亿级SQL Server运维的最佳实践

宋沄剑

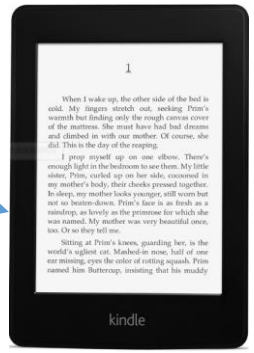
1# Not Only SQL Server


iPhone

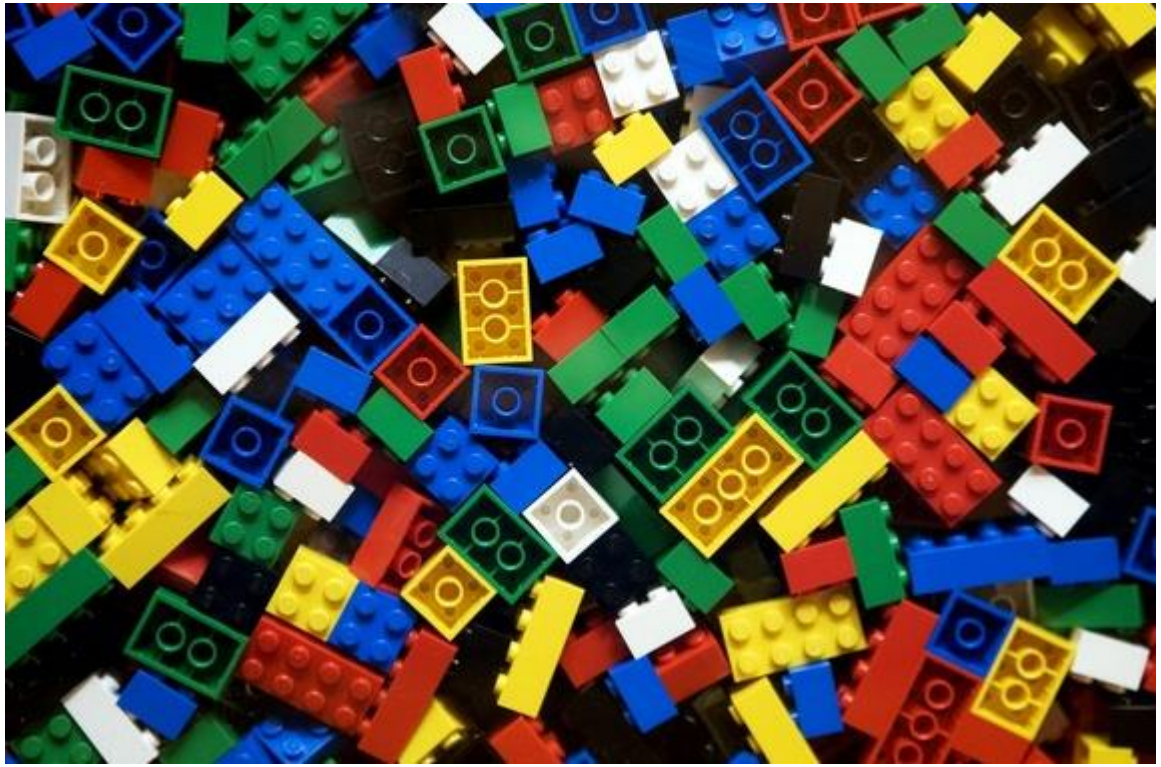

XBOX


Canon


AUTOMATIC
LONGINES


1
When I wake up, the other side of the bed is cold. My fingers stretch out, seeking Prim's warmth but finding only the rough canvas cover of the mattress. She must have had bad dreams and climbed in with our mother. Of course, she did. This is the day of the reaping.
I prop myself up on one elbow. There's enough light in the bedroom to see them. My little sister, Prim, curled up on her side, cocooned in my mother's body, their cheeks pressed together. In sleep, my mother looks younger, still worn but not so beaten-down. Prim's face is as fresh as a raindrop, as lovely as the primrose for which she was named. My mother was very beautiful once, too. Or so they tell me.
Sitting at Prim's knees, guarding her, is the world's ugliest cat. Mashed-in nose, half of one ear missing, eyes the color of rotting squash. Prim named him Buttercup, insisting that his muddy
kindle

