SYBASE®
An SAP® Company

性能和调优系列：监控表

Adaptive Server® Enterprise

15.7

文档 ID：DC01077-01-1570-01

最后修订日期：2011 年 9 月

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目录

第 1 章 监控表简介 1

Adaptive Server 中的监控表 1

监控信息来自哪里？ 2

使用 Transact-SQL 监控性能 2

安装监控表 3

低于 15.0.2 的版本， Cluster Edition 除外 3

远程访问和编辑监控表 4

配置监控表以收集数据 4

为管道错误消息分配内存 6

监控表的配置参数 6

错误 12036 10

mon_role 和其它访问控制 10

计数器数据类型归零 11

有态历史监控表 12

瞬时监控数据 14

在集群环境中使用监控表 15

配置系统视图 15

为监控实例添加的 InstanceID 16

语句高速缓存的监控表 17

配置 Adaptive Server 以监控语句高速缓存 17

从语句高速缓存中删除语句 18

从 SQL 文本获取散列键 18

显示高速缓存语句的文本和参数信息 18

查询监控表的示例 19

第 2 章 等待事件 21

事件 19：xact coord：在空闲循环期间暂停 23

操作 23

事件 29：等待常规缓冲区读取完成 23

操作 23

事件 30：等待写入 MASS 而 MASS 正在更改 24

操作 24

事件 31：在写入之前等待缓冲区写入完成 24

操作 25

事件 32：等待 APF 缓冲区读取完成 25
操作 25
事件 35：等待缓冲区验证完成 25
操作 25
事件 36：在更改之前等待 MASS 完成写入 26
操作 26
事件 37：在更改之前等待 MASS 完成更改 26
操作 26
事件 41：等待获取闩锁 26
操作 27
事件 46：等待缓冲区写入完成以便从 LRU 获取缓冲区 27
操作 28
事件 51：等待 MASS 上的最后一个 i/o 完成 28
操作 28
事件 52：等待 MASS 上被另一个任务启动的 i/o 28
操作 29
事件 53：等待 MASS 完成更改以便开始 i/o 29
操作 29
事件 54：等待最后一个日志页的写入完成 29
操作 30
事件 55：在写入最后一个日志页后等待 i/o 完成 30
操作 30
事件 57：检查点进程空闲循环 30
操作 30
事件 61：hk：暂停一段时间 31
操作 31
事件 70：等待设备信号 31
操作 31
事件 83：等待 DES 状态更改 31
操作 31
事件 84：等待检查点完成 32
操作 32
事件 85：等待刷新程序将完整的 DFLPIECE 排入队列 32
操作 32
事件 91：等待磁盘缓冲区管理器 i/o 完成 32
操作 33
事件 99：等待来自客户端的数据 33
操作 33
事件 104：等到引擎已脱机 33
操作 33
事件 124：在获取页时等待簇读取完成 34
操作 34
事件 142：等待逻辑连接释放 34
操作 34

事件 143：暂停以便与节点管理器同步 34
操作 35
事件 150：等待锁 35
操作 35
事件 157：等待对象返回到池 35
操作 36
事件 169：等待消息 36
操作 36
事件 171：等待 CTLIB 事件完成 36
操作 36
事件 178：等待分配新客户端套接字 37
操作 37
事件 179：在无需网络读取或写入时等待 37
操作 37
事件 197：等待读取在并行 dbcc 中完成 37
操作 38
事件 200：等待并行 dbcc 中的页读取 38
操作 38
事件 201：等待并行 dbcc 中的磁盘读取 38
操作 38
事件 202：等待以并行方式重新读取页 39
操作 39
事件 203：等待并行 dbcc 中的 MASS_READING 位 39
操作 39
事件 205：等待并行 dbcc 中的 TPT 锁 39
操作 40
事件 207：等待并行 dbcc 中将故障消息发送给父项 40
操作 40
事件 209：等待管道缓冲区读取 40
操作 40
事件 210：等待管道管理器中的可用缓冲区 41
操作 41
事件 214：退让之后在运行队列中等待 41
操作 41
事件 215：休眠之后在运行队列中等待 42
操作 42
事件 222：复制代理在刷新期间休眠 42
操作 42
事件 250：等待进站网络数据 42
操作 43
事件 251：等待网络发送完成 43
操作 43
事件 259：等到最后机会阈值被清除 43
操作 44

事件 260：等待 waitfor 命令中的日期或时间 44
操作 44
事件 266：等待工作线程邮箱中的消息 44
操作 44
事件 272：等待 ULC 上的锁 45
操作 45
事件 334：等待 Lava 管道缓冲区写入 45
操作 45
事件 374：等待锁挂起 / 数据挂起被清除 45
操作 46
事件 375：OCM 等待完成 BAST 处理 46
操作 46
事件 389：OCM 等待推送数据标志被清除 46
事件 380：在 OCM_ERR_DIDNTWAIT 时锁定 / 数据挂起重置 47
操作 47
事件 483：等待多播同步消息的确认 47
操作 47

索引 49

第 1 章

监控表简介

本章介绍了如何从 Adaptive Server® 的监控表中查询统计和诊断信息。

主题	页码
Adaptive Server 中的监控表	1
安装监控表	3
远程访问和编辑监控表	4
配置监控表以收集数据	4
mon_role 和其它访问控制	10
计数器数据类型归零	11
有态历史监控表	12
在集群环境中使用监控表	15
语句高速缓存的监控表	17
查询监控表的示例	19

Adaptive Server 中的监控表

Adaptive Server 提供一组系统表，这些表中包含监控信息和诊断信息。这些表中的信息为您提供了 Adaptive Server 状态的统计快照，能让您分析服务器以便分析服务器性能。例如，您可以执行查询以报告有关服务器进程和应用程序的活动、查询性能、数据库表的使用情况、数据高速缓存的效率、数据库设备上的 I/O 活动以及 Adaptive Server 影响系统性能的许多其它方面的信息。

监控表中的数据不存储在磁盘上。当您在一个监控表上执行查询时，数据便会被计算。表定义包含在服务器安装脚本所创建的代理表定义中。当您执行查询时，这些代理表使用一个到 Adaptive Server 的接口来收集监控数据。

必须使用服务器安装脚本创建监控表。请参见第 3 页的 “安装监控表”。

因为 Adaptive Server 版本可能会更改监控表的定义，所以 Sybase® 建议您在升级时，在使用监控表之前先运行适当的安装脚本。

注释 必须具有 mon_role 权限才能查询这些表。请参见第 10 页的“mon_role 和其它访问控制”。

监控信息来自哪里？

Adaptive Server 从以下来源收集有关监控表的信息：

- 全局监控计数器 （例如， monSysWaits，它只有有限数量的行）。
- 特定于资源的监控计数器 （例如， monCachedProcedures，结果行数取决于过程高速缓存中的高速缓存编译对象的快照），它与单个服务器资源 （如引擎、过程或数据高速缓存）相关联。
- 活动进程状态结构 （例如， monProcessWaits 和 monProcessSQLText）。monProcessWaits 或 monProcessSQLText 的结果行数分别取决于活动用户连接数和活动进程状态结构数。
- 循环缓冲区 （例如， monSysStatement 和 monDeadLock）。monSysStatement 或 monDeadLock 的结果行数与配置参数设置以及快速数据管道中包含的数据量有关。此缓冲区由所有历史监控表使用。

对于一些大型生产服务器，物化某些监控表可能需要消耗一些资源和时间。

使用 Transact-SQL 监控性能

以表的形式提供监控信息，使您可以使用 Transact-SQL 来监控 Adaptive Server。例如，若要确定 CPU 时间或逻辑 I/O 消耗量最大的进程，可使用：

```
select SPID, Login = suser_name(ServerUserID), CPUTime,
LogicalReads
from master..monProcessActivity
order by CPUTime desc
```

可以通过用 PhysicalReads 替换 CPUTime，使用同样的查询来查找物理 I/O 使用量最大的进程。

对于每个监控表中的信息，可以对其进行排序、选择、连接或插入到另一个表中，而且其处理方式与常规 Adaptive Server 表中的信息的处理方式非常相似。

监控表是只读的，不允许更新，因为它们是内存中的表，在被查询时生成。此外，不能在监控表上创建触发器。

可以使用 grant 和 revoke select 等访问控制命令来限制对监控表的访问。

监控表定义使用组件集成服务 (CIS) 代理表功能，该功能可让 Adaptive Server 将远程过程定义为本地表。

安装监控表

低于 15.0.2 版的 Adaptive Server 的监控表的安装过程不同于 15.0.2 版和更高版本的过程。本节说明早期版本的安装过程。

Adaptive Server 15.0.2 版和更高版本以及 Cluster Edition 的监控表：

- 当您运行 *installmaster* 脚本时安装
- 使用物化视图
- 要求您创建回送服务器

低于 15.0.2 的版本， Cluster Edition 除外

使用位于 *$SYBASE/ASE-15_0/scripts* 目录 （对于 Windows 为 *%SYBASE%\ASE-15_0\scripts*）中的 *installmontables* 脚本创建监控表。

使用 isql 实用程序运行 *installmontables* 脚本。例如：

```
isql -Usa -Ppassword -Sserver_name -i $SYBASE/ASE-15_0/scripts/installmontables
```

为 15.0.1 ESD #2 和更低版本配置回送代理服务器

Adaptive Server 15.0.1 ESD #2 版和更低版本要求在运行 *installmontables* 脚本之前，名为 “loopback” 的服务器已包括在 sysservers 中。若要创建此服务器，可输入：

```
declare @servernetname varchar(30)
select @servernetname=srvnetname
from master..sysservers
where srvname=@@servername
exec sp_addserver loopback, NULL, @servernetname
```

@@servername 不能为 NULL。如果它当前是 NULL，则使用 sp_addserver 定义一个 “本地” 服务器名。重新启动服务器，以使对 *@@servername* 所做的更改生效。

