IBM Software Group

Introduction to the WebSphere XD ObjectGrid

Alan Chambers
IBM Consulting IT Specialist
alan_chambers@uk.ibm.com

WebSphere User Group (UK)

4th March 2008

Introduction

- **Do you…**
  - Have applications that require very high speed access to large quantities of data, for which traditional databases simply aren't fast enough?
  - Need to cache data from a database or other back-end that needs to be accessed at speeds that would otherwise be unattainable?
  - Have a compute intensive application that can be broken up into multiple parts, executing in parallel on different data?
  - Need to propagate application data rapidly between remote locations faster than traditional database replication?
- **Even if not, do you…**
  - like really cool technology?
  - like having something technically interesting to play with?
  - admire software that is small and has no UI and no pre-reqs?

Drivers for ObjectGrid

- Businesses are demanding from applications:
  - faster response times
  - more consistent response times
  - linear scaling
- Classic stateless approaches really just push the problem somewhere else, i.e. a database or backend system.
- With normal cache approaches:
  - wasteful copies of cached data stored in every server
  - caches require invalidation logic to remove stale copies
  - servers always start with a cold cache and as a result spike the backend at startup.
- Businesses are requiring constant millisecond level responses as applications scale from a couple of servers to hundreds or even thousands of servers.

Some industry trends

“Extreme transaction processing applications are characterized by exceptionally demanding requirements and complex, distributed architectures.”

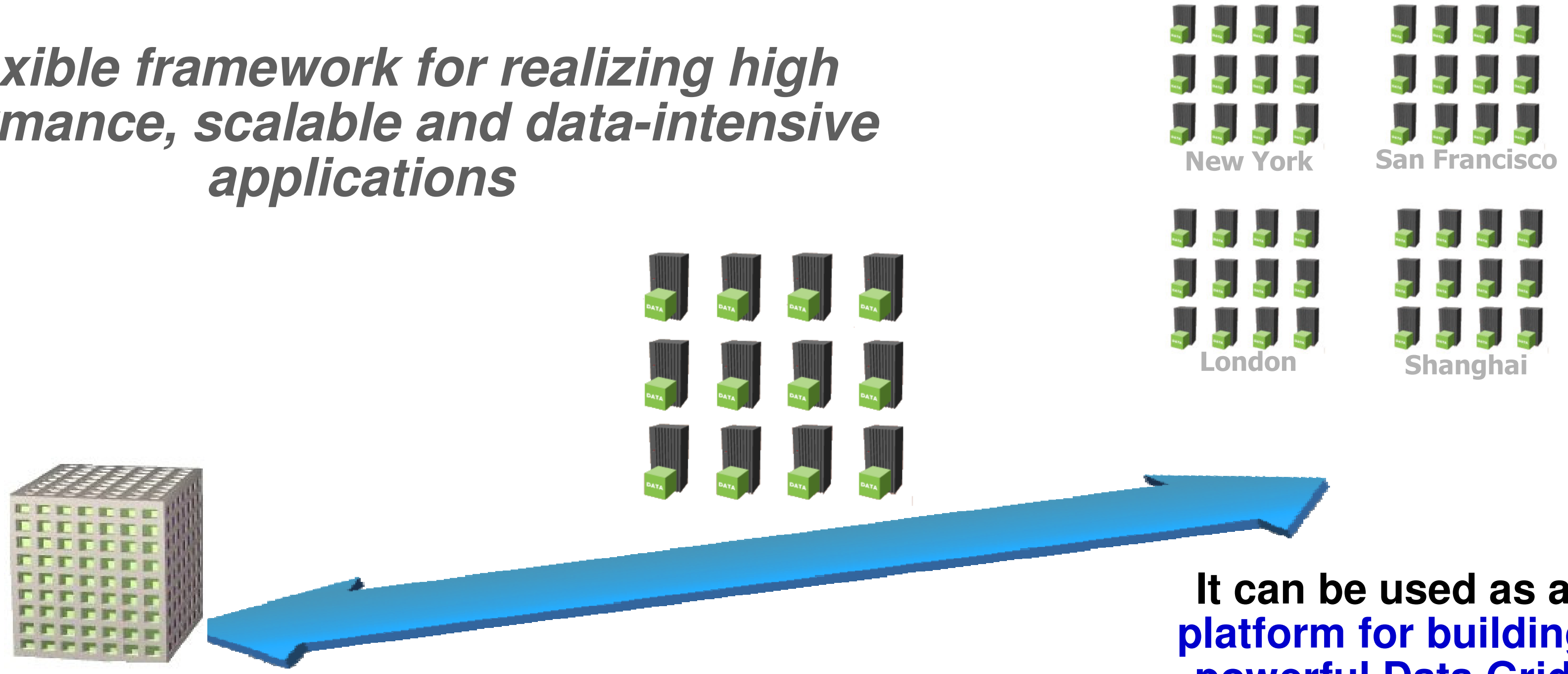
Gartner, 13th February 2007

“…we believe that OLTP should be considered a main memory market, if not now then within a very small number of years. Consequently, the current RDBMS vendors have disk oriented solutions for a main memory problem. In summary, 30 years of Moore’s law has antiquated the disk-oriented relational architecture for OLTP applications.”

