DTCC2013

利用XEvent进行高级Troubleshooting

王成辉
高级咨询顾问 | 上北智信

微博:@windbi
chwang@sharewinfo.com

本次议程目的

- 目的:
  - 介绍SQLServer扩展事件（Extended Events，简称XEvent）
  - 理解扩展事件架构
  - 使用扩展事件对SQLServer进行Troubleshooting
- 扩展事件显著地减少诊断和解决问题的时间
- 扩展事件可深度地揭示SQLServer的内幕

议程

- 什么是扩展事件
- 扩展事件的特点
- 扩展事件的架构
- 扩展事件工作原理
- 扩展事件对象
- 事件的生命周期
- 扩展事件的元数据
- 扩展事件的创建和跟踪数据的读取
- Demo

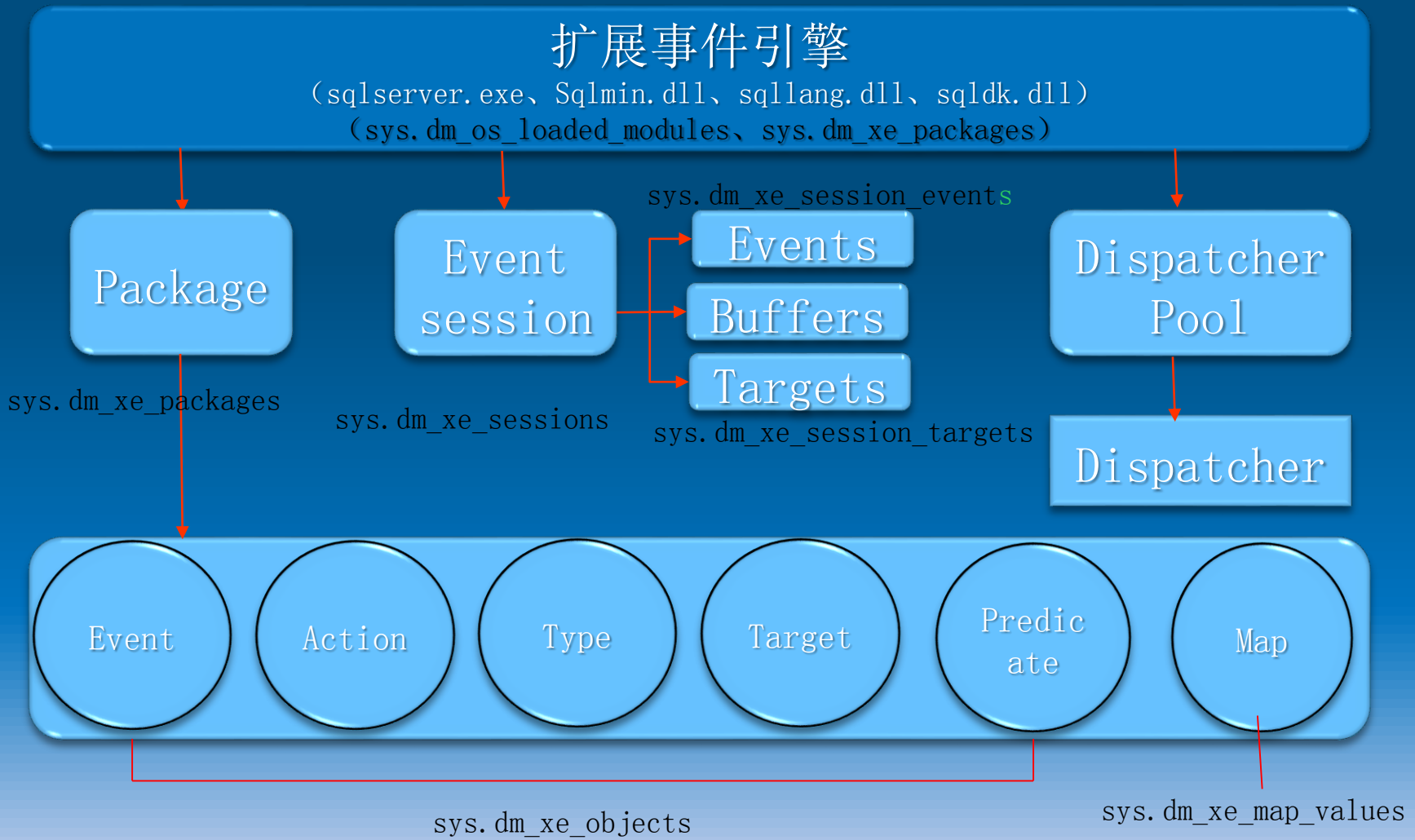
什么是扩展事件

- 事件
  - 显示某事发生的软件信息，比如单击（wikipedia）
- 扩展事件
  - 在数据库引擎里边，当某些事件（比如死锁）发生时产生一些信息，用来捕获这些信息的一个新的机制，就是扩展事件。