远程访问和编辑监控表

在 15.0.2 版和更高版本中， Sybase 提供 *installmontables* 作为示例脚本，用以说明如何远程访问监控表 （无需在直接监控 Adaptive Server 15.0.2 版和更高版本的服务器上运行 *installmontables* 脚本，即可创建监控表）。请运行 *installmontables* 以查看有关编辑的说明。例如：

```
isql -Usa -Psa_password -Sserver name -i $SYBASE/$SYBASE_ASE/scripts/
installmontables
---x---x-------------------------------------------------------------x---x---
It is no longer necessary to run this script to install the Monitoring
Tables.Monitoring Tables are now installed by the installmaster script.
This installmontables script is provided as a sample that can be copied and
modified to support remote access of Monitoring Tables. To do so you need to:
1) Replace all instances of @SERVER@ with the name of the remote ASE from which
monitoring data is to be obtained. Note that each remote ASE to be monitored
must be added to the local ASE's sysservers table using sp_addserver.
2) Create a database with the same name as the remote ASE. This database need
only be of the minimum size as these tables do not store any data.
3) Remove this header (i.e. these first 21 lines).
4) Run the script against the local ASE using the isql utility as follows:
    isql -Usa -P<password> -S<server name> -i<script name>
5) Retrieve remote monitoring data. E.g. to obtain monEngine information for an
ASE named REMASE you would execute the following SQL:
            use REMASE
            go
            select * from monEngine
            go
```

配置监控表以收集数据

缺省情况下， Adaptive Server 不收集监控表所需的监控信息。使用 sp_configure 将配置 Adaptive Server 为开始收集监控信息。有许多配置选项 （下面列出）用于控制许多区域的监控数据收集。

许多监控表要求您在 Adaptive Server 收集其数据之前启用一个或多个配置选项。不同表需要不同选项。表 1-1 介绍了每个监控表需要的配置选项。

若要配置 Adaptive Server 以收集常规监控信息，请执行以下操作：

1 缺省情况下，当您首次配置 Adaptive Server 时， enable cis 配置参数处于启用状态 （设置为值 1）。请检验此参数是否被启用。

2　enable monitoring 配置参数决定其它监控选项是否启用；请将 enable monitoring 设置为 1。

```
sp_configure "enable monitoring", 1
```

当您输入以下命令时， Adaptive Server 会显示专用于监控的配置参数的完整列表：

```
sp_configure Monitoring
```

用于控制监控信息收集的配置参数有：

- enable monitoring
- deadlock pipe active
- deadlock pipe max messages
- errorlog pipe active
- errorlog pipe max messages
- max SQL text monitored
- object lockwait timing
- per object statistics active
- plan text pipe active
- process wait events
- sql text pipe active
- sql text pipe max messages
- statement pipe active
- statement pipe max messages
- statement statistics active
- SQL batch capture
- wait event timing

注释　有关配置参数的说明，请参见 《系统管理指南：第一卷》中的第 5 章 “设置配置参数”。

为管道错误消息分配内存

许多监控表使用内存缓冲区 （称为 “管道”）来收集监控数据。以下参数可控制为每个管道分配的内存量：

- deadlock pipe max messages
- errorlog pipe max messages
- sql text pipe max messages
- plan text pipe max messages
- statement pipe max messages

Adaptive Server 可以动态向管道中添加内存，但无法动态从中删除内存，因此，如果减小管道参数的大小，则必须重新启动 Adaptive Server 才能让新的管道大小生效。

以下是用于确定这些参数的大小的算法：

- 对于各个 Adaptive Server，每个管道配置需要的内存为：

  configuration_value x *number_of_engines*

- 在集群环境中，每个集群实例都分配创建监控表管道所需的内存。请参见第 15 页的 “在集群环境中使用监控表”。

监控表的配置参数

Adaptive Server 使用配置参数来控制为监控表收集哪些数据。许多监控表要求您在 Adaptive Server 收集数据之前启用配置参数。必须先向用户授予 mon_role，然后他们才能查询某些监控表。

如果您不使用某个监控表，请禁用关联的配置参数，从而减少 Adaptive Server 上因收集监控数据导致的负载。

表 1-1 列出所有监控表及其适用的配置参数。

表 1-1：某些监控表需要的配置参数

	enable monitoring	SQL batch capture	deadlock pipe active	deadlock pipe max messages	errorlog pipe active	errorlog pipe max messages	object lockwait timing	per object statistics active	plan text pipe active	lock timeout pipe active	lock timeout pipe max messages	plan text pipe max messages	process wait events	SQL text pipe active	SQL text pipe max messages	statement cache size	statement pipe active	statement pipe max messages	statement statistics active	wait event timing	max SQL text monitored	enable stmt cache monitoring	capture compression statistics
monCachePool	X																						
monCachedObject																							
monCachedProcedures	X																		X				
monCachedStatement	X															X						X	
monCIPC																							
monCIPCEndpoints																							
monCIPCLinks																							
monCIPCMesh																							
monCLMObjectActivity	X																						
monClusterCacheManager																							
monCMSFailover																							
monDataCache	X																						
monDBRecovery																							
monDBRecoveryLRTypes																							
monDeadLock	X		X	X																			
monDeviceIO	X																						
monDeviceSpaceUsage																							
monEngine	X																						
monErrorLog	X				X	X																	
monFailoverRecovery																							
monInmemoryStorage																							
monIOController																							
monIOQueue	X																						
monLicense																							
monLocks	X																						
monLockTimeout	X									X	X												

	enable monitoring	SQL batch capture	deadlock pipe active	deadlock pipe max messages	errorlog pipe active	errorlog pipe max messages	object lockwait timing	per object statistics active	plan text pipe active	lock timeout pipe active	lock timeout pipe max messages	plan text pipe max messages	process wait events	SQL text pipe active	SQL text pipe max messages	statement cache size	statement pipe active	statement pipe max messages	statement statistics active	wait event timing	max SQL text monitored	enable stmt cache monitoring	capture compression statistics
monLogicalCluster																							
monLogicalClusterAction																							
monLogicalClusterInstance																							
monLogicalClusterRoute																							
monNetworkIO	X																						
monOpenDatabases	X																						
monOpenObjectActivity	X						X	X															
monOpenPartitionActivity	X							X															
monPCIBridge																							
monPCIEngine																							
monPCISlots																							
monPCM																							
monProcedureCache	X																						
monProcedureCacheMemoryUsage																							
monProcedureCacheModuleUsage																							
monProcess	X																			X			
monProcessActivity	X																			X			
monProcessLookup																							
monProcessMigration																							
monProcessNetIO	X																						
monProcessObject	X							X															
monProcessProcedures	X																		X				
monProcessSQLText	X	X																			X		
monProcessStatement	X																						
monProcessWaits	X												X							X			

	enable monitoring	SQL batch capture	deadlock pipe active	deadlock pipe max messages	errorlog pipe active	errorlog pipe max messages	object lockwait timing	per object statistics active	plan text pipe active	lock timeout pipe active	lock timeout pipe max messages	plan text pipe max messages	process wait events	SQL text pipe active	SQL text pipe max messages	statement cache size	statement pipe active	statement pipe max messages	statement statistics active	wait event timing	max SQL text monitored	enable stmt cache monitoring	capture compression statistics
monProcessWorkerThread	X																						
monRepLogActivity																							
monRepScanners																							
monRepScannersTotalTime																							
monRepSenders																							
monSQLRepActivity	X							X															
monSQLRepMisses	X							X															
monState																							
monStatementCache	X															X						X	
monSysLoad																							
monSysPlanText	X								X			X											
monSysSQLText	X	X												X	X						X		
monSysStatement	X							X									X	X	X				
monSysWaits	X																			X			
monSysWorkerThread	X																						
monTableColumns																							
monTableCompression	X							X															X
monTableParameters																							
monTables																							
monTableTransfer																							
monTask																							
monTempdbActivity	X						X	X															
monThread																							
monThreadPool																							
monWaitClassInfo																							
monWaitEventInfo																							

	enable monitoring	SQL batch capture	deadlock pipe active	deadlock pipe max messages	errorlog pipe active	errorlog pipe max messages	object lockwait timing	per object statistics active	plan text pipe active	lock timeout pipe active	lock timeout pipe max messages	plan text pipe max messages	process wait events	SQL text pipe active	SQL text pipe max messages	statement cache size	statement pipe active	statement pipe max messages	statement statistics active	wait event timing	max SQL text monitored	enable stmt cache monitoring	capture compression statistics
monWorkload																							
monWorkloadPreview																							
monWorkloadProfile																							
monWorkloadRaw																							
monWorkQueue																							

错误 12036

如果您查询监控表，但未启用这些表需要的所有配置参数， Adaptive Server 会发出错误 12036，但仍运行该查询。虽然即使您未启用所有配置参数，许多监控表中仍包含准确的数据，但某些数据是不正确的，因为 Adaptive Server 将不收集填充表中的一个或多个列所需的数据。

请考虑启用必需的配置参数。有关详细信息，请参见表 1-1。

mon_role 和其它访问控制

只有具有 mon_role 的用户才可以访问这些监控表。只有被授予此角色的用户可以在监控表上执行查询。可以授予或撤消特定登录名、角色或组对监控表的 select 权限，从而向部分 （或全部）监控表添加其它访问控制。有关获得角色的信息，请参见 《系统管理指南：第一卷》中的第 11 章 “管理用户权限”。

其中的一些监控表可能包含敏感信息。例如， monSysSQLText 和 monProcessSQLText 表中包含发送给 Adaptive Server 的所有 SQL 文本。此文本可能包含诸如对雇员薪水记录的更新之类的信息。管理员应考虑向这些表中添加更多访问限制 （如将访问权限限制为具有特定角色的用户），来满足其系统的安全性要求。

注释　如果您是在集群环境中使用监控表，则 Workload 和 LogicalCluster 监控表不需要 mon_role 权限。

计数器数据类型归零

监控表中的一些列包含整型计数器值，这些值在 Adaptive Server 的整个生存周期中是递增的。在计数器达到最高可能值 (2,147,483,647) 后，会将它重新设置为 0，这称为 “归零”。

因为有归零的可能，监控表中某些列的值可能不影响自服务器启动以来的累积的总计值。若要有效地使用此列数据，请计算计数器值在特定时间段内的差值，并使用此采样的结果 （而不是累计值）。例如，使用当前值和 10 分钟前的值之间的差值，而不是当前值。

