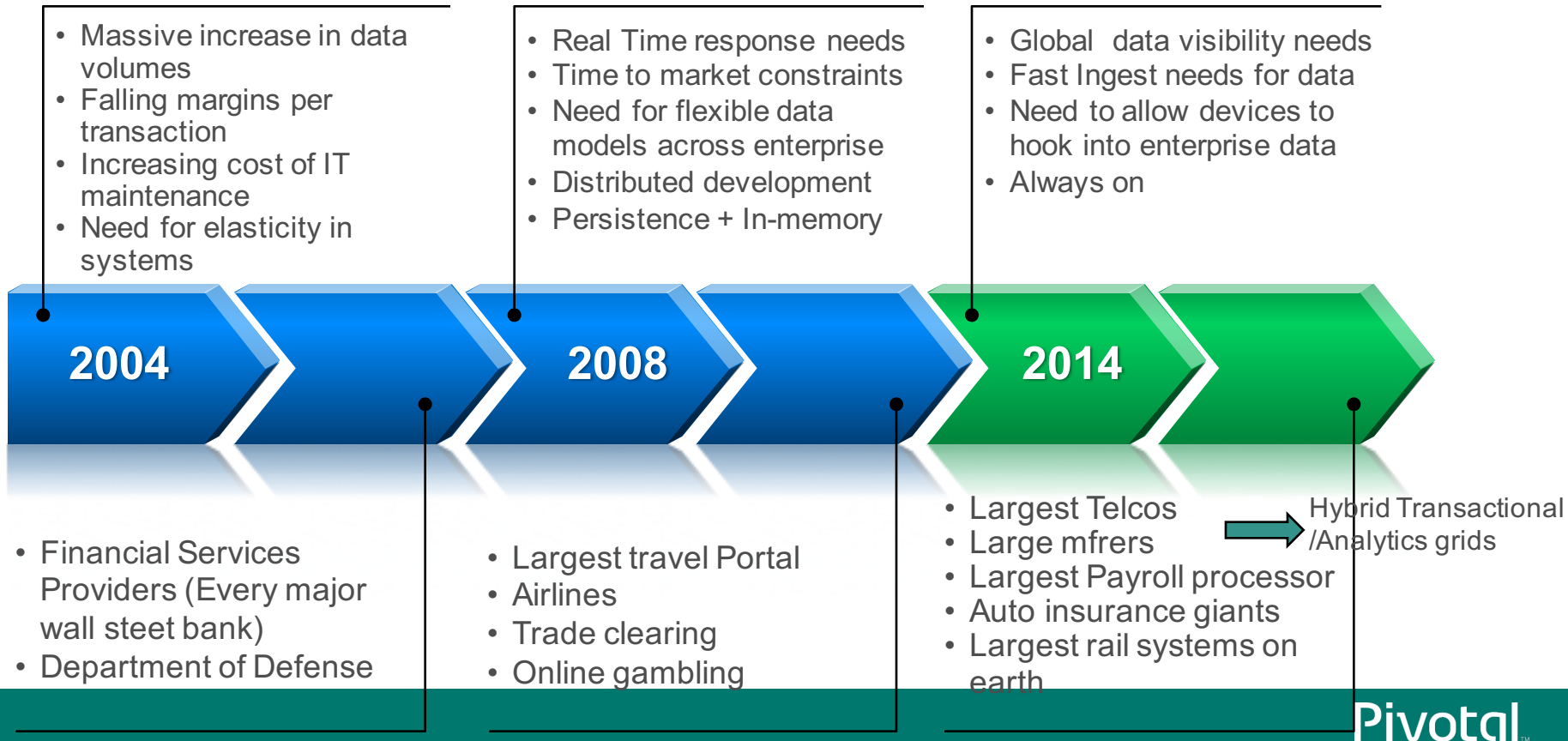
# Some IMDG, NoSQL offer ‘Shared nothing persistence’



- Append only operation logs
- Fully parallel
- Zero disk seeks
- But, cluster restart requires log scan
- Very large volumes pose challenges