扩展事件的特点

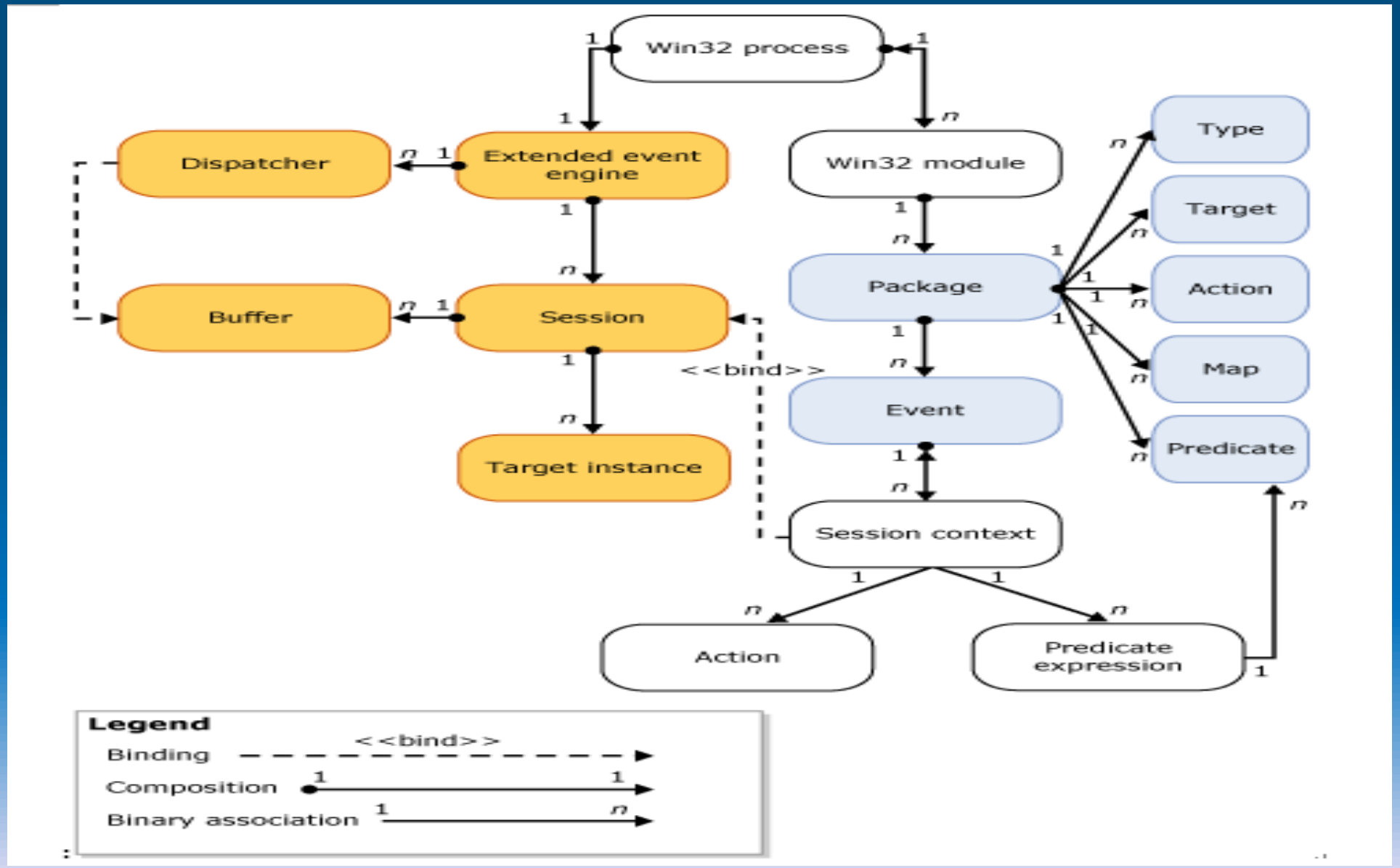
- 轻量级、高性能、更少系统资源
- 高扩展性高可配置性
- 更准确
- 更多的计数器
  - 支持trace所有的计数器
- 更底层更细粒度