不同计数器的值往往按不同比率增大。例如，在繁忙的系统上，monDataCache 表中的 LogicalReads 列急剧增大。使用 monTables 可确定可能归零的计数器； monTableColumns.Indicators 中的值 1 或 3 指定易于归零的列。服务器的行为取决于负载和应用程序行为， Indicator 列提供一般准则；请检查您的服务器数据以确定容易归零的计数器。

若要显示是计数器的列的列表，请执行以下操作：

```
select TableName, ColumnName
from master..monTableColumns
where (Indicators & 1) = 1
```

有态历史监控表

许多监控表提供各个事件的记录，而不是有关当前状态的信息。这些表称为 “历史的”，因为这些表中报告的事件提供服务器在一段时间内的历史记录。 Adaptive Server 维护访问历史表的每个客户端连接的上下文信息，而且，在对该表进行的每个后续查询中仅返回客户端以前没有收到的行。历史监控表的这种 “有态”属性旨在最大限度地提高性能，以及在用于填充历史数据的存储库时避免重复行。

历史监控表有：

- monErrorLog
- monDeadLock
- monSysStatement
- monSysSQLText
- monSysPlanText

注释 在 monSysPlan Text 和 monSysSQLText 中，列 BatchID、ContextID、ProcedureID 和 PlanID 的值经过修改，在 Adaptive Server 15.0.3 版和更高版本中生效。有关这些列中的更改，请参见 《参考手册：表》。

可以根据 monTables.Indicators 列来确定历史表：

```
select TableName
from master..monTables
where Indicators & 1=1
```

从历史表中返回的信息存储在缓冲区中，每个历史监控表存储在一个缓冲区中。这些缓冲区的大小 （由配置参数指定）影响数据存储的时间长度。使用 sp_configure 选项可配置缓冲区的大小和要捕获的信息。您使用的 sp_configure 选项取决于您要配置的监控表。例如，对于 monSysPlanText 表，请配置：

- plan text pipe max messages — 要为特定缓冲区存储的消息数。
- plan text pipe active — 指示 Adaptive Server 是否将信息写入缓冲区。

下表列出影响历史监控表的配置参数：

监控表	配置参数
monErrorLog	errorlog pipe active errorlog pipe active messages
monDeadLock	deadlock pipe active deadlock pipe max messages
monSysStatement	statement pipe active statement pipe max messages
monSysSQLText	sql text pipe active sql text pipe max messages
monSysPlanText	plan text pipe active plan text pipe max messages
monProcessSQLText 和 monSysSQLText	max SQL text monitored

注释　某些历史表要求您除了上面所列的配置参数，还要设置其它配置参数。请参见第 17 页的表 1-3。

max messages 参数的值决定每个引擎的最大消息数。将此值乘以配置的引擎数便可确定能存储的消息总数。

每条已存储的消息都会向监控表中添加一行。如果缓冲区中所有条目都已被使用，则新消息会覆盖缓冲区中的旧消息，因此仅返回最新的消息。

有关 sp_configure 的详细信息，请参见 《系统管理指南：第一卷》中的第 5 章 “设置配置参数”和第 4 页的 “配置监控表以收集数据”。

Adaptive Server 仅返回自上次读取以来添加的数据，所以您从尝试使用 where 子句过滤结果的查询获得的结果集可能看起来不一致，原因如下：

- 监控表中的 select 将表中所有以前未读的消息标记为已读。
- Adaptive Server 语言层执行过滤，因此，查询的结果集中不包含的行仍被连接认为是 “已查看”。

在以下示例中，与 monErrorLog 表关联的缓冲区包含两条消息：

```
select SPID, ErrorMessage
from master..monErrorLog
SPID      ErrorMessage
------    -------------------------------------
20        An error from SPID 20
21        An error from SPID 21

(2 rows affected)
```

如果您重新连接，将返回这两条消息，但是在使用 where 子句过滤结果集时，您会收到以下消息：

```
select SPID, ErrorMessage
from master..monErrorLog
where SPID=20
SPID         ErrorMessage
------      -------------------------------------
20           An error from SPID 20
(1 row affected)
```

和：

```
select SPID, ErrorMessage
from master..monErrorLog
where SPID=21
SPID         ErrorMessage
------      -------------------------------------
(0 rows affected)
```

因为第一个查询移动了客户端连接的上下文，从而包括 spid 20 和 21 的行，所以第二个查询不返回这两行。第一个查询中指定的过滤器要求服务器检索并计算这两行以返回指定的结果。Adaptive Server 将 spid 21 的行标记为 “已读”，即使它不参与返回到客户端连接的结果集也是如此。

注释 鉴于历史监控表的有态性质，不要将它们用于即席查询，而应使用：`select * into` 或 `insert into` 将数据保存到存储库或临时表中，然后对保存的数据进行分析。

瞬时监控数据

由于监控表经常包含瞬时数据，因此在查询中连接或使用集合时要小心：如果查询计划要求多次查询表，则这些操作返回的结果可能不同。例如：

```
select s.SPID, s.CpuTime, s.LineNumber, t.SQLText
from master..monProcessStatement s, monProcessSQLText t
where s.SPID=t.SPID
and s.CpuTime = (select max(CpuTime) from master..monProcessStatement)
```

此示例查询 monProcessStatement 两次，首先查找最大 CpuTime，然后匹配这个最大值。当 Adaptive Server 执行第二次查询时，从 monProcessStatement 返回的潜在结果有三种：

- 该语句执行更多工作，消耗更多 CPU，使 CpuTime 值大于前一个最大值，因此， where 子句中没有匹配项，查询不返回任何结果。
- 该语句在第二个查询执行之前就已执行完毕，不生成任何结果，除非另一个语句使用的 CPU 量完全和上次获取的最大值相同。
- 该语句不使用任何其它的 CPU，并且它的 CpuTime 值仍然与最大值匹配。只有这种情况会生成预期的结果。

Sybase 建议您在分析监控表中的数据之前，将其保存在临时表或存储库中。这样做会冻结数据并消除由于瞬时数据或历史监控表的无态性质可能造成的意外结果。

在集群环境中使用监控表

在集群环境中，缺省情况下，监控表按每个实例进行报告 （也就是说，集群中的单个服务器），而不是返回集群范围的结果。因此，您可以在整个集群内监控进程和查询的活动，更好地掌握可能在多个实例上打开的对象的统计信息以及集群中每个实例上的资源使用情况。例如，如果您在监控表中查询有关某个表的信息，而您要查询的表可能被集群中的多个实例打开或访问，因此，该表的描述符 （以及关联的统计信息）可能位于实例上的内存中。统计信息不是按集群进行聚合的。而是会返回一个由所有实例的统计结果组成的联合结果集，其中的各行是分别从每个实例中收集的。结果集中的每个实例都是用所在行的 InstanceID 列标识的。

配置系统视图

在集群服务器中， system_view 是特定于会话的设置，允许您控制查询从监控表、 sysprocesses、 sp_who 和其它命令返回的监控数据的作用域。如果将 system_view 设置为 cluster，监控表查询会返回集群中所有活动实例中的数据。如果将 system_view 设置为 instance，监控表查询会仅返回客户端所连接的实例上的活动进程或对象的数据。

使用 set 命令可以配置会话的作用域：

```
set system_view {instance | cluster | clear}
```

其中：

- instance — 仅返回本地实例的统计信息。跨集群请求不发送到集群中的任何其它实例。

- cluster — 返回集群中所有实例的统计信息。
- clear — 将系统视图返回到所配置的缺省设置。

如果您在查询监控表或调用监控表 RPC 时不指定 InstanceID，实例就会使用当前的 system_view 配置。

会话系统视图是从其主机逻辑集群中继承的。选择 *@@system_view* 全局变量可确定当前系统视图。

为监控实例添加的 InstanceID

表 1-2 描述 Cluster Edition 在其中添加 InstanceID 列的监控表。

表 1-2：含有 InstanceID 列的监控表

monCachePool	monDataCache
monCachedProcedures	monDeviceIO
monDeadLock	monErrorLog
monEngine	monIOQueue
monLicense	monLocks
monOpenDatabases	monNetworkIO
monOpenPartitionActivity	monOpenObjectActivity
monProcess	monProcedureCache
moProcessLookup	monProcessActivity
monProcessObject	monProcessNetIO
monProcessSQLText	monProcessProcedures
monProcessWaits	monProcessStatement
monResourceUsage	monProcessWorkerThread
monSysPlanText	monState
monSysStatement	monSysSQLText
monSysWorkerThread	monSysWaits
monCachedObject	

表 1-3 给出了对所有实例返回相同信息的监控表。

表 1-3：对所有实例包含相同信息的监控表

表名	说明
monMon	元数据视图在所有实例上都是相同的。
monTableColumns	元数据视图在所有实例上都是相同的。
monTableParameters	元数据视图在所有实例上都是相同的。
monTables	元数据视图在所有实例上都是相同的。
monWaitClassInfo	说明列表在所有实例上都是相同的。
monWaitEventInfo	说明列表在所有实例上都是相同的。

语句高速缓存的监控表

Adaptive Server 语句高速缓存一旦启用，便可存储即席 update、 delete 和 select 命令以及其它可能重用的语句的 SQL 文本。启用语句高速缓存后，这些语句的查询计划会被保存以便重用。发出新语句后， Adaptive Server 将搜索要重用的计划的语句高速缓存。如果 Adaptive Server 找到可重用的计划，则无需重新编译语句，因此会提高性能。

有关语句高速缓存的详细信息，请参见 《系统管理指南：第二卷》中的第 3 章 “配置内存”。

文字参数化能让 Adaptive Server 识别相同 （where 子句中的文字值不同除外）的查询。除性能优势外，文字参数化还会在将指标和语句存储到高速缓存时显著减少空间。

这些监控表包括两个可用于分析语句高速缓存的状态和性能的表：monStatementCache 提供语句高速缓存的摘要快照，monCachedStatement 显示有关每个高速缓存语句的详细信息。