# Our GemFire Journey Over The Years



# Why OSS? Why Apache?

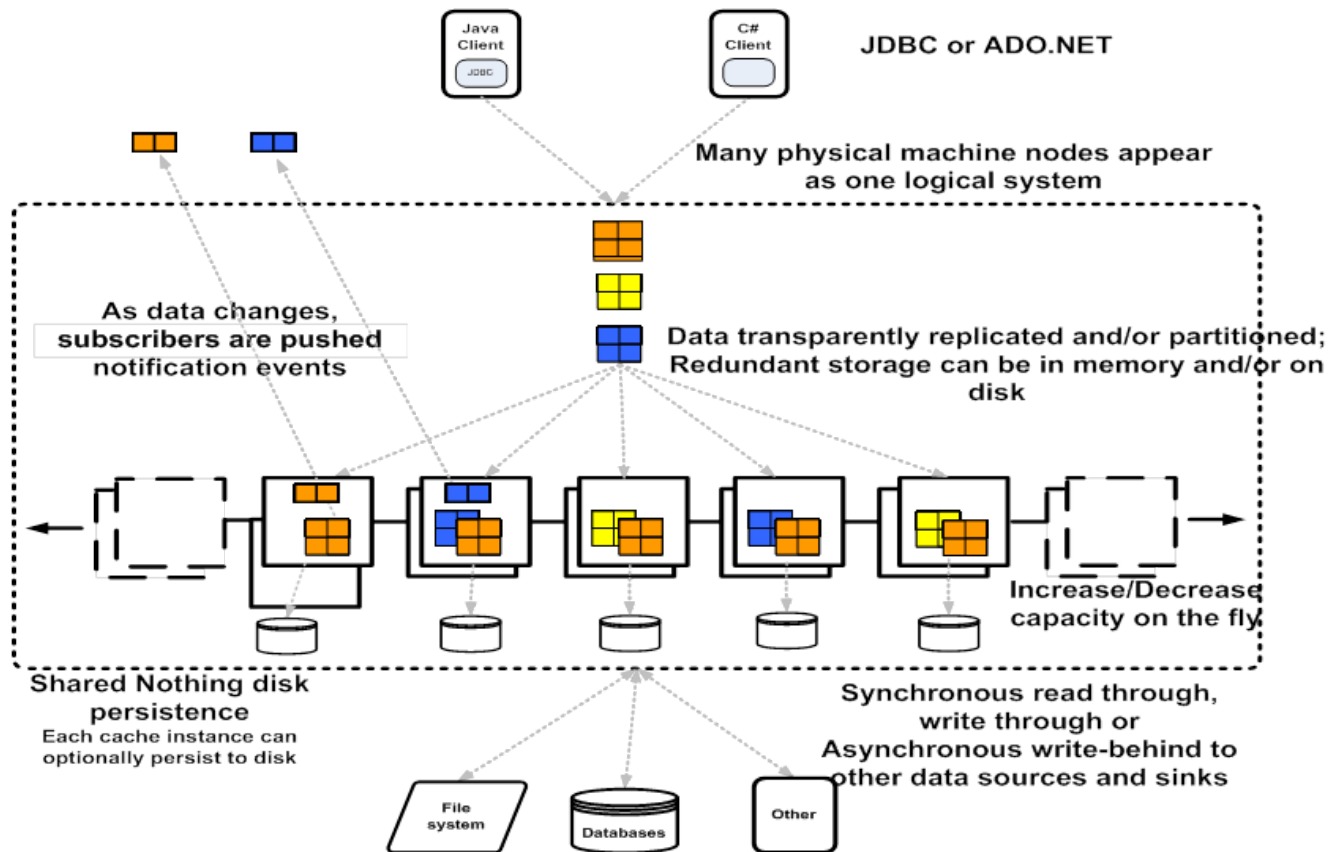
- Open Source Software is fundamentally changing buying patterns
  - Developers have to endorse product selection (No longer CIO handshake)
  - Community endorsement is key to product visibility
  - Open source credentials attract the best developers
  - Vendor credibility directly tied to street credibility of product
- Align with the tides of history
  - Customers increasingly asking to participate in product development
  - Resume driven development forces customers to consider OSS products
  - Allow product development to happen with full transparency
- Apache is where you go to build Open Source street cred
  - Transparent, meritocracy which puts developers in charge



# Geode Will Be A Significant Apache Project

- Over a 1000 person years invested into cutting edge R&D
- 1000+ customers in very demanding verticals
- Cutting edge use cases that have shaped product thinking
- Tens of thousands of distributed, scaled up tests that can randomize every aspect of the product
- A core technology team that has stayed together since founding
- Performance differentiators that are baked into every aspect of the product

# Gemfire High Level Architecture



# What makes it fast?

- Minimize copying
  - Clients dynamically acquire partitioning meta data for single hop access
  - Avoid JVM memory pools to the extent possible
- Minimize contention points .. avoid offloading to OS scheduler
  - Highly concurrent data structures
  - Efficient data transmission – Nagle's Algorithm
- Flexible consistency model
  - FIFO consistency across replicas but NO global ordering across threads
  - Promote single row transactions (i.e no transactions)