关系数据库 VS NOSQL



SQL Server不应该是今天用到的唯一DB

- SQL Server是典型的关系数据库
- 对于OLTP来说，对应的数据应该是价值密度高的数据
- SQL Server用于NoSQL、时序数据、全文检索、消息数据，图（2017中支持）、层级数据则有很多额外开销
- 关系数据库不适用Schema频繁变更的场景（变更Schema绝对是痛不欲生的事情）
- 读远远大于写的操作类型或昂贵操作应该放入缓存（关系数据库ACID特性会给读操作带来大量额外开销）
- 关系数据库难以横向扩展（Scale-Out）
- 大量低价值密度数据的分析（Spark等分布式模型都会比CUBE等有更好的性能和扩展性）

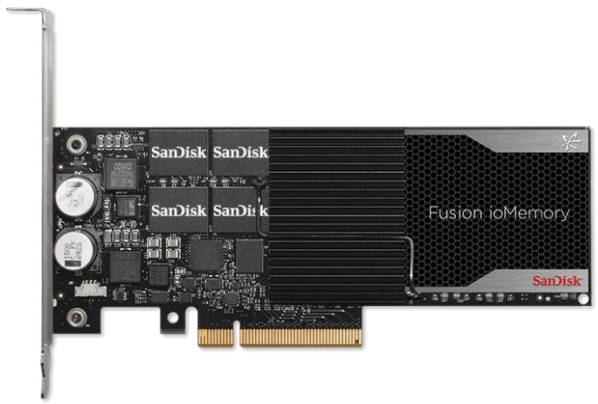
常用DB

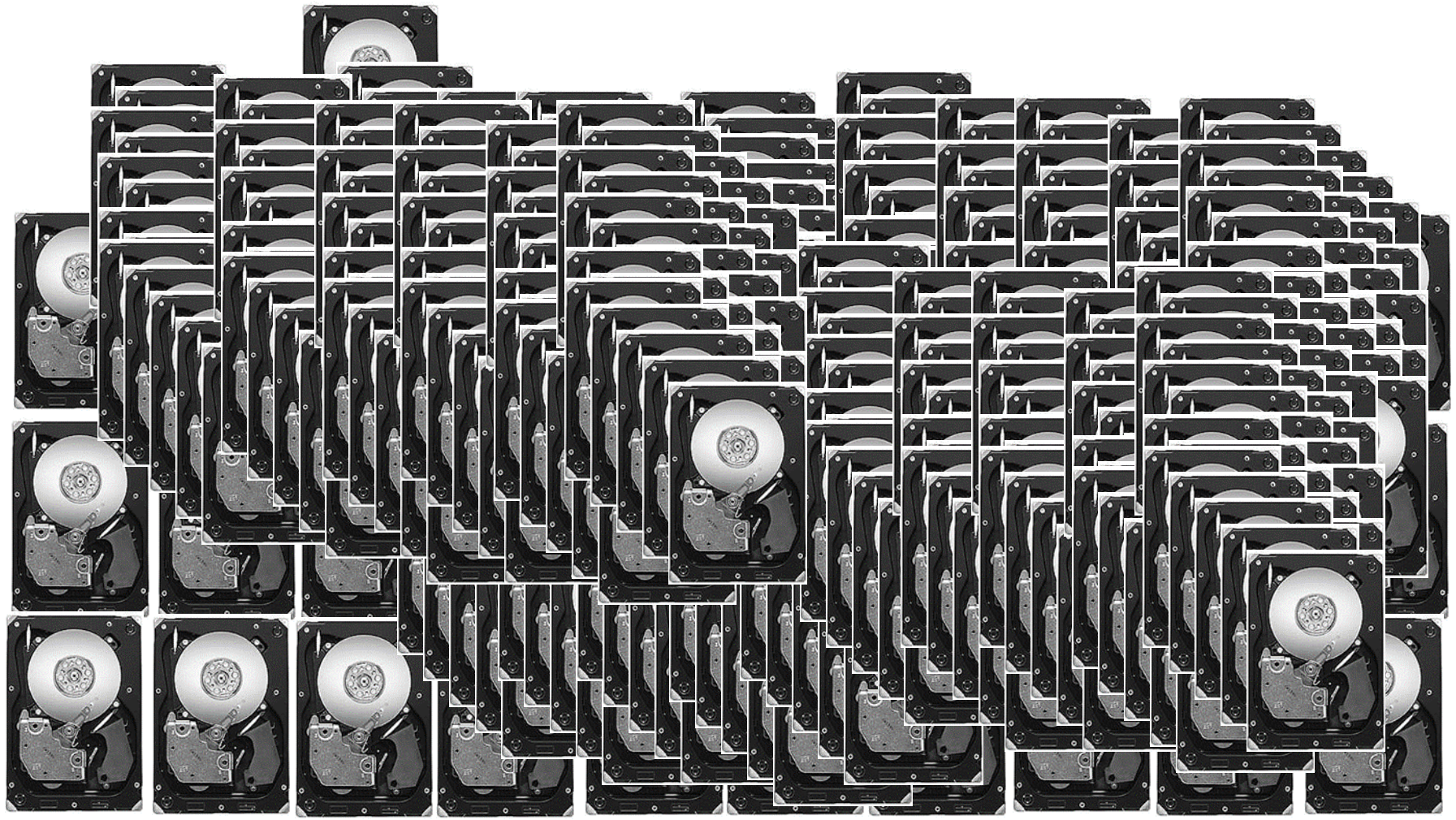
全文检索	ElasticSearch
NoSQL	MongoDB
时序数据库	InfluxDB Graphite
K-V	Redis Memory Cache
消息队列	Kafka RabbitMQ
图数据库	Neo4j
宽列	Cassandra Hbase