配置 Adaptive Server 以监控语句高速缓存

使用 enable stmt cache monitoring 可配置 Adaptive Server 以收集有关语句高速缓存的监控信息。

从语句高速缓存中删除语句

使用 dbcc purgesqlcache 可从语句高速缓存中删除语句。如果指定语句 ID，则只会从高速缓存中删除相应的语句。

dbcc purgesqlcache 的语法为：

```
dbcc purgesqlcache (int SSQLID)
```

从 SQL 文本获取散列键

散列键是基于语句的文本生成的，并用作语句高速缓存中搜索机制的近似键。由于其它监控表显示语句的文本，因此您可以将散列键用作近似键来查找和比较这些表中的 SQL 文本。

有关查看高速缓存语句的整个 SQL 文本的信息，请参见下面的 “显示高速缓存语句的文本和参数信息”。

Adaptive Server 提供两个可供您高效计算散列键的函数。使用 parse_text 可先检验 SQL 文本是否有效，然后再计算散列键。语法为：

```
select parse_text(text, prm_opt)
```

prm_opt 的有效值为：

- 1 — 指示 parse_text 将自动参数化输出文本。
- -1 — 指示文本参数化的当前会话设置决定是否参数化输入文本。

如果 SQL 文本无效，则 parse_text 函数返回空值。

使用 hashbytes 可通过语句的文本计算散列键。例如：

```
select hashbytes('xor32', 'select * from syskeys')
```

显示高速缓存语句的文本和参数信息

使用 show_cached_text 可查看高速缓存语句的 SQL 文本。show_cached_text 将语句 ID 用作输入，并显示相应语句的文本和参数信息。语法为：

```
select show_cached_text(SSQLID)
```

使用 show_cached_text 可在查询中获取语句高速缓存中的语句的文本。例如：

```
select SSQLID, show_cached_text(SSQLID)
from master..monCachedStatement
```

查询监控表的示例

本节提供查询监控表的示例。

- 若要返回所有可用监控表的列表：

```
select TableName
from master..monTables
```

- 若要列出特定监控表中的列，请输入：

```
select ColumnName, TypeName, Length, Description
from master..monTableColumns
where TableName=”monProcessSQLText”
```

通过在 where 子句中替代任一监控表的名称并运行查询，您可以确定此监控表中存在哪些列。

- 若要确定当前执行的查询中哪些最消耗 CPU 并列出这些查询的文本，请输入：

```
select s.SPID, s.CpuTime, t.LineNumber, t.SQLText
from master..monProcessStatement s, master..monProcessSQLText t
where s.SPID = t.SPID
order by s.CpuTime DESC
```

- 若要确定 Adaptive Server 生存周期内的过程高速缓存的命中率，请输入：

```
select "Procedure Cache Hit Ratio" = (Requests-Loads)*100/Requests
from master..monProcedureCache
```

以下查询还提供数据高速缓存的命中率。在此示例中，命中率是针对 10 分钟的间隔计算的，而不是针对服务器的整个生存周期：

```
select * into #moncache_prev
from master..monDataCache
waitfor delay "00:10:00"
select * into #moncache_cur
from master..monDataCache
select p.CacheName,
"Hit Ratio"=((c.LogicalReads-p.LogicalReads) - (c.PhysicalReads -
p.PhysicalReads))*100 / (c.LogicalReads - p.LogicalReads)
from #moncache_prev p, #moncache_cur c
where p.CacheName = c.CacheName
```

若要计算特定采样期间的性能指标，请创建一个基线表，用以存储采样期间开始时的监控值。可以通过从采样期间结束时的值中减去基线值，来计算采样期间监控值的更改。

使用以下示例中的查询可计算数据高速缓存的命中率、创建基线、计算采样期间的活动量。

- 若要创建一个存储过程，用以输出已执行的 SQL 以及当前执行存储过程的所有进程的反馈，请输入：

```
create procedure sp_backtrace @spid int as
begin
select SQLText
from master..monProcessSQLText
where SPID=@spid
print "Stacktrace:"
select ContextID, DBName, OwnerName, ObjectName
from master..monProcessProcedures
where SPID=@spid
end
```

- 若要确定用于 dbid 为 5、对象 ID 为 1424005073 的数据库中的表的任何索引，请输入：

```
select DBID, ObjectID, LastUsedDate, UsedCount
from master..monOpenObjectActivity
where dbid=5 and ObjectID=1424005073 and IndexID > 1
```

若要确定可否删除不被在服务器上运行的应用程序使用的索引，请执行以下操作：

a 在访问相关表的应用程序中运行所有查询。确保 Adaptive Server 运行足够长时间，以便所有应用程序都执行其 select。

b 若要确定您的应用程序是否未在数据库中使用任何索引，请执行：

```
select DB = convert(char(20), db_name()),
TableName = convert(char(20), object_name(i.id, db_id())),
IndexName = convert(char(20),i.name),
IndID = i.indid
from master..monOpenObjectActivity a,
sysindexes i
where a.ObjectID =* i.id
and a.IndexID =* i.indid
and (a.UsedCount = 0 or a.UsedCount is NULL)
and i.indid > 0
and i.id > 99 -- No system tables
order by 2, 4 asc
```

第 2 章

等待事件

主题	页码
事件 19：xact coord：在空闲循环期间暂停	23
事件 29：等待常规缓冲区读取完成	23
事件 30：等待写入 MASS 而 MASS 正在更改	24
事件 31：在写入之前等待缓冲区写入完成	24
事件 32：等待 APF 缓冲区读取完成	25
事件 35：等待缓冲区验证完成	25
事件 36：在更改之前等待 MASS 完成写入	26
事件 37：在更改之前等待 MASS 完成更改	26
事件 41：等待获取闩锁	26
事件 46：等待缓冲区写入完成以便从 LRU 获取缓冲区	27
事件 51：等待 MASS 上的最后一个 i/o 完成	28
事件 52：等待 MASS 上被另一个任务启动的 i/o	28
事件 53：等待 MASS 完成更改以便开始 i/o	29
事件 54：等待最后一个日志页的写入完成	29
事件 55：在写入最后一个日志页后等待 i/o 完成	30
事件 57：检查点进程空闲循环	30
事件 61：hk：暂停一段时间	31
事件 70：等待设备信号	31
事件 83：等待 DES 状态更改	31
事件 84：等待检查点完成	32
事件 85：等待刷新程序将完整的 DFLPIECE 排入队列	32
事件 91：等待磁盘缓冲区管理器 i/o 完成	32
事件 99：等待来自客户端的数据	33
事件 104：等到引擎已脱机	33
事件 124：在获取页时等待簇读取完成	34
事件 142：等待逻辑连接释放	34
事件 143：暂停以便与节点管理器同步	34
事件 150：等待锁	35
事件 157：等待对象返回到池	35
事件 169：等待消息	36

主题	页码
事件 171：等待 CTLIB 事件完成	36
事件 178：等待分配新客户端套接字	37
事件 179：在无需网络读取或写入时等待	37
事件 197：等待读取在并行 dbcc 中完成	37
事件 200：等待并行 dbcc 中的页读取	38
事件 201：等待并行 dbcc 中的磁盘读取	38
事件 202：等待以并行方式重新读取页	39
事件 203：等待并行 dbcc 中的 MASS_READING 位	39
事件 205：等待并行 dbcc 中的 TPT 锁	39
事件 207：等待并行 dbcc 中将故障消息发送给父项	40
事件 209：等待管道缓冲区读取	40
事件 210：等待管道管理器中的可用缓冲区	41
事件 214：退让之后在运行队列中等待	41
事件 215：休眠之后在运行队列中等待	42
事件 222：复制代理在刷新期间休眠	42
事件 250：等待进站网络数据	42
事件 251：等待网络发送完成	43
事件 259：等到最后机会阈值被清除	43
事件 260：等待 waitfor 命令中的日期或时间	44
事件 266：等待工作线程邮箱中的消息	44
事件 272：等待 ULC 上的锁	45
事件 334：等待 Lava 管道缓冲区写入	45
事件 374：等待锁挂起 / 数据挂起被清除	45
事件 375：OCM 等待完成 BAST 处理	46
事件 389：OCM 等待推送数据标志被清除	46
事件 380：在 OCM_ERR_DIDNTWAIT 时锁定 / 数据挂起重置	47
事件 483：等待多播同步消息的确认	47

Adaptive Server 任务管理包括三个进程状态：正在运行、可运行和已阻塞。当进程不在运行 （在 CPU 上执行）时，它会处于以下状态：

- 在 CPU 上等待 （“可运行”状态）
- 因为磁盘或网络 I/O 而休眠
- 在资源上被阻塞 （锁、信号、螺旋锁等）

当服务器进程将自己挂起、休眠、等待另一个事件唤醒自己时，会发生等待事件。 Adaptive Server 为这些等待事件各提供一个唯一的等待事件 ID。查询 monSysWaits 和 monProcessWaits 可查找进程等待每个等待事件的次数 （总时间）。

注释　monSysWaits 表中的 WaitTime 的值以秒为单位。 monProcessWaits 表中的 WaitTime 的值以毫秒为单位。

本章讲述一些更常见的等待事件以及为避免它们可执行的操作。

事件 19：xact coord：在空闲循环期间暂停

Adaptive Server 事务协调器 (ASTC) 休眠，等待警报或服务器任务唤醒它 （ASTC 处理涉及多个数据库服务器的事务）。如果服务器不执行许多分布式事务，则此事件的 time per wait 接近 60 秒钟。

操作

无需执行任何操作。即使 WaitTime 的值很大，事件 19 也不影响总体性能。

事件 29：等待常规缓冲区读取完成

物理读取导致的等待 （最可能是高速缓存未命中），当 Adaptive Server 在数据高速缓存中找不到页而必须从磁盘读取它时，会发生这种情况。 Waits 数是由于高速缓存未命中而发生的物理读取的次数。使用 monSysWaits.WaitTime 值可派生 I/O 响应次数。

操作

因为此事件的 monSysWaits.WaitTime 值以秒为单位，所以此事件的 WaitTime 值应比 Waits 值小很多（平均物理读取应为 2ñ6 毫秒，大于 10 毫秒被认为是太慢）。平均物理读取值大可能表示磁盘吞吐量性能差。查询 monIOQueue 和 monDeviceIO 可确定太慢或过载的磁盘。