# What makes it fast?

- Avoid disk seeks
  - Data kept in Memory – 100 times faster than disk
  - Keep indexes in memory, even when data is on disk
  - Direct pointers to disk location when offloaded
- Tiered Caching
  - Eventually consistent client caches
  - Avoid Slow receiver problems
- Partition and parallelize everything
  - Data. Application processing (procedures, callbacks), queries, Write behind, CQ/Event processing

# “low touch” Usage Patterns

## HTTP Session management

Simple template for TCServer, TC, App servers  
Shared nothing persistence, Global session state

## Hibernate L2 Cache plugin

Set Cache in hibernate.cfg.xml  
Support for query and entity caching

## Memcached protocol

Servers understand the *memcached* wire protocol  
Use any *memcached* client

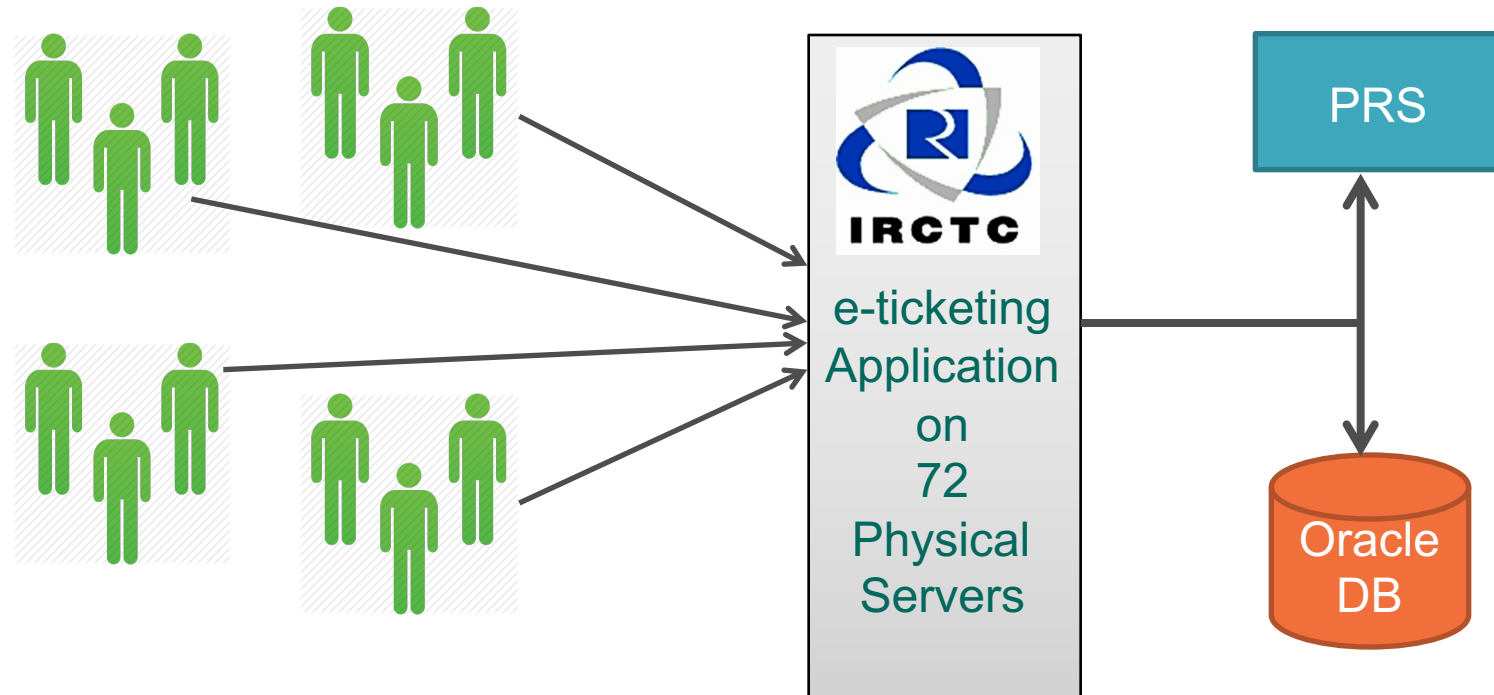
## Spring Cache Abstraction

```
<bean id="cacheManager"  
class="org.springframework.data.gemfire.support.GemfireCacheManager"
```