2# 硬件充足

充足的硬件或云资源能避免很多麻烦

- CPU
- IO
- 内存
- 网络


SanDisk
SanDisk
SanDisk
SanDisk
Fusion ioMemory
SanDisk



3# 缓存

SQL Server调优分为三个层面

- 应用程序设计
  - 缓存、数据归档
  - OLAP/OLTP分离、读写分离
- 数据库设计
  - 缺少索引、过多索引
  - 不适当的列类型、过度范式化、表分区
- 查询设计
  - 不好的查询语句（不适当的OR/LIKE/EXISTS操作、索引列转换）
  - 不恰当的使用视图/触发器/游标

大量用户

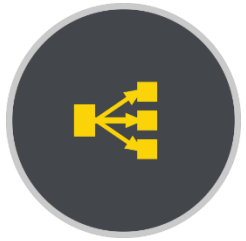
访问

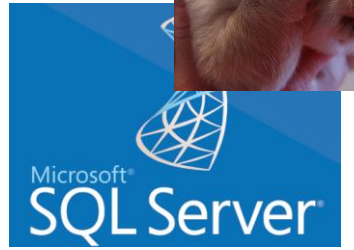
负载均衡

应用服务器

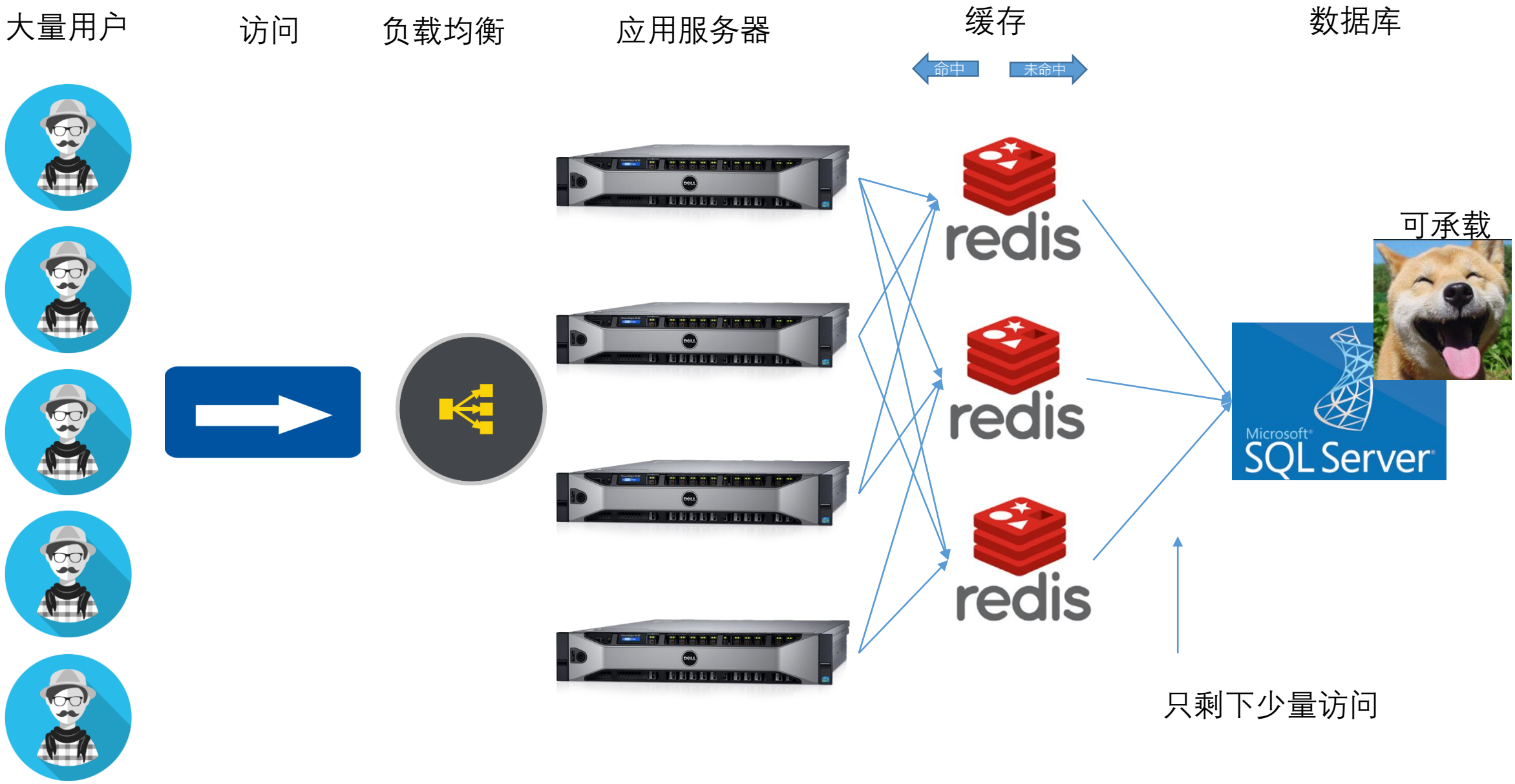
每秒大量访问

数据库





过载！

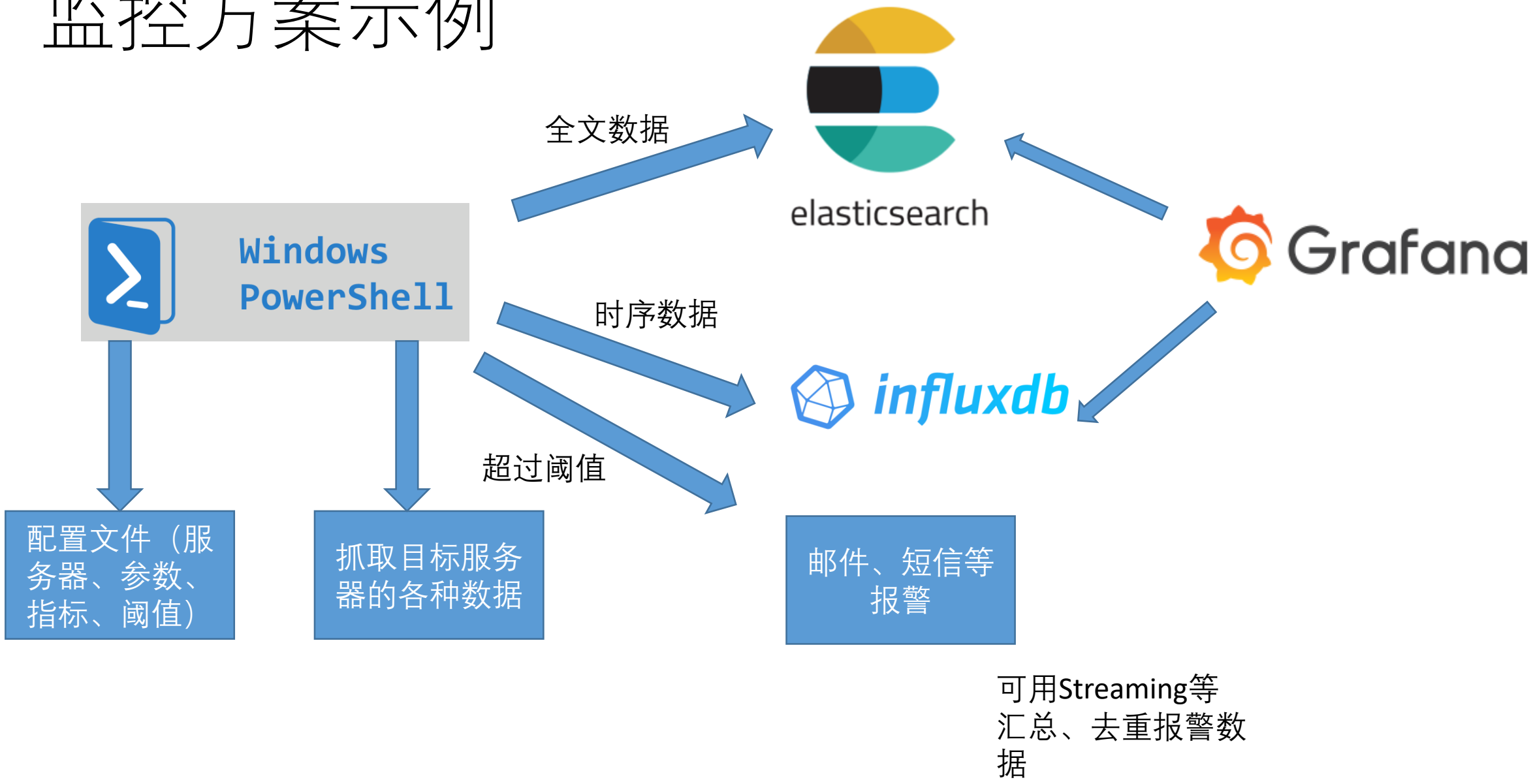

大量用户
访问
负载均衡
应用服务器
缓存
数据库
命中
未命中
redis
redis
redis
Microsoft
SQL Server
可承载
只剩下少量访问