Waits 的值大 （无论 WaitTime 的值如何）可能表示查询计划不是很有效。如果遇到很大的 Waits 值，则说明可能发生了表扫描或笛卡儿乘积，或者优化程序可能由于统计信息的错误、过时或缺失而选择了不当的计划。请考虑在发生这种情况的表上添加针对特定列的索引。

Waits 值大还可能表示数据高速缓存太小，先是将活动页推出，然后重新读取。查询 monOpenObjectActivity、 monProcessActivity、 monDataCache、monCachPool 和 monProcessObject 可确定如何继续。

事件 30：等待写入 MASS 而 MASS 正在更改

Adaptive Server 正尝试写入内存地址空间段 (MASS)。MASS 是 Adaptive Server 在数据高速缓存中保存的一个或多个相邻页。但是，在此事件中，MASS 的状态为 “正在更改”，意思是另一个 spid 正在更新 MASS。启动此写入的 spid 无法写入 MASS，直到 MASS 不再被使用为止。

事件 30 的 WaitTime 值大可能表示数据高速缓存太小，导致数据高速缓存中的页经常到达清洗区，迫使 checkpoint 进程执行一些不必要的写入。

操作

您可能可以通过执行以下操作来减少太大的等待次数：

- 增大数据高速缓存的大小
- 使用高速缓存分区或命名高速缓存可分离内存密集型对象
- 调优管家、清洗标记位置或方案暗示 （如顺序键表）
- 定位清洗标记
- 调整方案 （如顺序键表）

事件 31：在写入之前等待缓冲区写入完成

负责向磁盘写入数据页的服务器进程 （例如，检查点）已确定它必须写入 MASS。但是，涉及同一页的更早的 write 操作还未完成，因此第二个进程必须等到第一个 write 完成后才能启动其 write 操作。

操作

通常，事件 31 的 WaitTime 值应小于 Waits 值。 WaitTime 值大可能表示磁盘争用或性能太低。查询 monIOQueue 和 monDeviceIO 可确定过载或太慢的磁盘。

事件 31 的 WaitTime 值大还可能表示数据高速缓存太小，导致数据高速缓存中的页经常到达清洗区，迫使 checkpoint 进程执行一些不必要的写入。

事件 32：等待 APF 缓冲区读取完成

当 Adaptive Server 在某个页上发出异步预取 (APF) 时，另一个进程正在读取该页所属的 MASS。 Adaptive Server 必须等待此读取完成才能继续。

操作

Waits 的值大可能表示 Adaptive Server 过于频繁地使用异步预取。调优高速缓存池的本地 APF 限制可能会减少 APF 页的争用。

由于 Adaptive Server 经常使用 APF 进行表扫描，涉及 APF 读取的争用可能表示由于索引缺失等因素，应用程序执行太多的表扫描。

事件 35：等待缓冲区验证完成

表示某个进程正在尝试读取另一个进程已读入高速缓存的页中的数据。在将某个页读入数据高速缓存后， Adaptive Server 会验证 read 操作是否成功。因为 Adaptive Server 正在验证 read 是否成功，所以第二个进程必须等待此验证完成才能访问数据。

此事件通常发生在物理读取次数很大的期间。

操作

事件 35 的 WaitTime 值应该相当小。如果该值很大，则说明许多进程在同时访问同一页，或者存在 CPU 争用。查询 monEngine 可确定引擎是否过载，运行系统级实用程序可确定是否存在总体 CPU 争用。

事件 36：在更改之前等待 MASS 完成写入

一个 spid 必须对 MASS 进行更改，但另一个 spid 当前正在写入该 MASS。第二个 spid 必须等到此写入完成。

操作

WaitTime 的值大表示某种情况可能正在导致 I/O 或数据高速缓存管理器性能下降。通常， Waits 的值应该大于 WaitTime 的值。查询 monIOQueue 和 monDeviceIO 可确定磁盘设备是否太慢或过载。

注释 如果由于页更新而发生了事件 36，则对高速缓存进行分区会是无效的。但是，如果事件 36 不是在发生页更新时发生的，则对高速缓存进行分区可能会加速写入。

事件 37：在更改之前等待 MASS 完成更改

一个 spid 尝试对 MASS 进行更改，但另一个 spid 当前正在更改该 MASS。第一个 spid 必须等到这些更改完成才能对该 MASS 进行更改。

操作

通常，事件 37 的 Waits 值应该比 WaitTime 值大很多。如果 Waits 值相比而言不是更大，则说明要么许多进程在同时访问同一 MASS，要么存在 CPU 争用。查询 monEngine 可确定引擎是否过载。运行系统级实用程序可确定是否存在总体 CPU 争用。

事件 41：等待获取闩锁

事件 41 通常表示多个进程在同时尝试更新单个页上的行。

Adaptive Server 使用闩锁作为瞬时锁来保证在另一个进程读取或写入数据时页的内容不被更改。当多个进程同时读取或更新同一页上的行时，Adaptive Server 通常在仅数据锁定表中使用闩锁来保护页的内容。如果一个进程在尝试获取闩锁时，另一个进程已经持有该闩锁，则第一个进程可能需要等待。如果事件 41 经常发生，则可能表示索引或表中的单个物理页上的数据有很高程度的争用。

通过执行以下操作减少争用：

- 引入一个索引，其排序顺序跨页按不同方式分布数据，使其分布导致争用的行
- 更改应用程序，以便不发生这种争用

操作

考虑通过更改索引定义 （从而改变数据在表中跨数据和索引页的物理分布）或修改应用程序来减少页争用。

如果 WaitTime 的平均值很大，则可能会由于 Adaptive Server 资源短缺而发生事件 41，其导致原因为：

- 散列表对于锁而言太小，导致 Adaptive Server 必须搜索的散列链很长。
- 一个操作系统问题导致本应很快完成的调用成为瓶颈 （例如，因为操作系统资源的限制，启动本应立即返回的异步 I/O 发生阻塞）。
- 插入次数和扩展更新次数极大。经常发生页分配，而且分配页闩锁的争用导致 Waits 数很大。使用 dbcc tune(des_greedyalloc) 可减少这种争用。有关闩锁争用的信息，请参见 Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon （《性能和调优系列：使用 sp_sysmon 监控 Adaptive Server》）。

事件 46：等待缓冲区写入完成以便从 LRU 获取缓冲区

一个 spid 尝试从最近使用最少的 (LRU) 链获取缓冲区。但是，该缓冲区有未完成的写入，必须先完成这个写入才能使 Adaptive Server 将该缓冲区用于另外的页。

操作

事件 46 可能表示：

- 高速缓存太忙，使得位于 LRU 链结尾的缓冲区仍在处理。查询 monDataCache 和 monCachePool 可确定哪个高速缓存太忙。可能的解决方法有：增大高速缓存的大小；使用 sp_poolconfig 增大清洗大小；通过返回 enable housekeeper GC 增加管家活动。
- 磁盘写入需要太长时间才能完成。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 51：等待 MASS 上的最后一个 i/o 完成

当由于对某个对象进行了更改或从元数据高速缓存中删除了某个对象，进程将该对象的一组页写入磁盘时，会发生此事件。因为在写入某些页之前一定要先完成在其它页上的 I/O 操作，所以此进程必须等到通知它已启动的 I/O 完成了其任务。

操作

WaitTime 的值大表示写入可能需要很长时间才能完成。通常， Waits 的值应该比 WaitTime 的值大很多。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 52：等待 MASS 上被另一个任务启动的 i/o

由于对某个对象进行了更改或从元数据高速缓存中删除了某个对象，进程将该对象的一组页写入磁盘。但是，另一个 spid 在 MASS 上有未完成的 I/O，第二个进程必须休眠直到第一个进程完成写入为止。

操作

此事件的 WaitTime 值大表示写入可能需要太长时间才能完成。通常，Waits 的值应该比 WaitTime 的值大很多。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 53：等待 MASS 完成更改以便开始 i/o

一个 spid 尝试写入 MASS，但另一个 spid 已在更改该 MASS，因此，第一个 spid 必须等到此更改完成。

Adaptive Server 最大限度地减少它所执行的磁盘 I/O 操作数。如果负责写入页的进程 （例如，检查点进程）需要修改页，但确定另一个进程正在修改该页，则第二个进程会等到第一个进程完成，所以，页写入包括页修改。

操作

通常，事件 53 的 Waits 值应该大于 WaitTime 值。如果它不是更大，则说明要么许多进程在同时访问同一 MASS，要么存在 CPU 争用。查询 monEngine 可确定引擎是否过载。运行系统级实用程序可确定是否存在总体 CPU 争用。

事件 54：等待最后一个日志页的写入完成

当进程正要启动最后一个日志页的写入但发现已经排定了另一个进程执行 write 时，会发生事件 54。第二个进程等到第一个进程完成其 I/O，但第二个进程不启动 I/O 操作。

因为 Adaptive Server 经常更新事务日志的最后一页，所以 Adaptive Server 避免执行最后一个日志页的物理写入。这会减少服务器执行的 I/O 量并增大最后一个日志页对其它需要执行更新的进程的可用性，从而可提高性能。

操作

事件 54 的平均 WaitTime 值大表示写入需要很长时间才能完成。通常，Waits 的值应该比 WaitTime 的值大很多。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

Waits 值大 （无论平均时间如何）可能表示最后一个日志页有争用。增大用户日志高速缓存的大小可减少争用，或者对应用程序的操作进行分组可避免提交每行。

事件 55：在写入最后一个日志页后等待 i/o 完成

表示进程已经在事务日志的最后一页上启动了 write 操作，必须在 I/O 完成之前一直休眠。事件 55 的 Waits 列值大表示 Adaptive Server 正在对事务日志进行大量更新，因为已提交的事务日志或其它操作要求将事务日志写入磁盘。

操作

事件 55 的 WaitTime 值大表示写入可能需要很长时间才能完成。通常，Waits 的值应该比 WaitTime 的值大很多。

事件 57：检查点进程空闲循环

检查点进程在运行之间休眠以避免检查点独占 CPU 时间。

操作

事件 57 可能累计自服务器启动时检查点进程启动以来的大量时间。但您无需根据此事件执行任何操作。

事件 61：hk：暂停一段时间

管家偶尔暂停以使管家功能不独占 CPU 时间。

操作

事件 61 是预料之中的，可能会显示运行了很长时间的服务器上的大值。通常，您无需根据此事件执行任何操作。

事件 70：等待设备信号

如果您在使用 Adaptive Server 镜像 （即， disable disk mirroring 设置为 0），则每个磁盘设备访问都必须先为该设备保留信号。事件 70 度量等待该信号所花费的时间，可能会在磁盘 I/O 结构太低时发生。