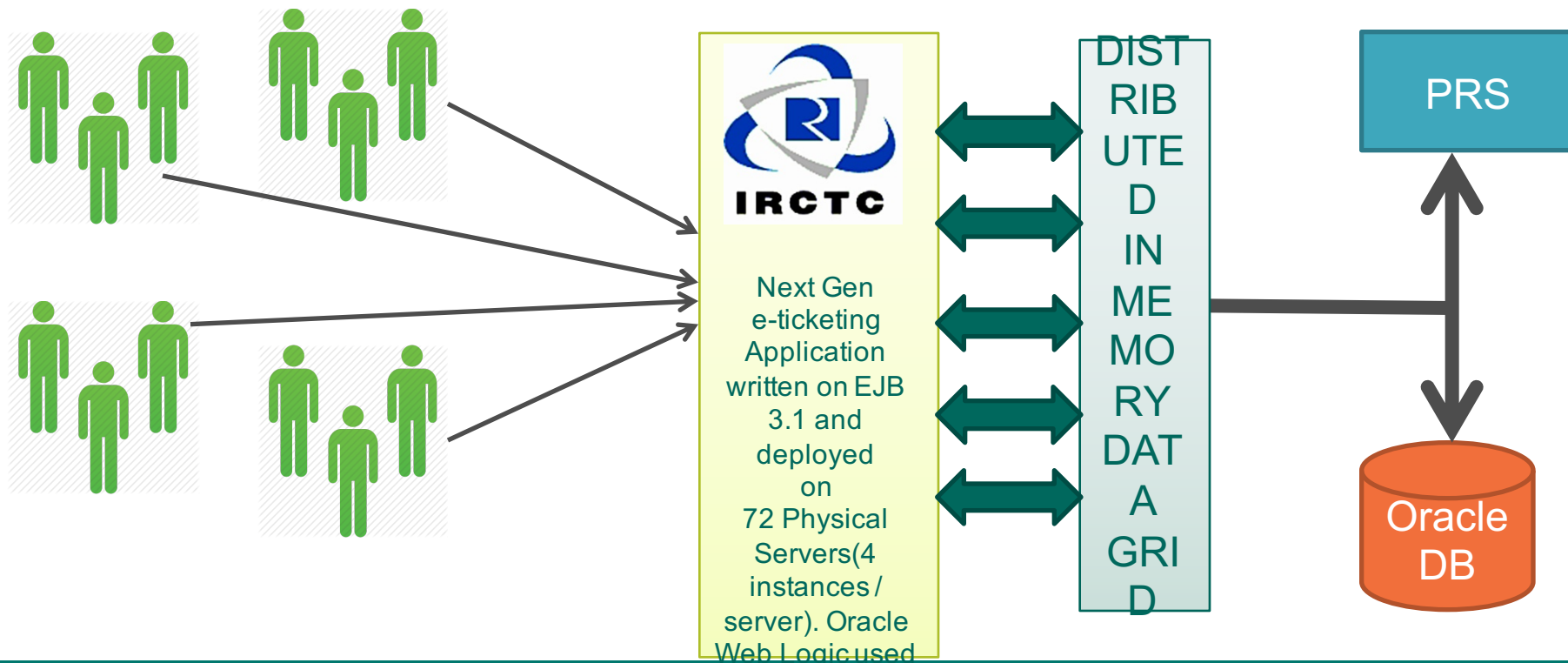
# A GemFire customer use case : IRCTC

- World's second largest railway network, 7000 stations, 30 million users, 12000 trains
- Longer queues at railway booking counters
- Not able to scale during peak hours, 8AM, 10AM
- System designed back in 2005/2006
- Frequent downtimes, more than 10 mins delay to book a ticket, or timeout.