Stonebraker et al, 2008

What is ObjectGrid?

A flexible framework for realizing high performance, scalable and data-intensive applications



It can be used as a very powerful cache that scales from simple in-process topologies to powerful distributed topologies.

It can be used as a form of in memory database to manage application state (and it scales to 1000’s of servers). This is sometimes referred to as Distributed Application State Management.

It can be used as a platform for building powerful Data Grid applications.

Last Generation in-memory databases

- Traditional in-memory databases look very conventional.
- You set up a static pair and it replicates between them
- It provides a shared memory link to allow local access to the memory.
- They leave all the hard work to the customer.
  - What if I need 500 servers?
  - Do I have to set it up myself?
  - If I add servers does it go faster?
  - Does it understand data centers etc?
  - What if I move my server to a new IP? Are clients impacted?
  - It costs how much?

ObjectGrid is different

- Advanced replication capabilities

- Grid based programming patterns (Map Reduce) are supported
  - allow the full power of the grid to be used to process large quantities of data at memory speeds with little change in response time as the grid grows.

- ObjectGrid stores transactional data in exactly one place and support N copies of various quality levels.
  - reference data distributed throughout the grid

- Business logic can also run against copies for additional scalability for read based traffic.

- Automatic policy driven placement and built-in high availability leads to massive TCO reduction compared with traditional disk or memory based databases.

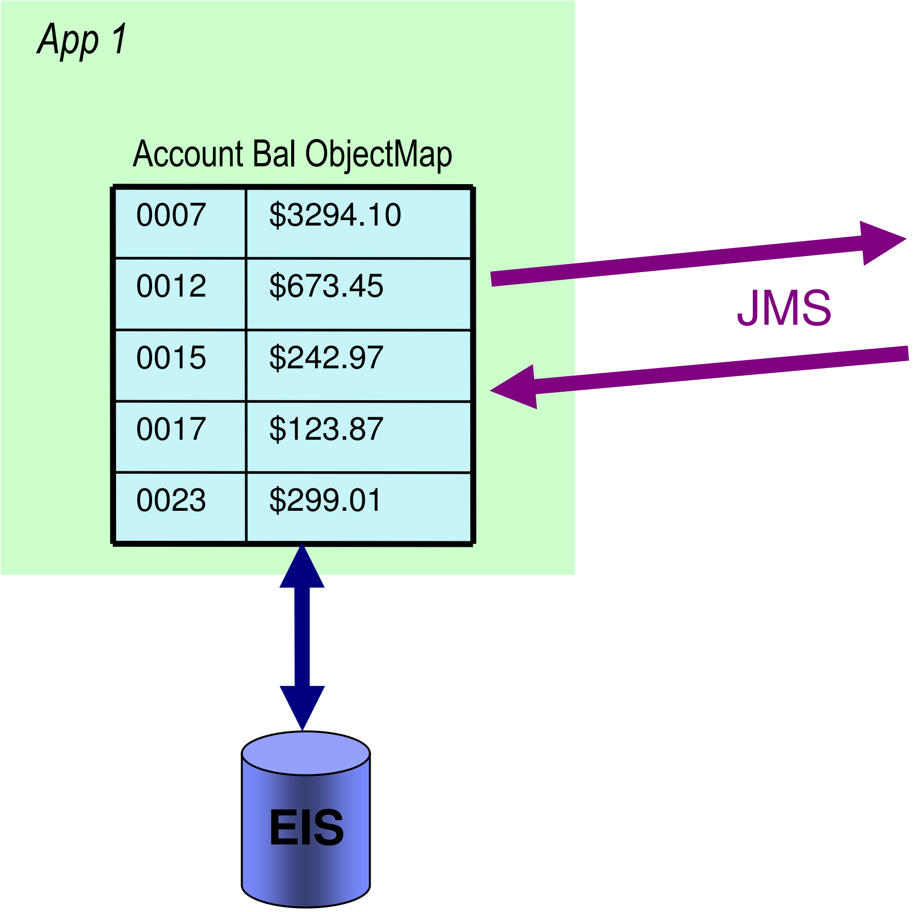
Dependencies

- It has no WebSphere dependency and works with
  - Current and older versions of WebSphere
  - competitive application servers.
  - Straight J2SE (1.4.2 or higher)
  - Eclipse RCP.
- While ObjectGrid is self contained it requires an external framework for installing applications and start/stop the JVMs hosting those applications.
  - WebSphere XD
  - WebSphere ND
  - WebLogic, JBoss
  - Third party grid management software

