Taking Autonomics to a New Level in IMS Version 9

By Ben Johnson and Jack Yuan

IMS™, or "Information Management System," is IBM's premier transaction and hierarchical database management software. Even after more than 35 years, IMS development continues to incorporate the latest technological advances into the IMS product, and IMS is still the product of choice for critical online operational applications and data where support for high availability, performance, capacity and integrity, and low cost are key factors.

With IMS Version 9, IBM has further developed IMS autonomic capabilities – and the company's commitment to autonomic computing, in general – so that it is more self-managing than ever.

Autonomics is frequently divided into four categories:

- Self-configuring (able to adapt to changes in the system)
- Self-healing (able to recover from mistakes)
- Self-optimizing (able to improve performance)
- Self-protecting (able to anticipate and cure intrusions)

IMS Version 9 introduces features in each of these categories.

Self-healing

IMS Version 9 introduces several features that fall into the self-healing category.

Identifies indoubt data in DB2 after IMS failures

When an IMS system is connected to DB2® or another subsystem through the IMS External Subsystem Attach Facility (ESAF), IMS or an IMS Fast Database Recovery (FDBR) region can identify the IMS units of work (UOW) associated with DB2 data that might be indoubt as the result of a failure. During the FDBR recovery process or the IMS emergency restart, the identity of an indoubt UOW is passed to an exit routine, which allows the user to resolve the indoubt data in DB2 prior to completing the restart of IMS.

Accesses and identifies online logs in the event of errors

IMS Version 9 can access the online log data set (OLDS) and determine which online log it should read during a restart, even when the OLDS has been closed to IMS due to an error. IMS cannot run if it does not have access to an online log in the OLDS.

Monitors the open and closed status of Fast Path DEDB areas

IMS Version 9 monitors the opened and closed status of Fast Path DEDB areas so that during an emergency restart, IMS can reopen only those DEDB areas that were open at the time of the IMS failure. This ensures that DEDB areas that are defined as preopen or preload are not automatically opened if they were closed for maintenance at the time of the IMS failure.

IMS Version 9 performs a similar service in the event of IRLM failures. During IRLM reconnect, IMS can restart only those DEDB areas that were started at the time IRLM failed.

Additionally, IMS Version 9 can optimize emergency restart processing by opening DEDB areas asynchronously to the completion of the restart process and the resumption of application program processing. Application programs can access DEDB areas before IMS finishes area open processing.

Detects and manages internal IMS Connect message errors

The integrated IMS Connect function of IMS Version 9 can detect errors in the user data prefix of messages that are returned from IMS. Instead of abending, IMS Connect rejects the incorrect message, and then IMS stores the message in a dead letter queue defined by IMS Connect.

Self-configuring

IMS Version 9 introduces several features that fall into the self-configuring category.

Requires no operator intervention for status changes of an FDBR-tracked system

An IMS Version 9 Fast Database Recovery (FDBR) system, with the help of the MVS™ Cross-System Coupling Facility (XCF) Group exit routine, can now respond appropriately when the IMS system the FDBR system is tracking changes from inactive to active or from active to inactive. In previous versions of IMS, FDBR required operator intervention when the status of the IMS system it was tracking, changed.

Simplifies installation

IMS Version 9 dynamically installs its Resource Manager cleanup routine. Users do not need to install the DFSMRCL0 module as part of the IMS installation. Registration of the IMS Resource Manager cleanup routine with the operating system is done automatically during IMS startup.

Self-optimizing

IMS Version 9 introduces several features that fall into the self-optimizing category.

Reorganizes HALDB databases online

The integrated HALDB Online Reorganization (OLR) function of IMS Version 9 reorganizes HALDB PHDAM and PHIDAM partitions without requiring the user to stop access to the partition and then unload and reload the partition, greatly reducing the amount of time required for planned maintenance outages. HALDB Online Reorganization provides continuous availability for the largest IMS databases.

Uses less common storage area

IMS Version 9 uses less common storage area (CSA) by moving two DBRC load modules (DSPCINT1 and DSPURX00) into private and common storage above the 16 MB line.

Communicates directly with other IMS systems in shared queues environments

IMS Version 9 front-end systems communicate directly with back-end IMS systems in an APPC/OTMA shared queues environment, allowing the back-end IMS systems to respond immediately when a front-end IMS system abends. The back-end shared queues systems can then quickly clear the dependent regions that are in a RRS-PC wait status.

Self-protecting

IMS Version 9 can now protect itself from message flood conditions.

Protects from a flood of input messages from OTMA clients

IMS Version 9 can monitor the flow of incoming messages from an OTMA client, such as IMS Connect or WebSphere® MQSeries®, to prevent a flood of messages from exhausting system resources. After the user defines the maximum number of active OTMA input messages allowable from the OTMA client, OTMA monitors the input messages. As the number of input messages approaches the maximum number, IMS issues warning messages. If the maximum number is reached, IMS rejects any subsequent input message from the OTMA client until the flood condition is relieved.

Autonomic IMS: now and in the future

IMS Version 9 exhibits the commitment of IMS development to autonomic computing. And the future releases of IMS will continue to rigorously adhere to the principles of autonomic computing, endeavoring to make the IMS of the future truly autonomic: self-diagnosing, self-healing, self-optimizing, and self-protecting.

About the authors

Ben Johnson is an information developer at IBM's Silicon Valley Labs in San Jose, California. Ben has worked in IMS since 2003.

Jack Yuan is a senior software engineer at IBM's Silicon Valley Labs in San Jose, California. Jack specializes in providing open access to IMS TM and DB.