# Old Architecture



# Architecture Using GemFire

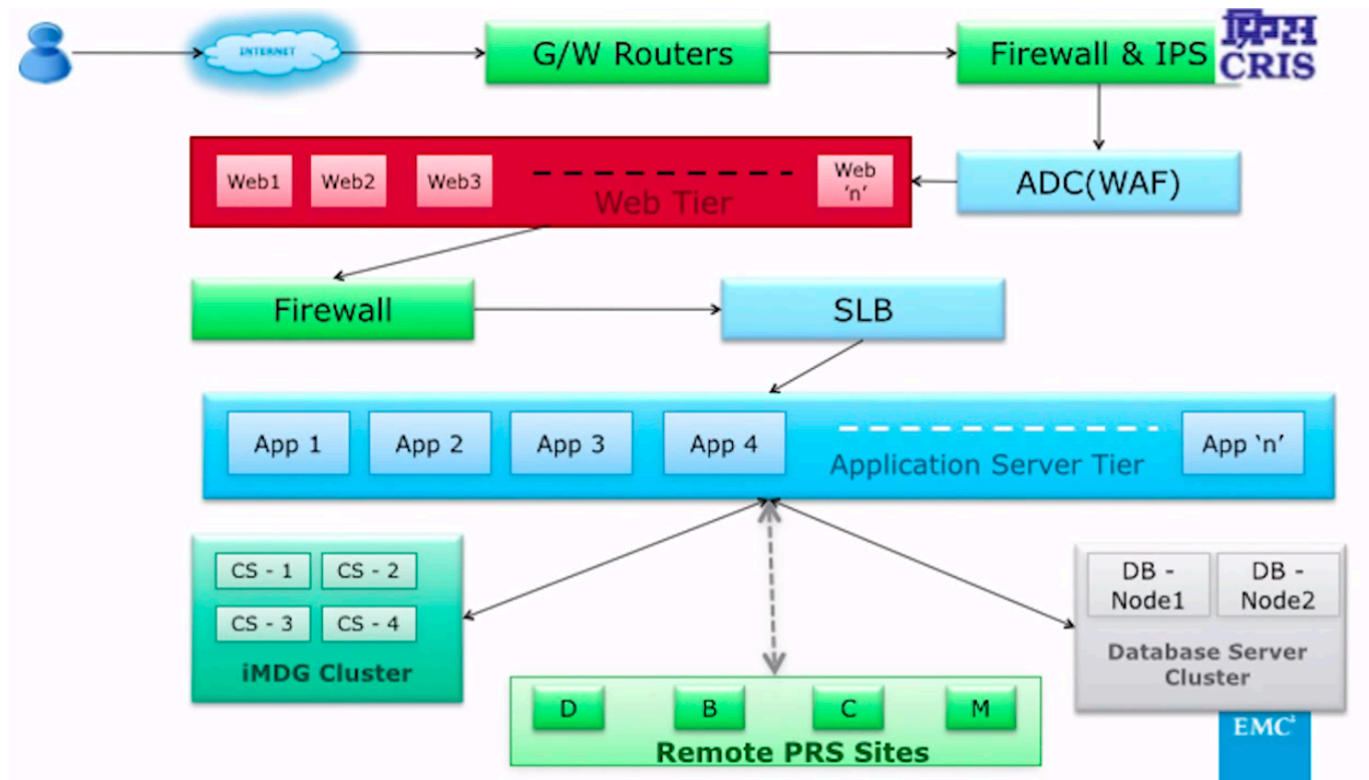




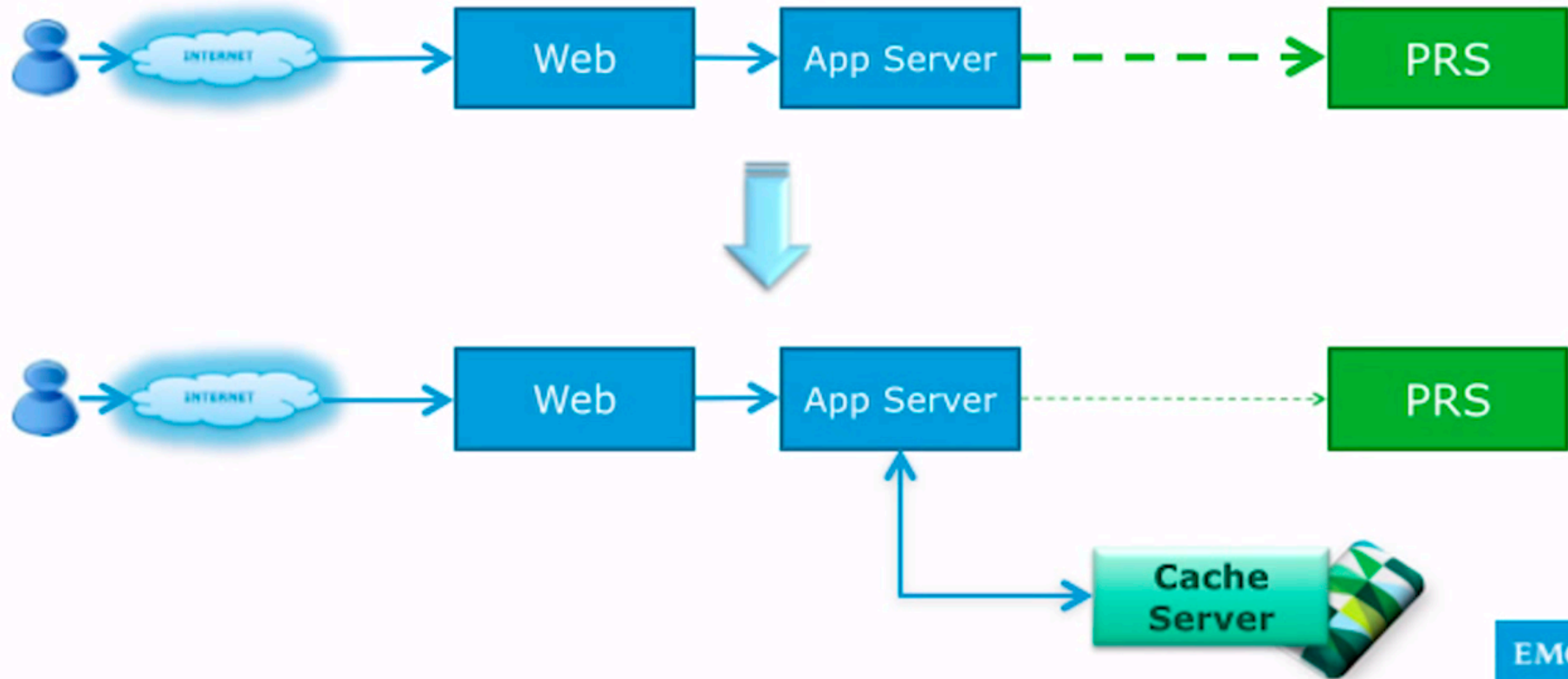
# Challenges

- Social infrastructure site
- Migrating 30 million registered users
- Booking transaction checkpoints because of supply demand gaps