A closer look at ObjectGrid

ObjectGrid within a single JVM

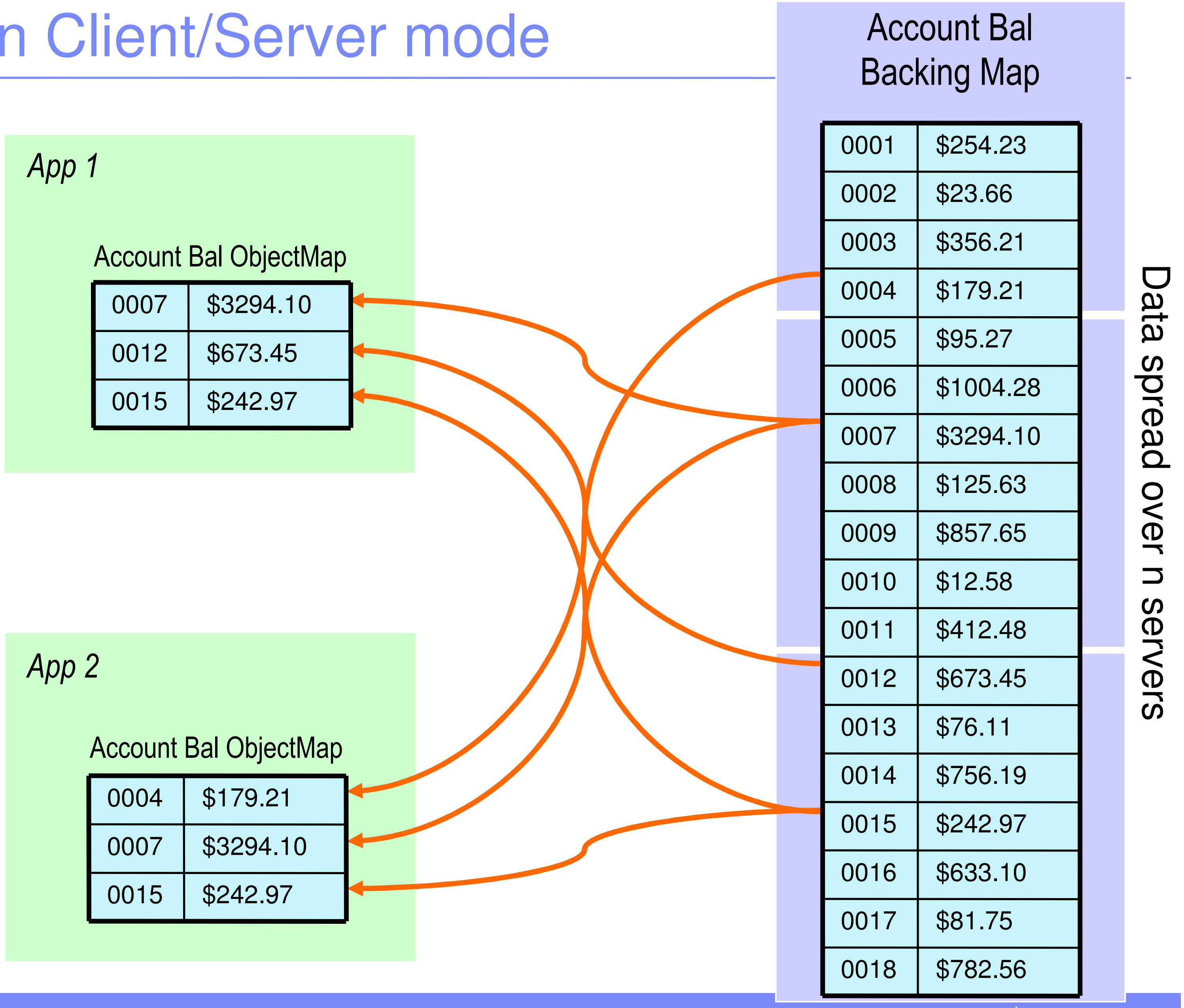
- **Data held in same process as application**
- **Can be replicated via JMS to other ObjectGrids in other processes**
- **Can be loaded from/persisted to a bac end**