操作

如果您没在使用 Adaptive Server 镜像，则将 disable disk mirroring 设置为 1。如果您在使用镜像，则 WaitTime 值大可能表示因为设备争用而导致性能损失。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。对结果进行评估可确定能否将某些负载转移到其它设备上。

事件 83：等待 DES 状态更改

对每个打开的对象 （临时表、高速缓存查询计划与语句高速缓存、存储过程、触发器、缺省值、规则、表，等等）都会分配一个对象描述符（称为 “DES”）。当 Adaptive Server 释放分配的描述符 （通常，当 Adaptive Server 删除对象时会发生这种情况）时，会发生事件 83。

操作

事件 83 的 Waits 值大可能表示对象描述符的短缺。您可能需要增大打开的对象数。

事件 84：等待检查点完成

Adaptive Server 正在删除 DES，通常当 Adaptive Server 在删除对象时，会发生这种情况。事件 84 表示删除必须等待检查点在数据库上完成。

操作

虽然事件 84 的 Waits 值不可能很大，但大值可能表示同时发生了许多删除，或者检查点进程花费很长时间。如果检查点运行了过长时间，则尝试减少恢复间隔 （以分钟为单位）。

事件 85：等待刷新程序将完整的 DFLPIECE 排入队列

当 Adaptive Server 运行 dump database 时，它使用 “刷新程序”进程创建位于数据高速缓存中且已发生更改的页的列表（包括名为 DFLPIECE 的结构）。 Adaptive Server 向 Backup Server 发送一个将要包括在转储中的页的列表。

事件 85 度量转储进程等待刷新程序进程填充 DFLPIECE 并将其排入队列所花费的时间。

操作

此事件在 dump database 期间是正常的。如果 WaitTime 的平均值极大（大于 2），则检查其它事件以确定是什么减慢了刷新程序进程。

事件 91：等待磁盘缓冲区管理器 i/o 完成

当 Adaptive Server 运行 load database 时，它可能要求负载进程验证磁盘 I/O 是否完成才能继续。事件 91 度量 Adaptive Server 等待验证所花费的时间。

操作

通常，事件 91 的 WaitTime 值应该比 Waits 值小很多。WaitTime 值大表示可能存在磁盘争用或速度太慢。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 99：等待来自客户端的数据

当进程使用节点处理器连接到远程服务器时，它必须偶尔等待服务器返回数据。事件 99 度量进程必须等待的时间。

节点处理器是一种将 RPC 从本地服务器传输到远程服务器的方法。节点处理器在本地服务器和远程服务器之间建立单个物理连接，并建立 RPC 所需的多个逻辑连接。

操作

事件 99 的平均 WaitTime 值大表示与远程服务器的通信很慢。这可能是因为复杂的 RPC 调用需要很长时间才能完成、远程服务器发生了性能问题，或者网络太慢或过载。

事件 104：等到引擎已脱机

Adaptive Server 包括一个连续运行的引擎清除后台进程。该服务在引擎脱机后执行清除任务。该进程通常一直保持休眠状态，每隔 30 分钟唤醒一次以检查是否有要做的工作。事件 104 度量该进程在任务之间休眠的累计时间。

操作

事件 104 的平均 WaitTime 值应该非常接近 30。如果引擎经常脱机，则该值可能略小。如果 WaitTime 的平均值显著大于或小于 30，请与 Sybase 技术支持部门联系。

事件 124：在获取页时等待簇读取完成

当一个进程尝试执行物理读取，但另一个进程已执行了读取请求时，会发生事件 124 （这也算作 “高速缓存未命中”）。

操作

事件 124 的 WaitTime 值应该比 Waits 值小很多。如果磁盘性能很差，则平均 WaitTime 值会很大。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 142：等待逻辑连接释放

当 Adaptive Server 使用节点处理器机制在远程服务器上执行 RPC 时，它会创建逻辑连接。

当 Adaptive Server 必须关闭逻辑连接，但发现另一个进程在使用它时，会发生事件 142。 Adaptive Server 必须等到逻辑连接不再被使用，才能关闭逻辑连接。

操作

事件 142 通常应该有很小的平均 WaitTime 值。 WaitTime 的值大可能表示与远程服务器的通信存在问题。

事件 143：暂停以便与节点管理器同步

Adaptive Server 使用节点管理器与远程服务器通信，但另一个进程在尝试与该远程服务器连接。事件 143 度量 Adaptive Server 等待与服务器建立连接所花费的时间。

操作

事件 143 的平均 WaitTime 值大可能表示远程服务器上发生了性能问题或者网络太慢或过载。在 monProcessWaits 中查询 WaitEventID 143 可确定哪些 spid 的等待时间很长。

事件 150：等待锁

一个进程尝试在对象上获取逻辑锁，但另一个进程已经在该对象上持有一个冲突锁。事件 150 是一个常见事件，在 Adaptive Server 执行需要锁来保护被读取或更新的数据时发生。涉及的锁可能位于多个级别，包括表、页或行。

释放所有冲突的锁后， Adaptive Server 会唤醒正在等待的进程并向其授予访问该对象的权限。

操作

如果特定的表或页有争用 （例如，大量堆插入），则此事件的 WaitTime 值可能会很大。查询 monLocks 和 monOpenObjectActivity 可确定锁争用很严重的对象。

在某些情况下，您可以通过将表的锁定方案从所有页锁定更改为仅数据锁定来减少锁争用。应用程序或数据库设计通常会导致锁争用；请评估您的应用程序设计以确定减少争用的最佳方法，同时仍要考虑其它应用程序要求。

事件 157：等待对象返回到池

Adaptive Server 内存管理器为存储数据分配内存 （描述单独的内存“池”中的范围广泛的内部对象）。当池的可用内存很低时，增加内存的请求可能会被延迟，直到另一个操作向该池中返回内存。发生这种情况时，请求进程必须等到有更多内存可用。

当进程必须等到内存可用才能分配对象的数据时，会发生事件 157。

操作

如果事件 157 的平均 WaitTime 值很低，则性能可能会显著降低。但是，此事件上的任何 Waits 都表示在此情况下，您可以通过增大所配置的 Adaptive Server 等待的结构数来解决问题。使用 sp_countmetadata 和 sp_monitorconfig 可确定哪些结构正在使用最大配置，从而确定应增大哪些资源。

事件 169：等待消息

某些 Adaptive Server 进程 （例如，工作线程、审计、磁盘镜像，等等）使用名为 “邮箱” 的结构来传递消息。事件 169 度量 Adaptive Server 等待邮箱中的消息所花费的时间。

操作

通常，事件 169 的平均 WaitTime 值非常小。但是，如果 WaitTime 的值很大，则在 monProcessWaits 中查询 WaitEventID 值为 169 的行可确定哪些作业对于此事件的等待事件很长。

事件 171：等待 CTLIB 事件完成

表示 Adaptive Server 正在等待远程服务器响应。如果您将组件集成服务 (CIS) 用于代理表和 RPC 调用，则会出现事件 171。

操作

此事件的平均 WaitTime 值大可能表示远程 CIS 服务器发生了性能问题或者网络太慢或过载。在 monProcessWaits 中查询 WaitEventID 171 可确定哪些 spid 对于此事件的等待时间很长。

事件 178：等待分配新客户端套接字

网络监听器是一个处理客户端传入连接请求的 Adaptive Server 进程。事件 178 度量 Adaptive Server 等待新连接请求所花费的时间。

操作

您无需根据事件 178 执行任何操作。但您可以使用它的某些信息进行分析。 WaitTime 的值近似等于服务器已经运行的时间。 Waits 的值度量自服务器启动以来进行了多少次连接尝试。

事件 179：在无需网络读取或写入时等待

如果没有服务器必须发送或接收的网络 I/O， Adaptive Server 网络任务会在事件 179 上休眠。当有网络活动时，服务器任务便会唤醒，处理这些请求，然后回到休眠状态。

操作

事件 179 的值大表示网络活动量很大。如果网络活动量出奇地大，则查询其它监控表 （如 monNetworkIO 和 monProcessNetIO）可确定哪些作业降低网络性能。

事件 179 的 Waits 列值大可能表示 dbcc checkstorage 发现了大量的一致性故障。有关详细信息，请查看来自 dbcc checkstorage 的报告。

事件 197：等待读取在并行 dbcc 中完成

当您运行 dbcc checkstorage 时， Adaptive Server 必须偶尔在工作空间上执行异步 I/O 来读取或写入单个保留缓冲区。事件 197 度量 Adaptive Server 等待这些磁盘 I/O 所花费的时间。

操作

通常，事件 197 的 WaitTime 值应该比 Waits 值小很多。平均 WaitTime 值大可能表示磁盘吞吐量性能差。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 200：等待并行 dbcc 中的页读取

当您使用多个工作进程运行 dbcc checkstorage 时，会发生事件 200。此事件度量等待读取在 dbcc 检查的页上完成所花费的时间。

操作

通常，事件 200 的 WaitTime 值应该比 Waits 值小很多。平均 WaitTime 值大可能表示磁盘吞吐量性能差。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 201：等待并行 dbcc 中的磁盘读取

当您运行 dbcc checkverify 时， Adaptive Server 会执行磁盘读取来验证页的磁盘副本中是否存在潜在的故障；事件 201 度量等待这些读取完成所花费的时间。

操作

通常，事件 201 的 WaitTime 值应该比 Waits 值小很多。平均 WaitTime 值大可能表示磁盘吞吐量差。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 202：等待以并行方式重新读取页

当您运行 dbcc checkstorage 时， Adaptive Server 会确定它是否需要执行磁盘读取来验证页的磁盘副本中是否存在潜在的故障；事件 202 度量等待这些读取完成所花费的时间。