扩展事件的架构



上北智信
Sharewinfo

Event Session 工作原理



上北智信 Sharewinfo

扩展事件的对象：Package

- 包是一个容器
  - 包含Event、Target、Action、Predicate、Type、Map
  - 各包之间的对象可以混合使用
- 每个包都有GUID和名称
- 包在模块载入的时候注册
- Sys.dm_xe_packages

	name	guid	description
1	package0	60AA9FBF-673B-4553-B7ED-71DCA7F5E972	默认包。包含所有标准的类型、映射、比较运算符、操作和目标
2	sqlos	BD97CC63-3F38-4922-AA93-607BD12E78B2	SQL 操作系统的扩展事件
3	XeDkPkg	52FC232C-03D5-4E1F-A6BF-BBC66FE20E6A	Extended events for SQLDK binary
4	sqlserver	655FD93F-3364-40D5-B2BA-330F7FFB6491	Microsoft SQL Server 的扩展事件
5	SecAudit	F235752A-D5C0-4C9A-A735-9C3B6F6E43B1	安全审核事件
6	ucs	C0AB75C5-B1EA-445B-B7DF-F897686F94E7	统一通信堆栈的扩展事件
7	sqlclr	1E99FE90-A4FE-45E6-9DFD-A45041F02314	SQL CLR 的扩展事件
8	filestream	B086C2F3-2738-4389-B119-D80B5362B5CA	Extended events for SQL Server FILESTREAM and FileTable
9	sqlserver	03FDA7D0-91BA-45F8-9875-8B6DD0B8E9F2	Microsoft SQL Server 的扩展事件

上北智信
Sharewinfo

扩展事件的对象：Event

- 事件对应数据库引擎里某件事情发生的代码执行路径里的特定的点
- 所有事件都是同步执行的
- 每个事件都都包含一组字段称作payload，它是事件默认收集的数据。
- 事件可以按照channel和keyword来进行分类，与ETW分类相同，所以可以与ETW进行集成

- sys.dm_xe_objects、sys.dm_xe_object_columns

扩展事件的对象：Action

- Action是事件触发时候可以执行的一些特定的命令或任务
  - 收集额外的信息
  - 捕获堆dump文件和检查数据
  - 汇总事务数据
  - 存储变量的本地上下文状态信息
  - 停止服务器执行，创建调试断点
- 与事件是同步执行的，所以可能对性能有影响
- 可以和任何事件关联
- sys.dm_xe_objects

扩展事件的对象：Target

- Target是事件的数据存储的地方
  - 内存或文件
  - 同步或异步
- 可以有多个Target
- sys.dm_xe_objects

PackageName	TargetName	TargetDescription
package0	etw_classic_sync_target	针对 Windows (ETW)同步目标的事件跟踪
package0	event_counter	使用 event_counter 目标计算各事件在事件会话中出现的次数。
package0	event_file	使用 event_file 目标将事件数据保存到 XEL 文件中，可以将该文件存档，供将来分析和查看。可以合并多个...
package0	event_stream	异步实时流目标。
package0	histogram	使用直方图目标基于特定事件数据字段或与该事件关联的操作聚合事件数据。该直方图允许您分析事件数据在...
package0	pair_matching	配对目标
package0	ring_buffer	异步环形缓冲区目标。
sqlserver	asynchronous_router	将事件路由到异步侦听器。
sqlserver	router	将事件路由到侦听器。

上北智信
Sharewinfo

扩展事件的对象：Predicate

- Predicate动态过滤事件
- 与事件同步发生
- 每个事件可以使用不同的Predicate
- 它是一个布尔表达式,支持短路
- Predicate可以在本地上下文的payload进行过滤，也可以在一些全局状态信息上过滤
- 要求数据类型一致
- 有一些predicate可以存储状态信息以便允许事件每隔多少次执行一次或者只执行多少次,可用于抽样
- sys.dm_xe_objects