Account Bal ObjectMap

0007	$3294.10
0012	$673.45
0015	$242.97
0017	$123.87
0023	$299.01

ObjectGrid in Client/Server mode



App 1

Account Bal ObjectMap

0007	$3294.10
0012	$673.45
0015	$242.97

App 2

Account Bal ObjectMap

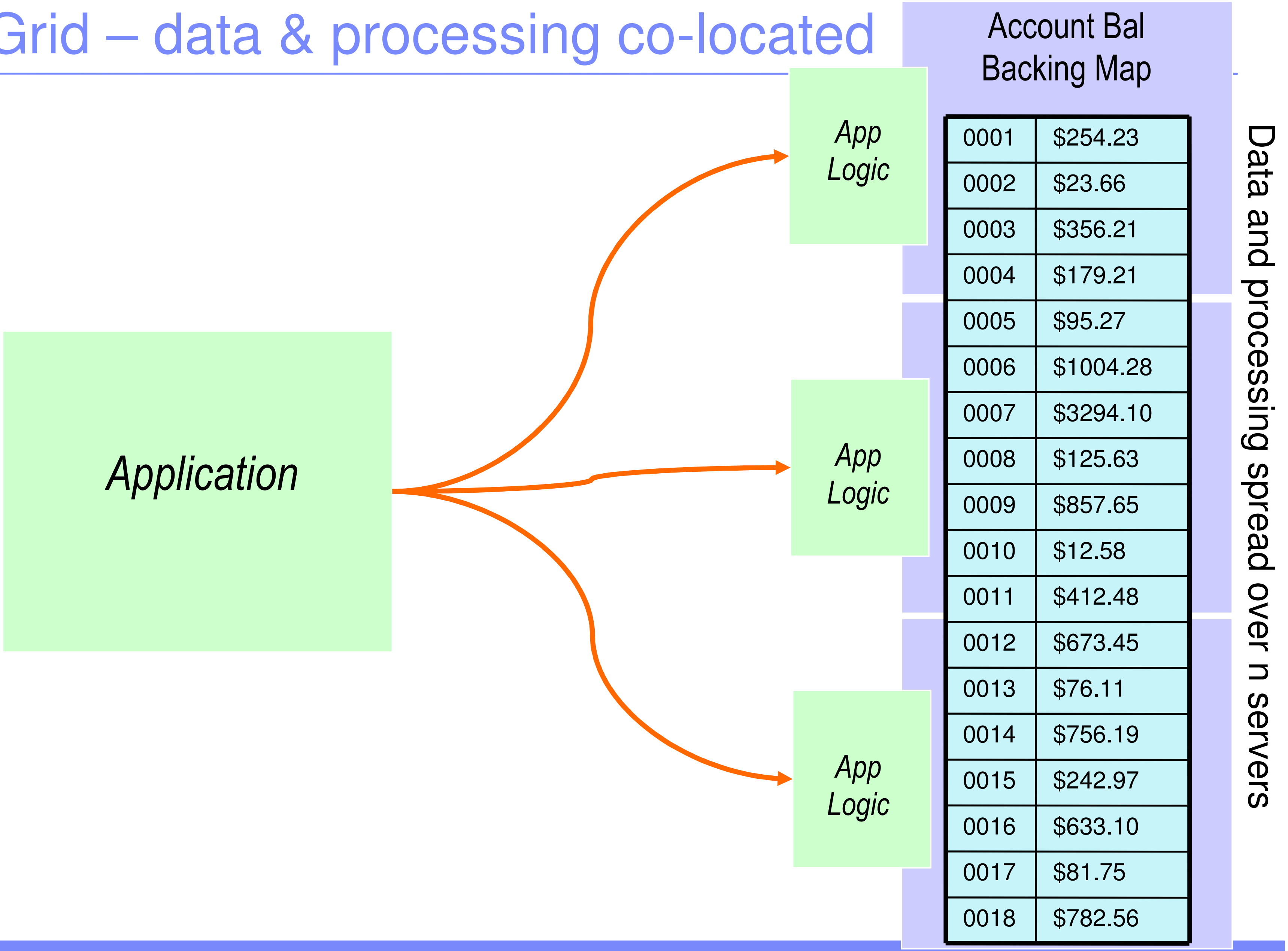
0004	$179.21
0007	$3294.10
0015	$242.97

Account Bal Backing Map

0001	$254.23
0002	$23.66
0003	$356.21
0004	$179.21
0005	$95.27
0006	$1004.28
0007	$3294.10
0008	$125.63
0009	$857.65
0010	$12.58
0011	$412.48
0012	$673.45
0013	$76.11
0014	$756.19
0015	$242.97
0016	$633.10
0017	$81.75
0018	$782.56

Data spread over n servers

DataGrid – data & processing co-located



Account Bal Backing Map

0001	$254.23
0002	$23.66
0003	$356.21
0004	$179.21
0005	$95.27
0006	$1004.28
0007	$3294.10
0008	$125.63
0009	$857.65
0010	$12.58
0011	$412.48
0012	$673.45
0013	$76.11
0014	$756.19
0015	$242.97
0016	$633.10
0017	$81.75
0018	$782.56

Data and processing spread over n servers

Ease of integration

- **It works with the application server you already have.**
  - Older versions of WebSphere.
  - Other application servers.
  - Spring Framework
- **Typically there will be existing integration points to add ObjectGrid with minimal or no impact on the existing applications.**
  - Object Relational Mapper second level cache (ORMapper, Hibernate, CMP).
  - ESB Mediation
  - HTTP server filters for session management.
- **ObjectGrid APIs are simple and easy to learn:**
  - Java Collections API
  - EntityManager (POJO with relationship) based API
- **Transactional integration**
  - ObjectGrid has hooks to allow it to integrate with third party transaction frameworks.
  - This means that the begin/commit/rollback calls are integrated with the application container transaction and do not need to be called by the application.