操作

通常，事件 202 的 WaitTime 值应该比 Waits 值小很多。平均 WaitTime 值大可能表示磁盘吞吐量差。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 203：等待并行 dbcc 中的 MASS_READING 位

当您运行 dbcc checkstorage 时， Adaptive Server 会确定它是否需要执行磁盘读取来验证 MASS 的磁盘副本中是否存在故障。但是，另一个进程可能已经启动该读取。事件 203 度量等待这些读取完成所花费的时间。

操作

通常，事件 203 的 WaitTime 值应该比 Waits 值小很多。平均 WaitTime 值大可能表示磁盘吞吐量差。查询 monIOQueue 和 monDeviceIO 可确定是否有太慢或过载的磁盘设备。

事件 205：等待并行 dbcc 中的 TPT 锁

当您运行 dbcc checkstorage 以检查 text 和 image 页时，Adaptive Server 必须持有锁以防止多个工作线程同时访问页链接。事件 205 度量等待这些锁所花费的时间。

操作

事件 205 的发生频率取决于您要检查的表中包含多少 text 和 image 列。WaitTime 的平均值极大可能表示持有锁的工作线程有一些资源争用。检查 CPU 和磁盘指标可确定是否有争用。

事件 207：等待并行 dbcc 中将故障消息发送给父项

当您运行 dbcc checkstorage 时，每个工作进程都会通过将消息排入父 spid 的队列中，将可能的故障报告给父进程。如果父进程的邮箱已满，则工作进程必须等待邮箱中有更多空间，然后才能将下一条消息排入队列。事件 207 度量工作进程为等待所花费的时间。

操作

事件 207 通常由报告大量故障的 Adaptive Server 导致。您无需对此事件进行任何操作，仅需按照正常过程运行 dbcc checkverify 验证和分析故障即可。

事件 209：等待管道缓冲区读取

当 Adaptive Server 以并行方式执行排序 （例如， create index，指定 `consumers` 子句）时，它使用内部机制在多个任务之间发送数据。事件 209 度量任务为等待其它任务向管道添加数据所花费的时间。

操作

事件 209 的平均 WaitTime 值应该非常低。 WaitTime 的平均值大可能表示排序管理器生产者进程无法足够快地生成数据以使消耗程序进程繁忙。检查总体系统性能可确定 Adaptive Server 是否有足够的 CPU 和 I\O 带宽。

事件 210：等待管道管理器中的可用缓冲区

当 Adaptive Server 以并行方式执行排序 （例如， create index，指定 `consumers` 子句）时，它使用内部机制 （称为管道）在多个任务之间发送数据。事件 210 度量进程等待 Adaptive Server 分配可用管道缓冲区所花费的时间。

操作

事件 210 的平均 WaitTime 值应该非常低。WaitTime 的平均值大可能表示 Adaptive Server 有一些资源争用。运行 sp_monitor 或 sp_sysmon 或者查询 monEngine 可确定 Adaptive Server 是否有足够的 CPU 资源。

事件 214：退让之后在运行队列中等待

事件 214 度量进程退让之后在运行队列中等待以便其它进程运行所花费的时间。该进程是 “可运行的”，不等待锁、物理 I/O 或任何其它等待条件。此事件可能由 CPU 不足 （即，服务器是 CPU 密集型的）或内存中的表扫描导致。

事件 214 不同于事件 215，因为它表示进程在执行一个超过了 time slice 所分配的 CPU 时间的 CPU 密集型任务：该进程主动放弃 CPU 并进入可运行状态，同时它会等待 Adaptive Server 调度程序分配更多 CPU 时间。发生这种情况时，该进程会继续进行它在放弃 CPU 之前所进行的活动。

事件 215 还表示进程处于可运行状态，但对于事件 214，进程进入此状态不是因为它超过了 CPU 时间，而是因为它在继续执行其任务之前遇到了必须等待资源的条件 （如磁盘或网络 I/O，或者逻辑锁）。

操作

繁忙的服务器通常具有很大的 Waits 值。但是， WaitTime 值或 time slice 设置大可能表示 Adaptive Server 有大量 spid 等待执行，或者正在运行的 spid 是 CPU 密集型的，而且不会轻易放弃 CPU。查询 monProcessActivity 可确定 CPUTime 大的作业。

事件 215：休眠之后在运行队列中等待

当进程不再等待另一个等待事件 （例如，逻辑锁、磁盘 I/O 或另一个等待事件）并进入服务器的可运行队列时，会发生事件 215。该进程必须等到调度程序分配 CPU 时间才能继续执行其任务。

有关事件 214 和 215 之间的不同，请参见事件 214 的说明。

操作

事件 215 是一般的等待事件。事件 215 的 Waits 值通常很大。繁忙的服务器会具有很大的 WaitTime，因为进程长时间等待 Adaptive Server 可运行队列。减少 time slice 的值能让更多进程访问 CPU （这也会减少某些进程在 CPU 中花费的平均时间），或者，如果主机上有足够的 CPU 可用，则增大联机引擎数。

事件 222：复制代理在刷新期间休眠

如果 Adaptive Server 是指定复制的主服务器，则 RepAgent 进程会休眠，等待要做的工作 （例如，向数据库的日志中添加行时）。事件 222 度量 RepAgent 休眠所花费的时间。

操作

根据复制型数据库中的活动级别，事件 222 通常可能有很大的 WaitTime 值。通常，您无需对此事件执行任何操作。

事件 250：等待进站网络数据

此事件度量应用程序进程处于活动状态，但等待客户端发来下一个请求（即，当作业处于 `AWAITING COMMAND` 状态时）的时间。

通常，当应用程序一直与 Adaptive Server 连接但处于空闲状态时，会发生事件 250。

操作

由于事件 250 是在 Adaptive Server 处理客户端的每个命令之前发生的，因此 Waits 和 WaitTime 数通常会很大。

可以使用事件 250 来估计服务器已经处理了客户端发来的多少请求。

此事件的 WaitTime 值大可能表示空闲客户端连接很多，或者某些客户端连接长时间保持空闲。此等待事件可能发生在客户端应用程序发送的批处理或命令之间，因此，如果应用程序提交大量单独的命令或批处理，Waits 值可能会很大。

事件 251：等待网络发送完成

事件 251 度量作业在向客户端发回答复数据包时所等待的时间。

操作

事件 251 可能表示 Adaptive Server 正在向客户端发送大答复集，或者可能表示网络太慢或过载。检查 monNetworkIO 和 monProcessNetIO 表中的平均包大小。在这些表中，每个表的平均大小都是：

(BytesSent) / (PacketsSent)

增大客户端应用程序的网络包大小可能会提高网络性能。

事件 259：等到最后机会阈值被清除

当 Adaptive Server 超过数据库日志的最后机会阈值时，每个尝试分配更多日志空间的进程都会收到消息 7415 并转入休眠或挂起状态，同时等待可用的日志空间。事件 259 度量进程等待此空间所花费的时间。

操作

此事件的 Waits 值大可能表示某些数据库需要更大的日志段。 WaitTime 的平均值大可能表示您还未定义阈值过程，或者过程需要很长时间才能释放日志空间。

增大数据库上的事务转储频率或向日志段分配更多空间可能会减小 WaitTime 的值。

事件 260：等待 waitfor 命令中的日期或时间

当进程使用 waitfor 命令时，事件 260 是正常的、意料之中的。

操作

当进程使用 waitfor 命令时， Adaptive Server 会将其转入休眠状态，直到请求的时间到期。事件 260 度量此休眠时间。

事件 266：等待工作线程邮箱中的消息

Adaptive Server 工作线程通过称为邮箱的内部 Adaptive Server 机制彼此进行通信以及与父 spid 进行通信。事件 266 度量工作进程等待邮箱添加消息所花费的时间。

操作

若要估计事件 266，请根据 monSysWorkerThread.ParallelQueries 确定运行的并行查询数。如果每个查询的 WaitTime 的值很大，则说明 Adaptive Server 可能有资源短缺（通常是 CPU 时间）。WaitTime 的值大还可能表示对象上的分区不平衡，导致某些工作线程等待其它工作线程完成。

事件 272：等待 ULC 上的锁

每个进程都分配一个用户日志高速缓存 (ULC) 区，用于减少最后一个日志页上的争用。 Adaptive Server 使用一个锁保护 ULC，因为多个进程可以访问 ULC 的记录并强制刷新。事件 272 度量 ULC 为等待该锁所花费的时间。

操作

通常，事件 272 的平均 WaitTime 值相当小。 WaitTime 的平均值大可能表示等待其它事件的时间长，迫使 ULC 锁持有者等待。可以分析其它等待事件来确定是什么导致这些等待。

事件 334：等待 Lava 管道缓冲区写入

Adaptive Server 15.0 版引入了 lava 队列执行引擎。当此引擎执行并行查询时，它会使用叫做 “管道缓冲区”的内部结构在工作进程之间传送数据。事件 334 度量 Adaptive Server 等待管道缓冲区可用所花费的时间。

操作

当进程执行正常时， WaitTime 的值应该很小。如果仍是这种情况，请与 Sybase 技术支持部门联系。

事件 374：等待锁挂起 / 数据挂起被清除

锁请求被阻塞，因为请求的锁在本地节点不可用，是当前挂起的请求。

操作

同一个锁跨不同节点的冲突请求数大表示应用程序从多个节点发出在同一个对象上冲突的锁请求。锁从对象一致性管理器中发出，该管理器用于管理跨集群中所有实例描述数据库对象的元数据。

可以通过将对所需对象的大多数请求定向到单个节点来提高性能。

事件 375：OCM 等待完成 BAST 处理

一个或多个 ocm_lock 请求被其它节点发来的冲突 ocm_lock 请求阻塞。

操作

此等待事件的值大可能表示其它节点正在发送大量对 ocm_lock 的冲突请求。应用程序可能正在从多个节点发出这些冲突锁请求。可以通过将对 ocm_lock 的大多数请求定向到单个节点来提高性能。