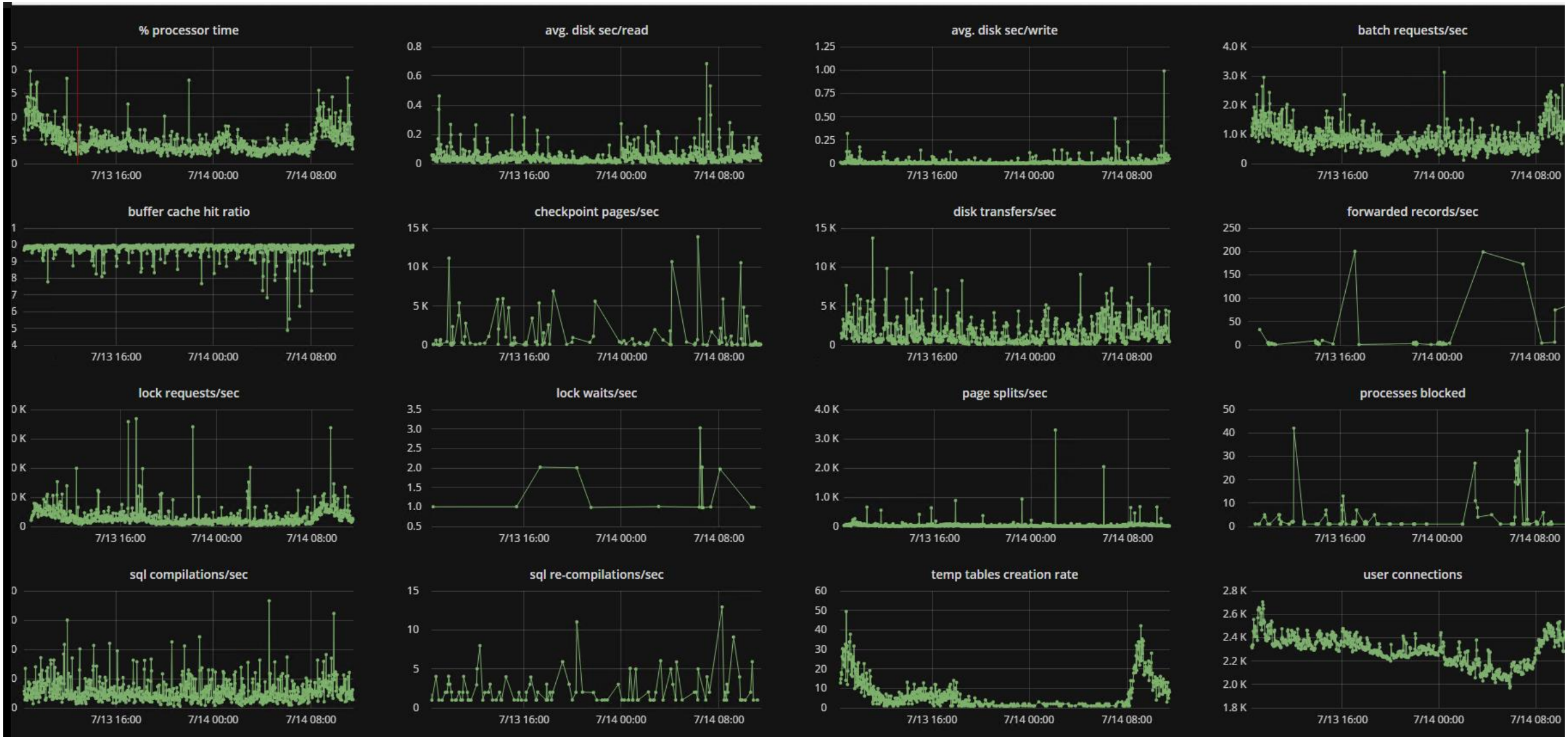
4# 监控

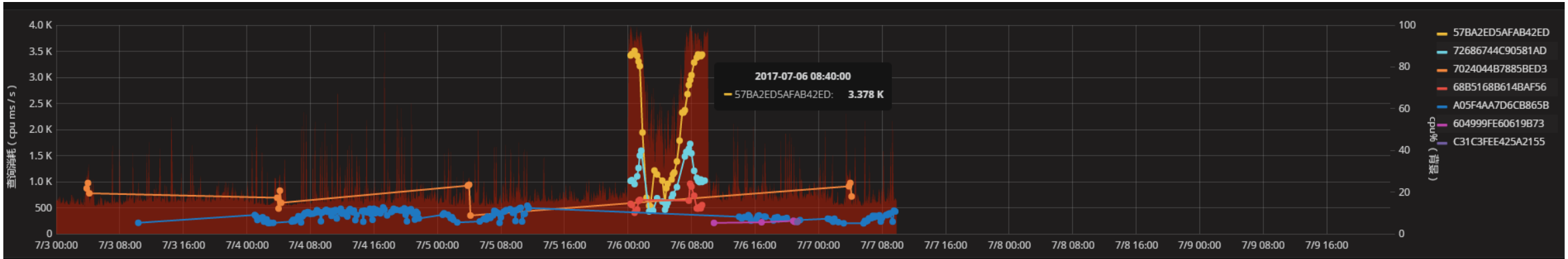
不仅仅是常规监控

- CPU使用率、端口能否通、磁盘空间等（常规Zabbix等通用工具可完成）
- 各种性能计数器
- 高消耗语句
- 超时语句
- 异常作业
- DDL、DML审核