Partitioning and Replication

- **Data is partitioned based on the map key**
  - Hashing algorithm can default or you can provide one
  - all you have to do is specify the number of partitions
    - Too many may mean some overhead
    - Too few limits the number of ObjectGrid containers you can run

- **Data is replicated synchronously and/or asynchronously**
  - For resilience if a container (i.e. JVM) fails
  - Location of replicas is always different from the primary partition
  - Synchronous replicas always updated within the transaction
  - Asynchronous replicas update (slightly) later, so may not be 100% up to date (milliseconds usually)

- **Each Primary Partition, Sync replica and Async replica is known as a “Shard”**

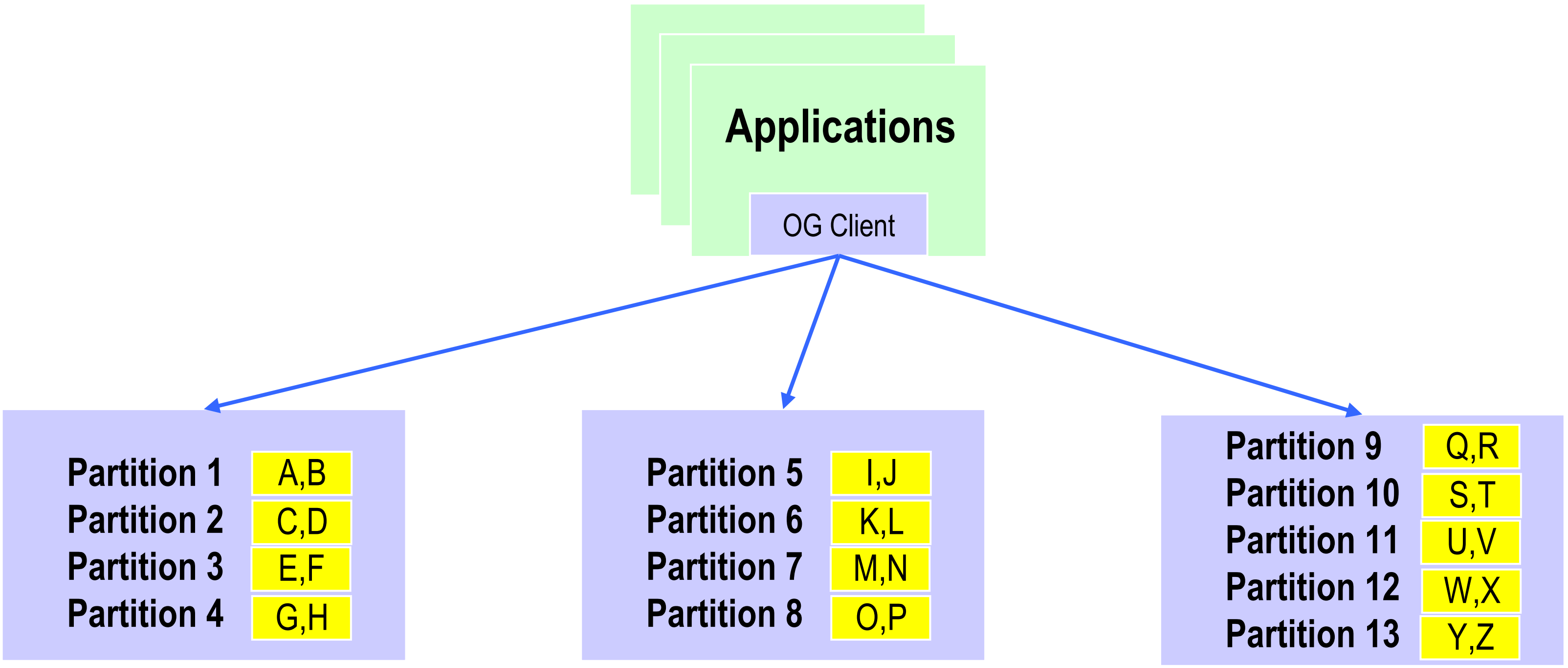
- **“Zones” support enables control of replica location**
  - E.g. for geographic failover

- **ObjectGrid tries to balance distribution of shards across available container**
  - Moving shards as necessary, as containers are added or removed