上北智信
Sharewinfo

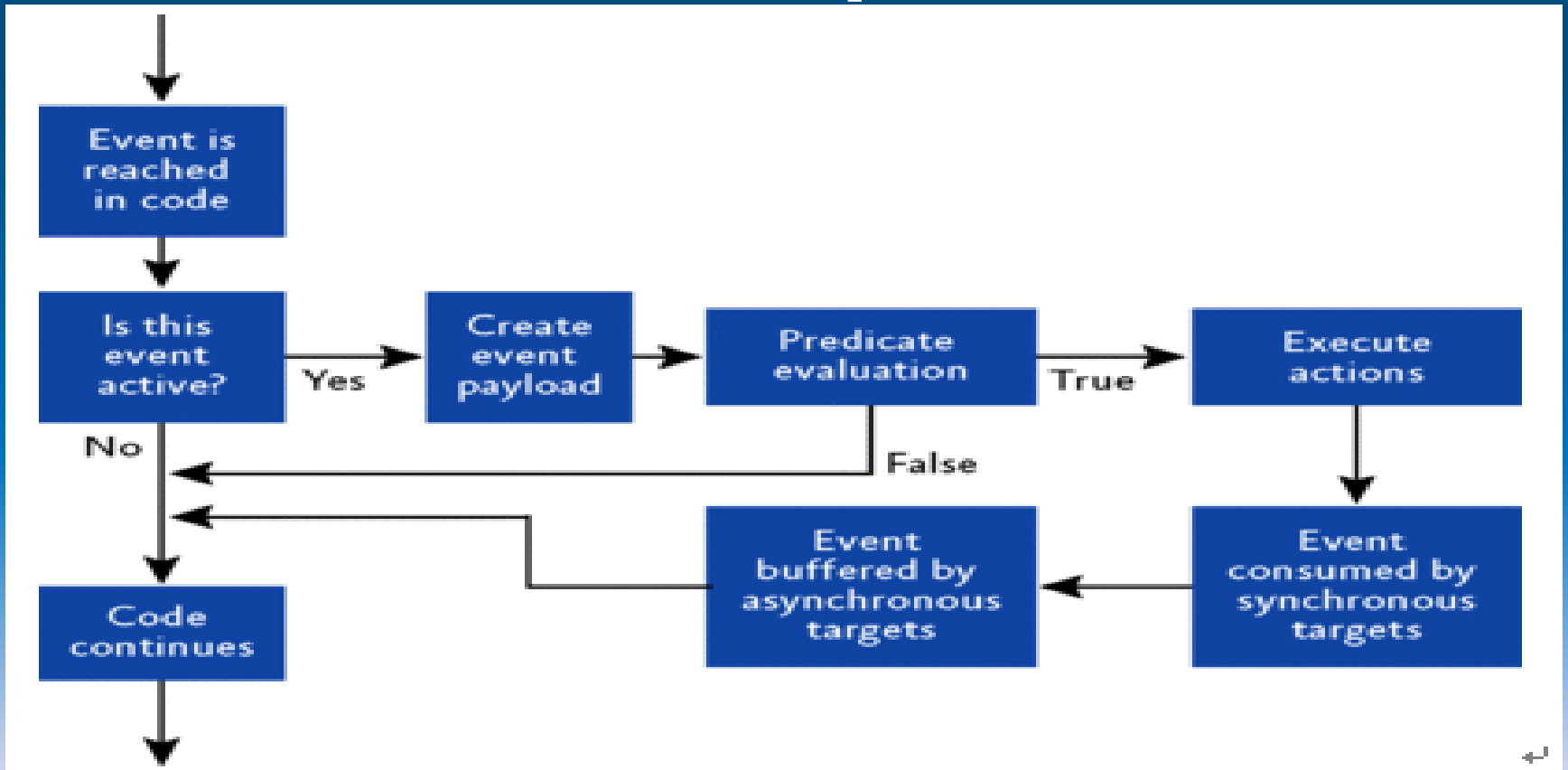
扩展事件的对象：Type&Map

- Type定义了事件的信息元素即payload的数据类型
- Map是一个key/value形式的字典表，它把一些数字映射成用户能看得懂的文本

- 二者都不用于事件会话定义的一部分

事件会话

- 事件会话是一组Event、Action、Predicate、Target的任意组合
- DDL语句：Create（alter、drop） Event Session



上北智信
Sharewinfo

事件会话的选项

- 事件会话有7个选项可以设置
  - ① EVENT_RETENTION_MODE
    - ALLOW_SINGLE_EVENT_LOSS、ALLOW_MULTIPLE_EVENT_LOSS、NO_EVENT_LOSS
  - ② MAX_DISPATCH_LATENCY
  - ③ MAX_EVENT_SIZE
  - ④ MAX_MEMORY
  - ⑤ MEMORY_PARTITION_MODE
    - NONE、PER_MODE、PER_CPU
  - ⑥ STARTUP_STATE
  - ⑦ TRACK_CAUSALITY

