

### What are Consistency Groups ?

The concept of grouping all system, middleware, and application volumes that are required to be managed as a consistent entity has become known as a *Consistency Group*.

Dependent write I/O sequences are illustrated in several examples in which the start of one write is time dependent on the completion of a previous write.

- Database and log
- Catalogs, volume table of contents
- Index and data components

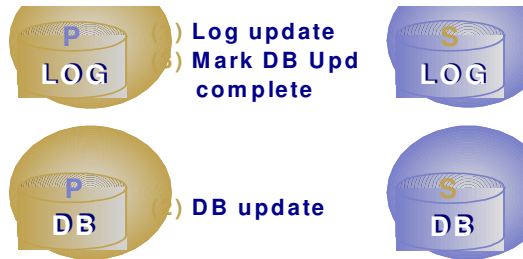
Time sequences can be exposed whenever a copy of the data is being made at the physical record/block level.

Consistency Groups enable customers to group 'related volumes' together representing a "business transaction" that may involve multiple transactions across a single or multiple databases / file systems located on a single or multiple heterogeneous server platform.

### Why must I have CGs ?

With Data Replication Consistency Groups, cross-volume data integrity / data consistency dependent write I/O consistency is maintained on the physical level. However, data consistency at this level is very different from data consistency at a database, file system, or application level. For block / record-based data replication techniques, Consistency Groups insure that the alternate physical image of the data across the entire Consistency Group is maintained. Consistency Groups enable databases to perform a database RESTART function, which examines the DB logs and tables, backing out "in flight and In doubt 'transactions,' making the DB "transaction" consistent. Depending on the application, this may or may not be enough. A "business transaction" may involve a single or multiple transactions across a single or multiple applications. So, the recovery process at a "business transaction" level may require multiple levels of application(s) restarts in which each may involve a single or multiple database and/or file system restart(s). Again, the scope of "business transactions" may be across a single or multiple servers and operating system platforms.

### Dependent write I/O sequences



It is imperative that these "dependent writes" be written to disk in the same sequence that they were issued by the application, as shown in points (1) through (3) above. Furthermore, continuing with the example above, there is no guarantee that the database log and the database itself reside on the same device or even in the same storage subsystem. Failure to execute the write sequence correctly may result in I/O operations (1) and (3) being executed, followed immediately by a system failure. When it came time to recover the database, the database log would indicate that the transaction had completed successfully when, in fact, that is not the case. The transaction would be lost, and the integrity of the database would be in question. Maintaining "dependent write I/O consistency", also commonly called "Power Fail Consistency" is a critical component in all data replication products..

©International Business Machines Corporation, 2007.  
 IBM, the IBM logo, FlashCopy, GDPS, Global Dispersed Parallel Sysplex, HyperSwap, z/VM, z/OS, z.VSE, System z, are trademarks of International Business Machines Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

April 2007



# Consistency Groups in a Nutshell

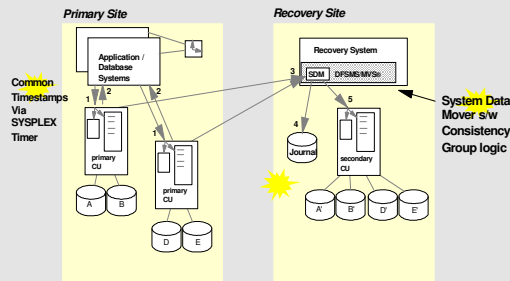
This trifold covers the following topics:

- **What are Consistency Groups (CGs) ?**
- **Dependent write I/O sequences**
- **Why MUST I have Consistency Groups ?**
  - Important for PiT Copy Solutions
  - Important for D/R (DB Restart instead of DBRecovery)
  - Important for data integrity
  - Scope can be Disk Storage Subsystem BUT generally requires Global Systems Level Monitoring.
- **What are the various ways for data replication technologies to create CGs?**
  - Time based CGs
  - Data freeze CGs
  - Ordered I/O CGs
  - Quiescing / Stopping the application(s)
  - Specific application, database, or file system management

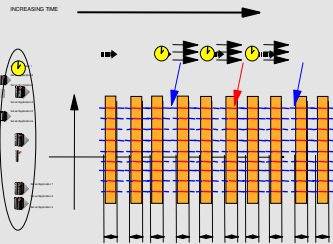
### Time Based CGs

XRC uses the IBM System z™ SYSPLEX Timer to timestamp all write I/O to XRC Active devices. The DFSMSsdm (System Data Mover) reads all updates from all primary storage subsystems and forms CGs based on time.