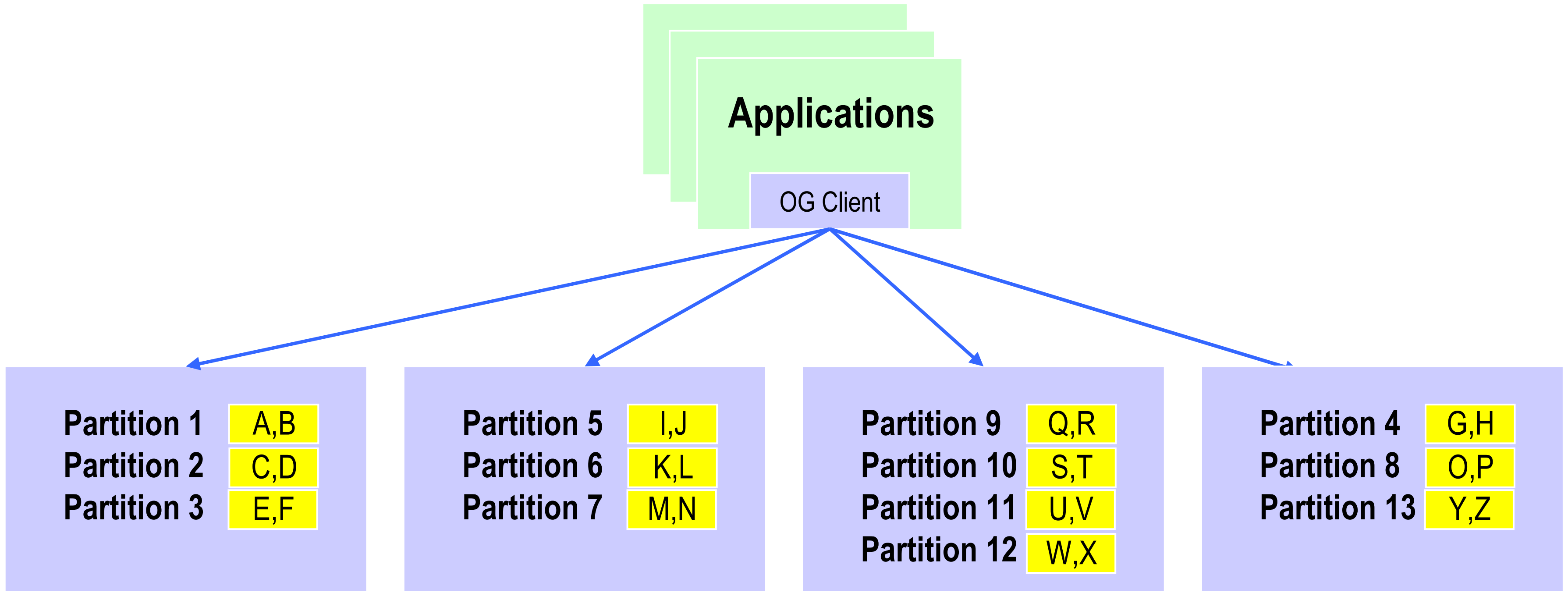
Horizontal Scale-out and Scale-in

- **Scale-Out**
  - As new servers are added then shards are migrated from overloaded servers to the new servers.
  - The new server registers with the catalog service and the catalog service then starts sending migration instructions to the existing servers.
- **Scale-In**
  - As servers fail then shards are also moved around to balance the system.
  - Shards can even be removed to ensure system stability if the grid drops below a certain size.

Partitioning: 13 partitions, 3 servers



Partitioning: 13 partitions, 4 servers



XML Configuration files

- **Located on classpath or in a well known folder in an EJB or WEB module.**
- **Objectgrid.xml**
  - **Lists each Map**
  - **Specifies path to entity.xml**
  - **Specifies plugins for maps and grid.**
- **Entity.xml**
  - **Lists each entity**
  - **Schema fetched via reflection from annotations on classes or specified in xml file.**
- **Deployment.xml describes:**
  - **Number of partitions**
  - **Number and type of replicas**
  - **Placement rules**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://ibm.com/ws/objectgrid/config
../objectGrid.xsd"
      xmlns="http://ibm.com/ws/objectgrid/config">
         <objectGrids>
            <objectGrid name="MyObjectGrid">
               <backingMap name="MyBackingMap01" />
            </objectGrid>
         </objectGrids>
</objectGridConfig>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPo
licy ../deploymentPolicy.xsd"
   xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

   <objectgridDeployment objectgridName="MyObjectGrid">
      <mapSet name="MyMapSet" numberOfPartitions="20"
         minSyncReplicas="1" maxSyncReplicas="1"
         maxAsyncReplicas="1" numInitialContainers="3">
         <map ref="MyBackingMap01" />
      </mapSet>
   </objectgridDeployment>