- Journey Planner, user authentication migration to in memory
- Capable of scaling up as the demand increases in future.
- High number of concurrent users at the peak times

# Architecture Using GemFire



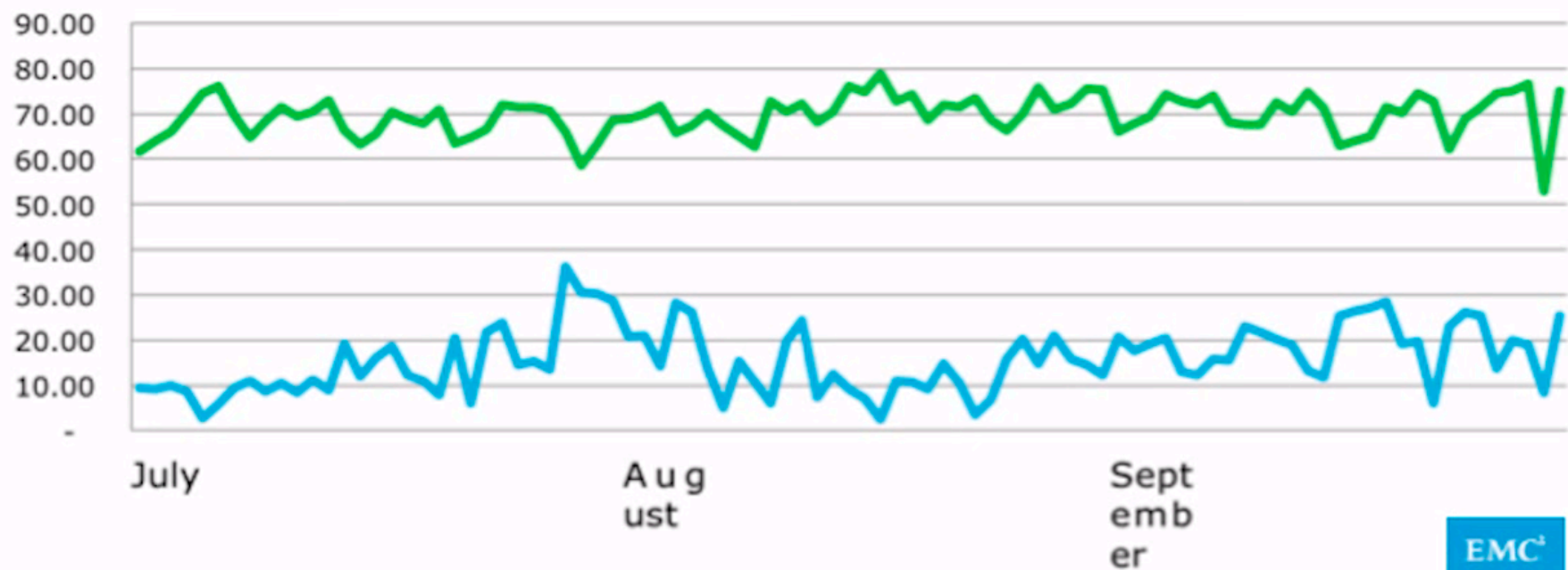
# Journey Planning & Availability Enquiry