事件会话相关的DMV

① Sys.server_event_sessions
② Sys.server_event_session_events
③ Sys.server_event_session_actions
④ Sys.server_event_session_targets
⑤ Sys.server_event_session_fields
⑥ Sys.dm_xe_sessions
⑦ Sys.dm_xe_session_targets
⑧ Sys.dm_xe_session_events
⑨ Sys.dm_xe_session_event_actions
⑩ Sys.dm_xe_session_object_columns

上北智信
Sharewinfo

创建扩展事件会话

- 跟踪页拆分的会话

```
CREATE EVENT SESSION xe_event_page_split ON SERVER
ADD EVENT  sqlserver.page_split
    (ACTION
(sqlserver.database_id,sqlserver.sql_text,sqlserver.nt_userna
me,sqlserver.transaction_id)
    WHERE sqlserver.database_id > 4)
ADD TARGET package0.asynchronous_file_target
    (SET FILENAME=N'c:\xe\xe_event_page_split.xel',
    metadatafile=N'c:\xe\xe_event_page_split.xem');
```


上北智信
Sharewinfo

跟踪孤立事务的会话


```
CREATE EVENT SESSION OrphanedTransactionHunter ON SERVER
ADD EVENT sqlserver.database_transaction_begin (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.database_transaction_end   (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.sql_statement_starting     (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
sqlserver.sql_text)    ),
ADD EVENT sqlserver.sql_statement_completed    (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.sp_statement_starting      (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
sqlserver.sql_text)    ),
ADD EVENT sqlserver.sp_statement_completed     (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.rpc_starting               (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
sqlserver.sql_text)    ),
ADD EVENT sqlserver.rpc_completed              (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.module_start               (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
sqlserver.sql_text)    ),
ADD EVENT sqlserver.module_end                 (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD EVENT sqlserver.error_reported             (    ACTION(sqlserver.session_id, sqlserver.database_id, sqlserver.tsql_
ADD TARGET package0.ring_buffer,
ADD TARGET package0.pair_matching
( SET begin_event = 'sqlserver.database_transaction_begin', begin_matching_actions = 'sqlserver.session_id', end_event
'sqlserver.database_transaction_end',
      end_matching_actions = 'sqlserver.session_id', respond_to_memory_pressure = 1
)
WITH (MAX_DISPATCH_LATENCY=5 SECONDS, TRACK_CAUSALITY=ON)
```


上北智信
Sharewinfo

跟踪数据库备份的会话


```
CREATE EVENT SESSION BackupMonitoring ON SERVER
ADD EVENT sqlserver.sql_statement_starting  (   ACTION (sqlserver.database_id, sqlserver.sql_text)     WHERE
(sqlserver.session_id = 97)),
ADD EVENT sqlserver.sql_statement_completed (   ACTION (sqlserver.database_id, sqlserver.sql_text)    WHERE
(sqlserver.session_id = 97)),
ADD EVENT sqlserver.databases_backup_restore_throughput (   WHERE (sqlserver.session_id = 97)),
ADD EVENT sqlos.wait_info (   ACTION (sqlserver.database_id)     WHERE (sqlserver.session_id = 97  AND
duration > 0)),
ADD EVENT sqlos.wait_info_external (   ACTION (sqlserver.database_id)     WHERE (sqlserver.session_id = 97
AND duration > 0)),
ADD EVENT sqlserver.trace_print (   WHERE (sqlserver.session_id = 97)),
ADD EVENT sqlserver.file_read (   WHERE (sqlserver.session_id = 97)),
ADD EVENT sqlserver.file_read_completed (   WHERE (sqlserver.session_id = 97)),
ADD EVENT sqlserver.physical_page_read (   WHERE (sqlserver.session_id = 97)),
ADD EVENT sqlserver.databases_log_cache_read (   WHERE (database_id = 41)),
ADD EVENT sqlserver.databases_log_cache_hit (   WHERE (database_id = 41)),
ADD EVENT sqlserver.databases_log_flush (   WHERE (database_id = 41)),
ADD EVENT sqlserver.checkpoint_begin (   WHERE (database_id = 41)),
ADD EVENT sqlserver.checkpoint_end (   WHERE (database_id = 41))
ADD TARGET package0.asynchronous_file_target(     SET filename='C:\SQLBlog\BackupMonitoring1.xel',
        metadatafile = 'C:\SQLBlog\BackupMonitoring1.xem')
```


上北智信
Sharewinfo

跟踪死锁的会话

Demo

谢谢

王成辉
高级咨询顾问 | 上北智信

微博:@windbi
chwang@sharewinfo.com