</deploymentPolicy>
```

WebSphere integration

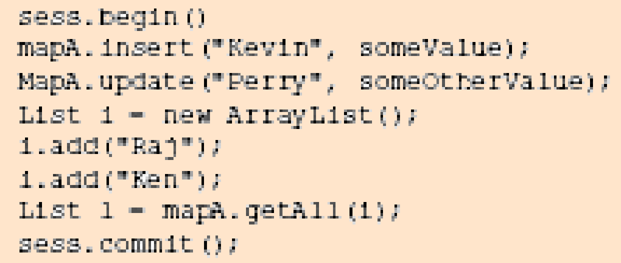
- **Configuration files can be hosted in a module and detected automatically**
- **Catalog service automatically started within a cell**
- **Cells can share a catalog service thus allowing a single grid to span N cells and scale as well as span data centers**
- **HTTP Session support**
  - A servlet filter allows ObjectGrid to act as a HTTP session manager for a servlet container.
  - Full zone support is available for advanced replication topologies.

ObjectGrid Application Programming Model

Developers can develop ObjectGrid using <u>these</u> styles:

- **The traditional JCache style Map API, or**
- **A more streamlined POJO centric approach, the EntityManager API, which provides a dramatically simpler programming model for in memory data management than previously available**
- **Parallel stored procedure like model to grid level performance**

Map

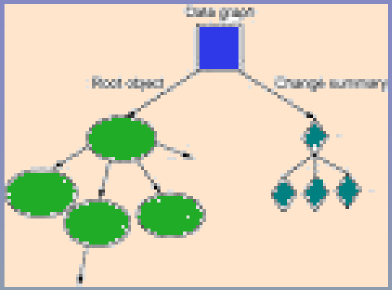


```
sess.begin()
mapA.insert("Kevin", someValue);
MapA.update("Perry", someOtherValue);
List l = new ArrayList();
l.add("Raj");
l.add("Ken");
List l = mapA.getAll(l);
sess.commit();
```

Based on the Java Map interface, extended to allow operations to be grouped into transactional blocks

Allows a set of keywords to be associated with a key

EntityManager



Allows graphs of objects to be <u>annotated with meta-data</u> and be both read from and written to the grid

Each entity/object in the graph corresponds to a Map; ObjectGrid <u>automatically maintains relationships and detects changes</u> to those objects

The programmer simply says to persist the graph to the grid, or retrieve the graph from the grid

ObjectMap API Sample

```
Session session = grid.getSession();

session.begin();
ObjectMap personMap = session.getMap("Person");

Person billy = (Person)
  personMap.get("Billy Newport");
billy.city = "Wexford";

Person child = new Person("Baby Newport");
child.fatherKey = billy.getKey();
billy.childrenKeys.add(child.getKey());

personMap.put(billy.getKey(), billy);
personMap.insert(child.getKey(), child);

session.commit();
```

```
public class Person
  implements Serializable
{
  // References are keys, not object refs
  String key;
  String fatherKey;
  String motherKey;
  ArrayList<String> childrenKeys;
…
}
```

EntityManager API Sample

```
EntityManager em =
 grid.getSession().getEntityManager();

em.getTransaction().begin();
Person billy = (Person)
 em.find(Person.class, "Billy Newport");
billy.city = "Wexford";

Person child = new Person("Baby Newport");
child.father = billy;
billy.children.add(child);

em.getTransaction().commit();
```

```
@Entity
public class Person
{
 @Id String key;

 @OneToOne(cascade=CascadeType.PERSIST)
 Person father;

 @OneToOne(cascade=CascadeType.PERSIST)
 Person mother;