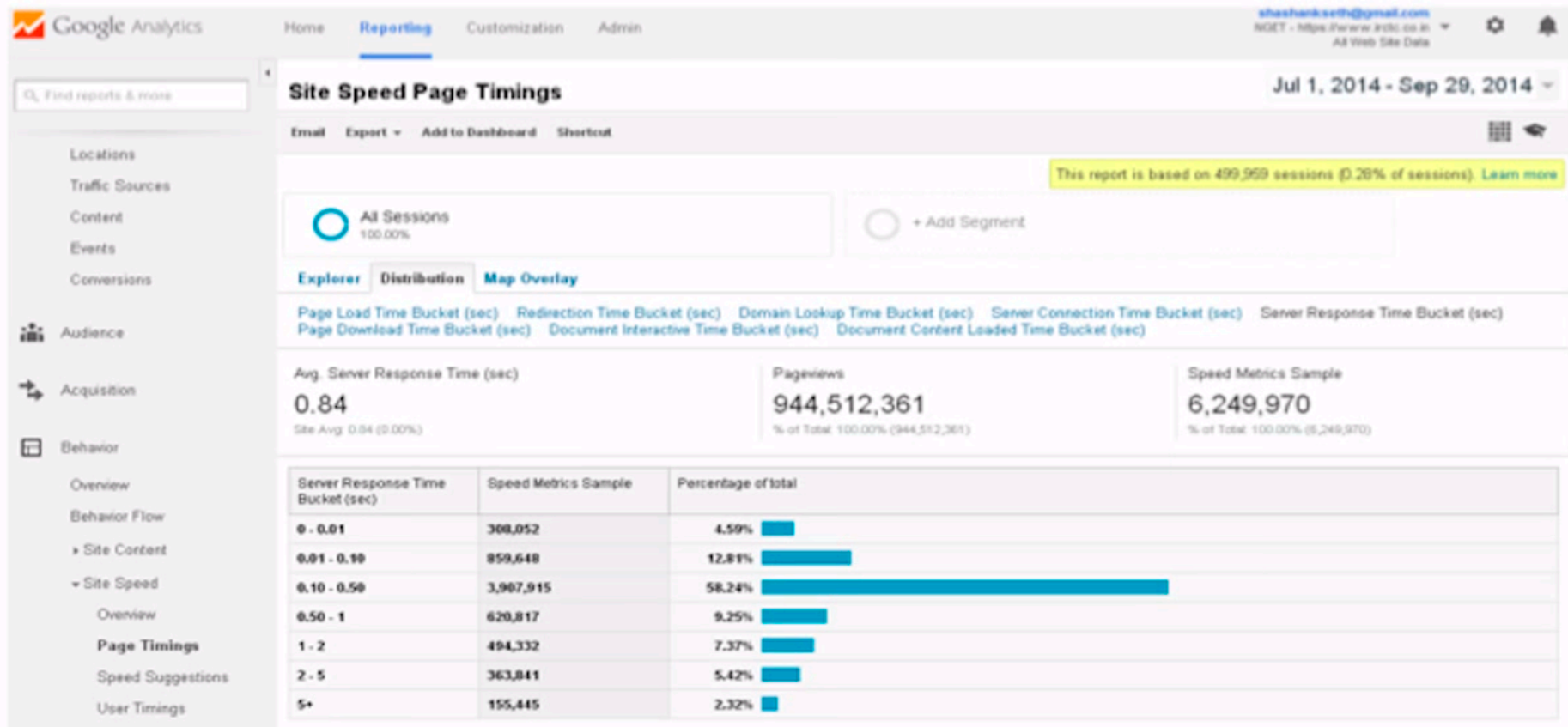
# Increase in e-ticket booking In initial 10 minutes

**10:00 - 10:10 A.M.**

% Internet Ticket 2013    % Internet Ticket 2014



# Average response time < 1 sec



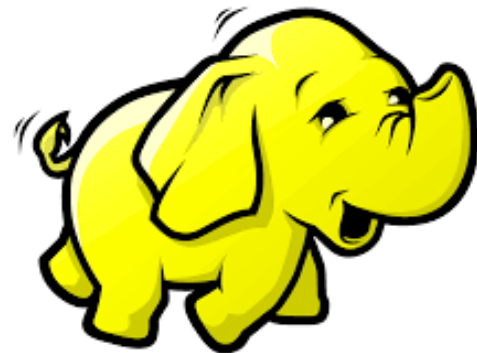
# Benefits

- Supports More than 200,000 Concurrent Purchases
- Provide Stable Performance to Book Approximately 150,000 TPH, Compared to 60,000 in the Old System
- Transformed Customer Experience so Reservation Transactions Complete in Seconds Instead of 15 minutes
- Shifted Online Purchasing From 50% of Tickets Sold to 65%
- Boosting Revenue Generated From E-ticket Sales to INR600 Million Daily
- Capable of scaling up as the demand increases in future.
- CPU Usage during peak hours (**Tatkaal**) is less than 9%