事件 389：OCM 等待推送数据标志被清除

Adaptive Server 无法处理一个或多个 ocm_lock 请求，原因如下：

- 另一个锁请求正在持有该锁

  这种情况下，值大表示有许多排它锁请求需要执行大量操作而不失去 ocm_lock。这是内部问题。

  若要提高性能，请确定为何 Adaptive Server 如此长时间地持有锁。

- 另一个请求正在推送数据。

  这种情况下，值大可能表示数据仅驻留在内存中， Adaptive Server 必须定期将此数据推送到其它节点，以避免在节点发生故障时丢失数据。这将对性能产生不利影响，因为 Adaptive Server 在将数据推送到其它节点时不执行有用的工作。这是内部问题。

  若要提高性能，请确定为何 Adaptive Server 如此长时间地持有锁或推送数据。

- 一个取消的事务正在挂起（可能未在传送请求的数据，此节点上的 OCM 正在采取纠正措施）。

  此等待事件很少见（通常，当实例需要从刚崩溃的节点获得数据时，会发生此事件）。此等待事件的值大可能起因于节点或硬件故障率很高。

事件 380：在 OCM_ERR_DIDNTWAIT 时锁定 / 数据挂起重置

如果集群锁管理器无法立即授予锁请求，它会返回 LOCK_DIDNTWAIT 消息， Adaptive Server 将该消息转换为 OCM_ERR_DIDNT_WAIT。 ocm_lock 请求转入休眠状态，直到 AST 从 master 数据库返回并带有对锁请求的响应。

操作

此等待事件的值大表示跨不同节点有许多对同一 ocm_lock 的冲突请求。这可能由从多个节点发出在同一个对象上冲突的锁请求的应用程序导致。可以通过将对所需对象的请求定向到单个节点来提高性能。

事件 483：等待多播同步消息的确认

虽然此等待事件时有发生，但它是正常活动的结果。

操作

无须用户操作。

索引

数字

12036, 错误 10

英文

Adaptive Server, 为语句高速缓存配置 17
Cluster Edition
  使用监控表 15
  添加 **instanceID** 16
  在 15.0.1 版中安装监控表 3
enable cis 配置参数 4
enable monitoring 配置参数 5
installmontables 脚本 3
instanceID 列 , 用于 Cluster Edition 16
MASS （内存地址空间段）
  等待更改 , 等待事件 30 24
  定义的 24
max messages 参数 13
mon_role 2
  not required in **Workload** 和 **LogicalCluster** 表 11
  和其它访问控制 10
monCachedProcedures 表 2
monCachedStatement 表 17
monDeadLock 表 2
monProcessSQLText 表 2
monProcessWaits 表 2
monStatementCache 表 17
monSysWaits 表 2
prm_opt 选项 , 有效值 18
set 17
  命令 15
show_cached_text 函数 , 查看高速缓存语句的 SQL 文本 18
sp_configure, 存储过程 4
Transact-SQL
  用于监控性能 2
xact coord, 等待事件 19 23

A

安装
  15.0.1 Cluster Edition 的监控表 3
  15.0.2 版和更高版本的监控表 3
  低于 15.0.2 的版本的监控表 3
  监控表 3

B

表
  monCachedStatement 17
  monStatementCache 17

C

参数
  max messages 13
查询监控表 , 示例 19
存储过程
  sp_configure, 用于配置选项 4
错误 12036, 如何使用 10

D

等待
  APF 缓冲区完成 , 等待事件 32 25
  CTLIB 事件完成 , 等待事件 171 36
  DES 状态更改 , 等待事件 83 31
  I/O on MASS, 等待事件 52 28
  I/O 在写入后完成 , 等待事件 55 30
  Lava 管道缓冲区写入 , 等待事件 334 45
  MASS 上的最后一个 IO, 等待事件 51 28
  MASS 完成 , 等待事件 53 29
  MASS, 等待事件 36 26

MASS, 等待事件 37 26
ULC 上的锁 , 等待事件 272 45
waitfor 命令中的日期或时间 , 等待事件 266 44
并行 dbcc 中的 MASS_READING 位 , 等待事件 202 39
并行 dbcc 中的 TPT 锁 , 等待事件 205 39
并行 dbcc 中的磁盘读取 , 等待事件 201 38
并行 dbcc 中的页读取 , 等待事件 200 38
并行 dbcc 中将故障消息发送给父项 , 等待事件 207 40
磁盘缓冲区管理器 i/o 完成 , 等待事件 85 32
等到最后机会阈值被清除 , 等待事件 259 43
读取在并行 dbcc 中完成 , 等待事件 197 37
对象返回到池 , 等待事件 157 35
分配 , 等待事件 334 37
分配新客户端套接字 , 等待事件 179 37
工作线程邮箱中的消息 , 等待事件 202 44
管道管理器中的可用缓冲区 , 等待事件 210 41
管道缓冲区读取 , 等待事件 209 40
缓冲区事件 , 等待事件 31 24
缓冲区验证完成 , 等待事件 35 25
获取闩锁 , 等待事件 41 26
检查点完成 , 等待事件 84 32
进站网络数据 , 等待事件 250 42
来自客户端的数据 , 等待事件 99 33
逻辑连接释放 , 等待事件 142 34
设备信号 , 等待事件 70 31
事件 179, 等待分配新客户端套接字 37
事件 203, 等待以并行方式重新读取页 39
事件 266, 等待 waitfor 命令中的日期或时间 44
刷新程序将完整的 DFLPIECE 排入队列 , 等待事件 85 32
锁 , 等待事件 150 35
退让之后在运行队列中 , 等待事件 214 41
完成以便从 LRU 获取缓冲区 , 等待事件 46 27
网络发送完成 , 等待事件 251 43
无需网络读取或写入 , 等待事件 334 37
消息 , 等待事件 169 36
休眠之后在运行队列中 , 等待事件 215 42
以并行方式重新读取页 , 等待事件 202 39
以并行方式重新读取页 , 等待事件 203 39
引擎已脱机 , 等待事件 104 33
在获取页时等待簇读取完成 , 等待事件 124 34
最后一个日志页的写入 , 等待事件 54 29
等待事件
104, 等到引擎已脱机 33
124, 在获取页时等待簇读取完成 34
142, 等待逻辑连接释放 34
143, 暂停以便与节点管理器同步 34
150, 等待锁 35
157, 等待对象返回到池 35
169, 等待消息 36
171, 等待 CTLIB 事件完成 36
178, 等待分配 37
179, 在无需网络读取或写入时等待 37
19, xact coord 23
197, 等待读取在并行 dbcc 中完成 37
200, 等待并行 dbcc 中的页读取 38
201, 等待并行 dbcc 中的磁盘读取 38
202 等待以并行方式重新读取页 39
203, 等待并行 dbcc 中的 MASS_READING 位 39
205, 等待并行 dbcc 中的 TPT 锁 39
207, 等待并行 dbcc 中将故障消息发送给父项 40
209, 等待管道缓冲区读取 40
210, 等待管道管理器中的可用缓冲区 41
214, 退让之后在运行队列中等待 41
215, 休眠之后在运行队列中等待 42
222, 复制代理在刷新期间休眠 42
250, 等待进站网络数据 42
251, 等待网络发送完成 43
259, 等到最后机会阈值被清除 43
266, 等待工作线程邮箱中的消息 44
272, 等待 ULC 上的锁 45
29, 等待常规缓冲区读取 23
30, 等待写入 MASS 24
31, 等待缓冲区写入 24
32, 等待 APF 缓冲区完成 25
334, 等待 Lava 管道缓冲区写入 45
35, 等待缓冲区验证完成 25
36, 等待 MASS 26
37, 等待 MASS 26
41, 等待获取闩锁 26

46, 等待完成以便从 LRU 获取缓冲区 27
51, MASS 上的最后一个 IO 28
52, waiting for I/O on MASS 28
53, 等待 MASS 完成 29
54, 等待最后一个日志页的写入 29
55, 等待 I/O 在写入后完成 , 30
57, 检查点进程空闲循环 30
61 hk, 暂停一段时间 31
70, 等待设备信号 31
83, 等待 DES 状态更改 31
84, 等待检查点完成 32
85, 等待刷新程序将完整的 DFLPIECE 排入队列 32
91, 等待磁盘缓冲区管理器 i/o 完成 32
99, 等待来自客户端的数据 33
避免 21
定义 23

F

访问控制 , 在 **mon_role** 中 10
分配内存 , 为管道错误消息 6
复制代理在刷新期间休眠 , 等待事件 222 42

G

管道错误参数
列表 6
管道错误消息
分配内存 6
列表 6

H

函数
show_cached_text 18
缓冲区 , 为监控表配置 12
缓冲区读取 , 等待 , 事件 29 23

J

即席查询 , 不使用的表 14
计数器数据类型 , 归零 11
计数器数据类型归零 11
监控
使用 Transact-SQL 时的性能 2
信息来源 2
监控表 1–20
15.0.2 和更高版本的程访问和编辑 4
CIS 和 , 3
installmontables 脚本 3
mon_role 2
安装 3
不是缺省创建 1
查询 19
简介 1
客户端连接 13
配置缓冲区 12
配置选项 4
使用 Transact-SQL 监控性能 2
示例 19–20
受配置选项影响 6
数据不存储在磁盘上 1
瞬时数据 14
用于语句高速缓存 **update**、**select**、**delete** 命令 17
有态历史监控表 12–15
远程访问和编辑 4
在集群环境中使用 15
监控计数器
全局 2
监控信息的来源 2
检查点进程空闲循环 , 等待事件 57 30
角色
mon_role 10

K

客户端连接和监控表 13

L

历史监控表 , 列表 12

M

命令
set 15

P

配置参数
enable cis sp_configure, 用于配置选项 4
enable monitoring sp_configure, 用于配置选项 5
列表 5
某些监控表需要 7

Q

全局监控计数器 2

S

散列键 , 从 SQL 文本中获取 18
使用 **sp_configure** 配置监控表 4
视图 , 系统 , 配置 **system_view**, 设置 15
瞬时 （有态）数据和监控表 14
算法 , 确定管道错误参数的大小 6

X

系统视图 , 配置 15
选项
prm_opt 18

Y

有态监控表 12–15
语句高速缓存
配置 Adaptive Server 17
删除语句 18
远程访问监控表 4
在 15.0.2 版和更高版本中 4

Z

暂停一段时间 , 等待事件 61 hk 31
暂停以便与节点管理器同步 , 等待事件 143 34