 @OneToMany(cascade=CascadeType.PERSIST)
 Collection<Person> children;
…
}
```

Query Support

- Built-in query engine for querying POJOs or Entities.
- Familiar, rich query language syntax for executing SELECT queries.
- Objects/Entity attributes can be indexed for performance.
- Supports:
  - Named and Positional Parameters
  - Pagination
  - Joins
  - Nested queries
  - Path expressions
  - Index utilization

Stream Query

- **Stream queries allow querying over temporal results**
  - For example: Fetch the top 10 performing items in the catalog over the last 60 minutes.
  - Query the 50 stocks with the highest transaction volumes over the last 5 minutes
  - Query the average of 30 longest transaction time over the last 5 minutes
  - Query the 10 longest average transaction time for each transaction type over the last hour

- **The result is continuously updated in real time.**

- **The result is stored as another Map in the ObjectGrid and can be used exactly as any other Map including replicate it to clients.**

Applications for ObjectGrid

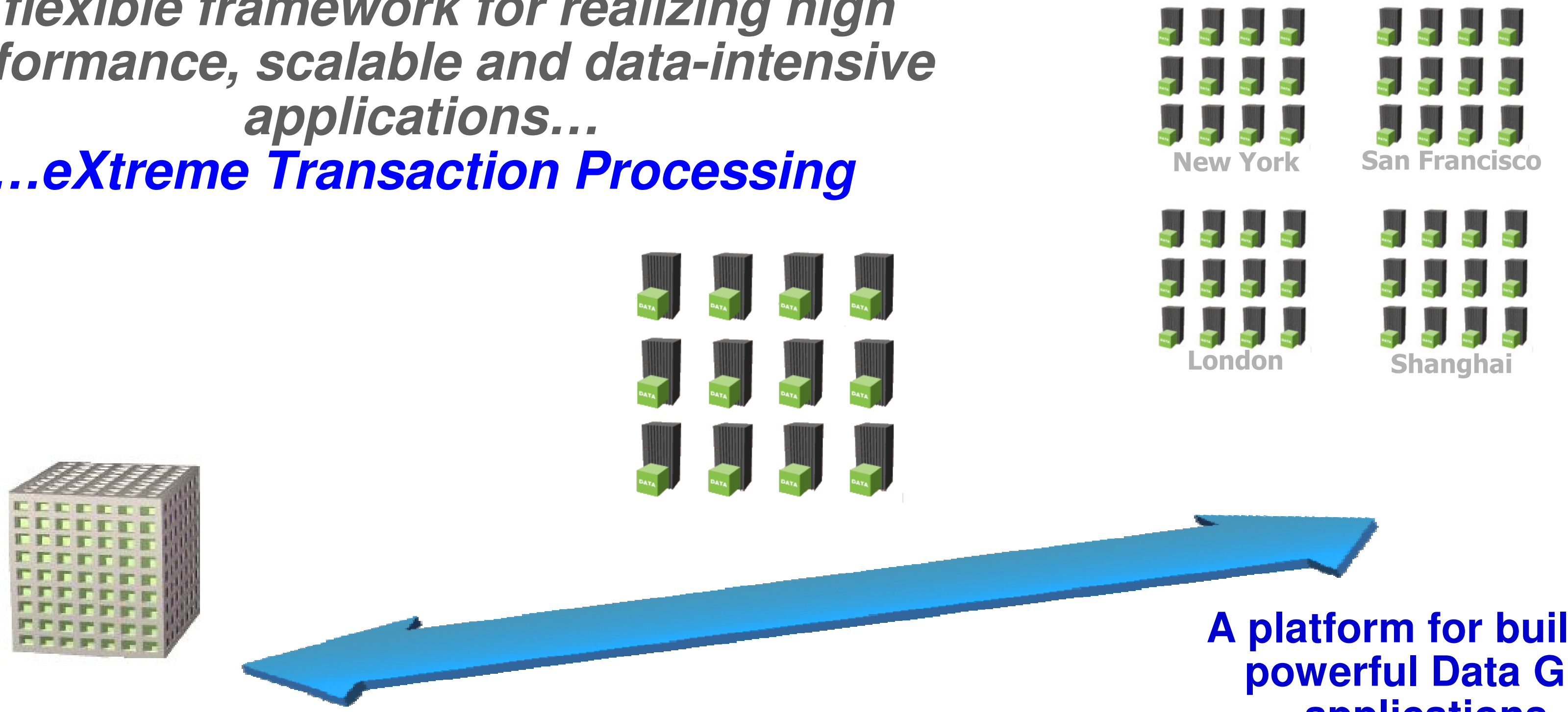
Slides removed here – sorry!

And now for a demo...

ObjectGrid

A flexible framework for realizing high performance, scalable and data-intensive applications…

…eXtreme Transaction Processing



A platform for building powerful Data Grid applications.

A form of in memory database

A very powerful and scalable cache

More info?

- **Download a fully functional ObjectGrid demo copy:**
  - http://www.ibm.com/developerworks/downloads/ws/wsdg/learn.html?S_TACT=105AGX10&S_CMP=JRNL
  - The only limitation is you are forced to restart your servers from time to time
- **My "*Getting started with ObjectGrid*" article:**
  - http://www.ibm.com/developerworks/websphere/techjournal/0711_chambers/0711_chambers.html
- **Highly scalable grid-style computing and data processing with the ObjectGrid component of WebSphere Extended Deployment**
  - http://www.ibm.com/developerworks/websphere/techjournal/0712_marshall/0712_marshall.html
- **Another tutorial "*Building Grid Ready Applications with ObjectGrid*":**
  - http://www.ibm.com/developerworks/edu/wes-dw-wes-objectgrid.html
- **The ObjectGrid documentation wiki:**
  - http://www.ibm.com/developerworks/wikis/display/objectgrid/Getting+started
- ***XTP Platforms: Ready to Make their Mark* - Derrick Harris, Grid Today, Feb 2008**
  - http://www.gridtoday.com/grid/2135376.html
- ***Is XTP just about memory based replication? No, it isn't and here's why* - Billy Newport, Feb 2008**
  - http://www.devwebsphere.com/devwebsphere/2008/02/is-xtp-just-abo.html