监控方案示例

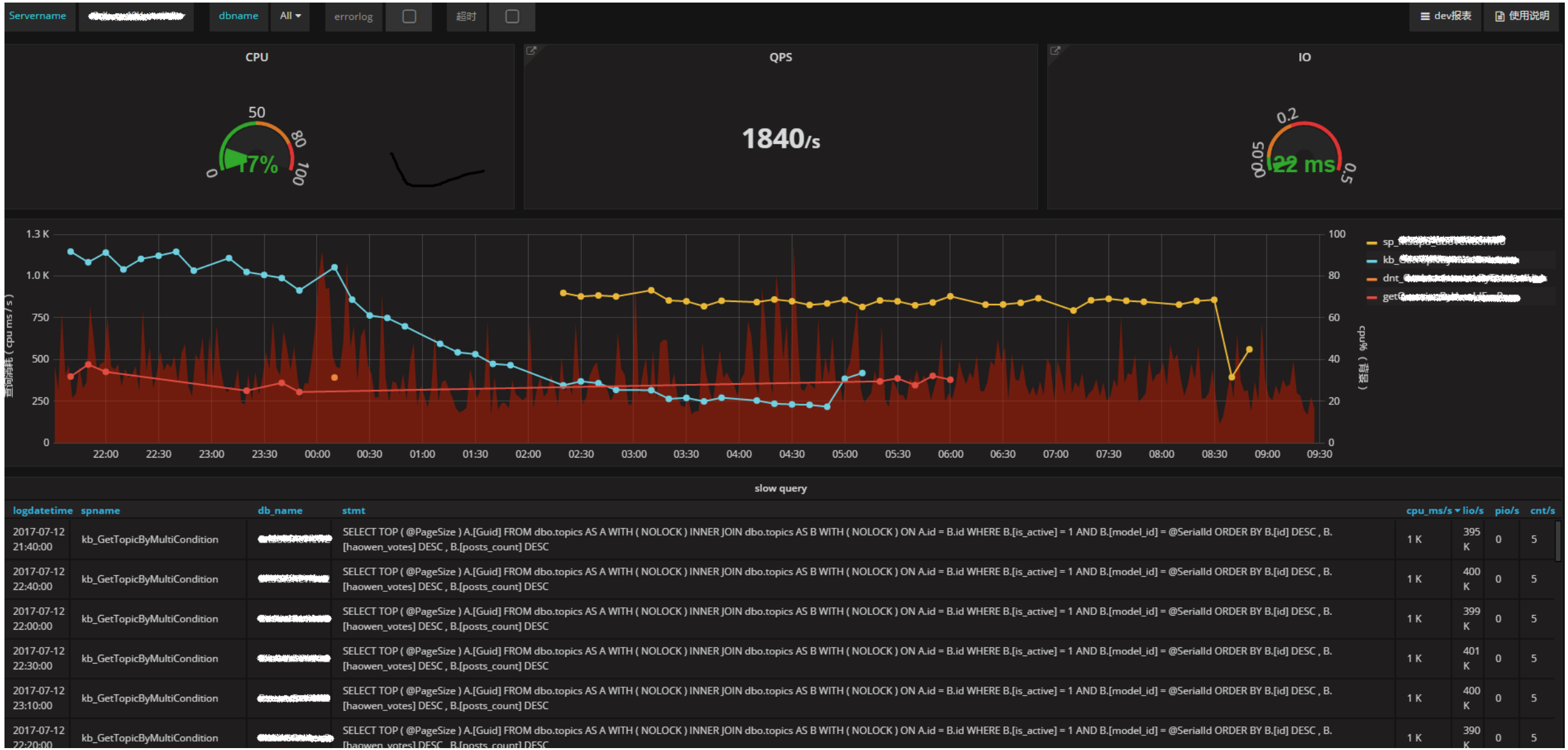

全文数据
elasticsearch
Grafana
Windows PowerShell
时序数据
influxdb
超过阈值
配置文件（服务器、参数、指标、阈值）
抓取目标服务器的各种数据
邮件、短信等报警
可用Streaming等汇总、去重报警数据


% processor time
avg. disk sec/read
avg. disk sec/write
batch requests/sec
buffer cache hit ratio
checkpoint pages/sec
disk transfers/sec
forwarded records/sec
lock requests/sec
lock waits/sec
page splits/sec
processes blocked
sql compilations/sec
sql re-compilations/sec
temp tables creation rate
user connections
7/13 16:00
7/14 00:00
7/14 08:00