# Roadmap

- HDFS persistence
- Off-heap storage
- Lucene indexes
- Spark integration
- Cloud Foundry service
- Distributed Transactions

...and other ideas from the Geode community!



Performance is key.  
Consistency is a must.

Geode is the open source distributed, in-memory  
database for scale-out applications.

[geode.incubator.apache.org](http://geode.incubator.apache.org)



# Geode community

A decorative background consisting of numerous triangles of various colors (blue, purple, grey) arranged in a circular pattern, resembling a stylized sunburst or a cluster of flags.

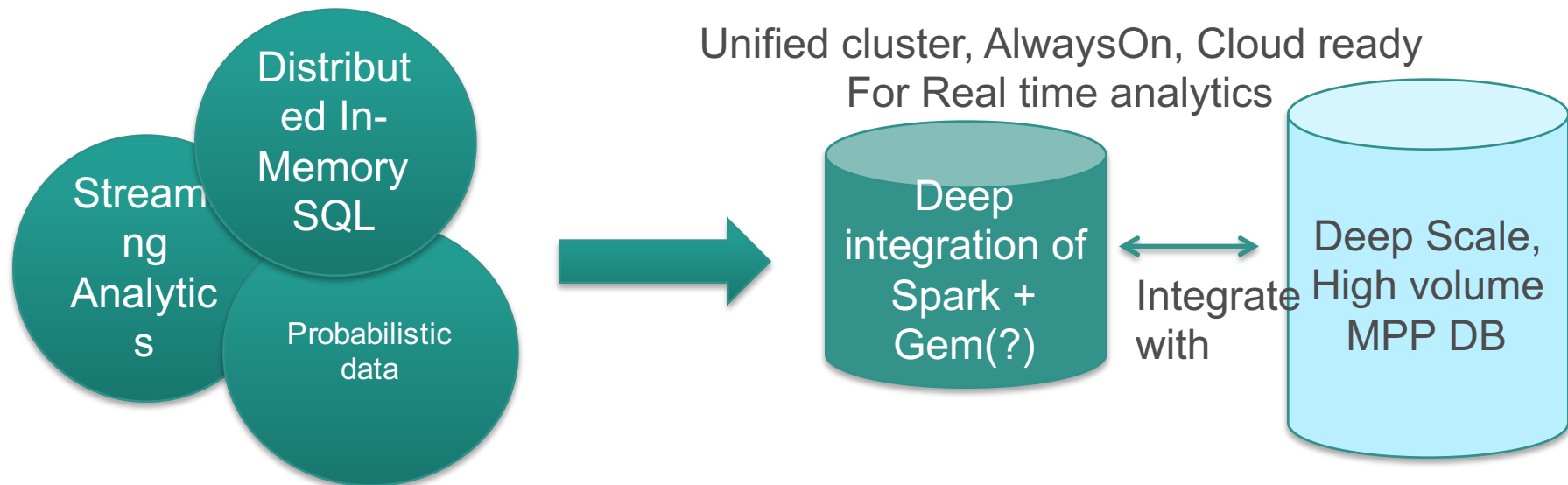
- <http://geode.incubator.apache.org>
- [dev@geode.incubator.apache.org](mailto:dev@geode.incubator.apache.org)
- [user@geode.incubator.apache.org](mailto:user@geode.incubator.apache.org)
- <http://github.com/apache/incubator-geode>

## Our in-memory computing journey



- We started GemFire team in Pune in 2005, the core team remains the same over the last decade
- We build a new product out of Pune , GemFire XD, In memory distributed SQL with GemFire and Apache Derby.
- We are now working on a new initiative, SnappyData.io, a startup funded by Pivotal, building a product based on Spark(Streaming/SQL), GemFire and Approximate Query Engine.
- And we are hiring

# SnappyData Positioning (snappydata.io)



Vision – Drastically reduce the cost and complexity in modern big data. ...Using fraction of the

resources  
10X better response time, drop resource cost 10X,  
reduce complexity 10X