

---

# Creating Grid-Based Data Infrastructures for the Enterprise

**Cameron Purdy**  
**President**  
**Tangosol, Inc.**  
**[cpurdy@tangosol.com](mailto:cpurdy@tangosol.com)**

## Speaker's Qualifications

- **Cameron Purdy is President of Tangosol**
- **JSR 107 Lead, JSR 236/237 Expert**
- **Involved with data grid architecture and implementation in several of the world's largest banks and equities firms**



## Overall Presentation Goal

- **Learn the meaning of “data grid”, understand how they are being created in Java, and learn what real world problems they help to solve**
- ***Understanding the reality (if any) behind the marketing buzz-words***

# Define Grid

## ● Background

- **Sometimes people are talking about clusters**
- **Sometimes people are talking about lots of loosely connected machines (e.g. SETA)**
- **The ideas come from the “power grid” – an infrastructure that localizes failure and attempts to eliminate SPOFs**

## Define Grid

- **There are two distinct meanings**
  - **Using an array of servers to create a mainframe-class processing service, e.g. many different and potentially unrelated jobs can be submitted and monitored, and the results come back when they are done**
  - **Using multiple servers to run a single application to improve its throughput and availability**

## Define Grid

- **The two meanings do not contradict in reality**
  - **The first meaning is actually an example of the second meaning (the provisioning and job management service itself is the application)**
  - **Applications built to the second meaning are quite often being deployed to grids built with the first meaning**

## Define Data Grid

- **A data grid provides scalable, reliable data management**
  - **In a grid, data management refers to the ability to access and manipulate read/write information across any number of servers**

# Define Computational Data Grid

- **A computational data grid combines data management with data processing**
  - **Because the amount of processing power is immense in comparison to the amount of network bandwidth, data processing should be localized as much as possible**

# Concepts

- **There are one two things you can move in a distributed environment: State and Behavior.**
  - **The distribution of state is often referred to as “replication”, “distributed caching”, etc.**
  - **The distribution of behavior has traditionally been referred to as RPC, RMI, etc.**
  - **Computational Data grids combine these two concepts**

# Concepts

- **To process data, you can either move the data to where the processing is, or ... move the processing to where the data is**
  - **Moving the parameters of execution is usually much lighter than moving the required data**
  - **Often eliminates the overhead of distributed transaction management**
  - **Supports easy parallelization of work**



# Concepts

- **You need to be able to move both**
  - **Distribution of state allows lots of servers to refer to the same data for their processing**
  - **Distribution of behavior allows lots of servers to process in parallel**
  - **Distribution of behavior also allows processing to occur on the server within a grid that has the best *locality of data***

# Concepts

## ● **Locality of data**

- **Most applications spend most of their time waiting for data**
- **This is still true in a distributed environment, even if all the data is in-memory in the grid**
- **If the data is partitioned into non-overlapping regions, the behavior can be moved to the server that “owns” the data to process**

# Examples

- **There are a limited number of applications that can benefit from a large-scale computational data grid approach**
  - **In our experience, computational data grids are currently most applicable in finance and bio-tech**
  - **Using a computational data grid, a bank was able to reduce an overnight process to less than one minute**
  - **Risk calculations, matching systems**

## How To

- **Requires a well-defined Domain Model**
  - **The Data Model needs to be partitionable, i.e. data objects should have unique identity**
  - **The Behavioral Model should be related to the Data Model, i.e. each behavior should have a data object as its *target* to process**
- **Requires pre-loading the data to process**
  - **Data Grids often keep it all in memory**

## How To

- **Requires partitioning**
  - **An algorithm for segmenting a large data set over multiple servers**
- **Requires mobile behavior**
  - **e.g. Command Pattern**
  - **We refer to the mobile behavior as “agents”**
- **Failover (HA) requires redundancy of data**
  - **In memory, or reload from the data source**

# Agents

- **Agents are targeted**
  - **To a specified object, a set of objects, the objects that match a query, or all objects in the data set**
- **Agents can be stateless**
  - **Operations that can get all of their required data from their target objects**
- **Agents can be stateful**
  - **Conceptually the same as method parameters**

## Summary

- **Grid is a buzz-word, and to keep people confused it has more than one meaning**
- **The term “data grid” is a much less ambiguous way to describe a particular type of data-centric grid application**
- **While only a small percentage of applications requires a large-scale computational data grid, the principle of locality is universal**

# Q&A