### Time Based Consistency Groups: XRC System Data Mover



### XRC System Data Mover CGs



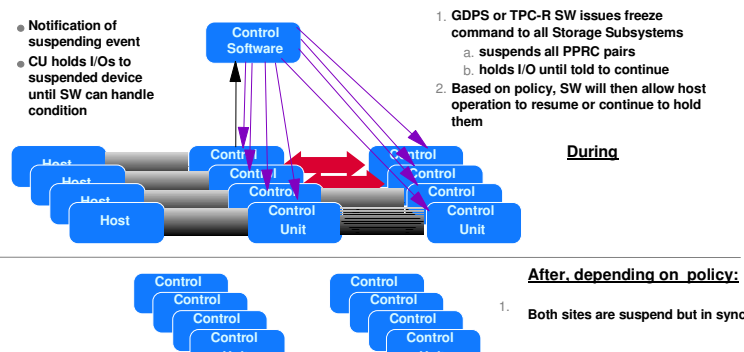
XRC SDM uses common Sysplex Timer timestamp to sort the incoming data, form consistency groups 10's of times per second across large numbers of volumes, disk frames, and z/OS® images.

XRC is able to back out in-flight incomplete write sequences because in event of outage, the XRC SDM does not write out incomplete data, thus what is on the disk is the most recent complete Consistency Group.

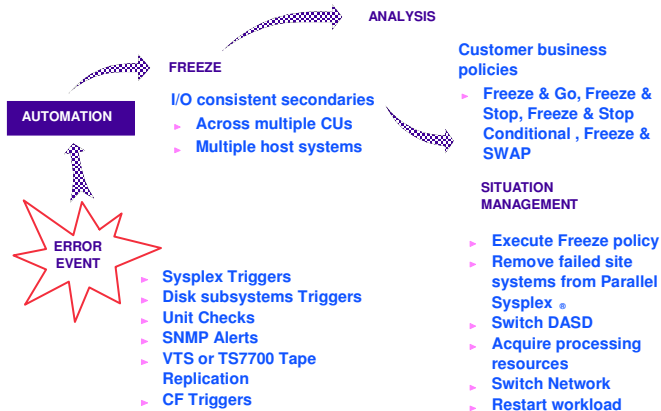
### Data Freeze CGs

Automation code such as Geographically Dispersed Parallel Sysplex (GDPS®) and TPC-R exploit a storage subsystem feature known as *CGROUP Freeze/Run*. Software automation invokes these functions on hardware- or software-related errors to cause all synchronous mirrored volumes to be suspended while maintaining all dependent I/O. Numerous 'trigger' events are raised from the storage subsystem hardware as well as various messages / alerts raised by the software. The data freeze is used in functions such as HyperSwap, Consistent FlashCopy®, MM session management (GDPS, TPC-R, etc.) and Global Mirror (toggle process).

### Disaster Recovery - Freeze



### GDPS/PPRC Data Freeze-based Consistency Groups



### Ordered I/O CGs

IBM Storage Systems SAN Volume Controller Global Mirror (as well as many SAN-based data replication products) maintain cross-volume data consistency within the scope of all I/Os that go through that device by ordering the I/Os "first in, first out" (FIFO) for transport to the remote device. Maintaining write I/O order in a single queue for all devices that the data replication product manages ensures that dependent I/O operations are managed correctly.

### Quiesce/Stop Application(s) CGs

A final method of maintaining dependent I/O within or across multiple storage subsystem-attached devices, when those devices are involved in some form of data replication, is to stop all I/O activity. This can be done a number of different ways, including:

- ✓ Database quiesce
- ✓ Stopping the application(s) involved
- ✓ DB Command SET SUSPEND LOG
- ✓ Application quiesce

On Open Systems, some customers like this method because it can flush writes completed in the host file system to the disk subsystem before a disk data replication "Sync" point is taken.

### Application Ordering

Specific application, database, or file system knowledge is used to maintain cross-volume data consistency and integrity with many software-based replication schemes that are available with specific software databases, file systems and applications.