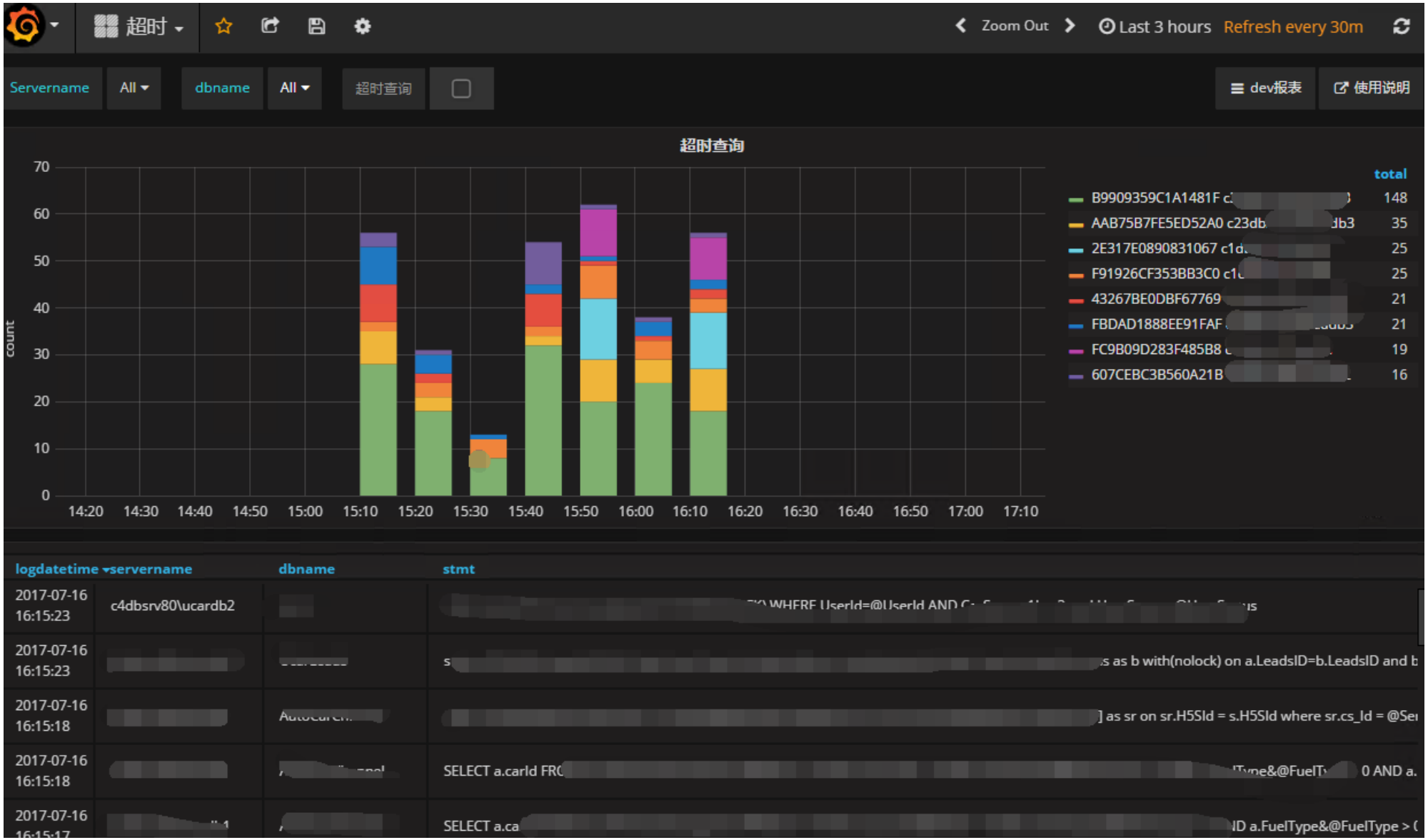


slow query

logdatetime	spname	db_name	stmt	cpu_ms/s	lio/s	pio/s	cnt/s
2017-07-06 00:50:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	4 K	1 Mil	0	16
2017-07-06 00:30:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	1 Mil	0	16
2017-07-06 08:50:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	2 Mil	0	16
2017-07-06 09:20:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	2 Mil	0	16
2017-07-06 00:20:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	1 Mil	0	16
2017-07-06 09:00:00	57BA2ED5AFAB42ED	[redacted]	SELECT @iCount = COUNT(1) FROM [redacted] WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	2 Mil	0	15
2017-07-06 01:10:00	57BA2ED5AFAB42ED	MobileData	SELECT @iCount = COUNT(1) FROM queryRecord WITH(nolock) WHERE 1=1 AND CiytID=@LocationID AND CsId=@CarSerialID	3 K	1 Mil	0	15



logdatetime	spname	db_name	stmt	cpu_ms/s	lio/s	pio/s	cnt/s
2017-07-12 21:40:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	395 K	0	5
2017-07-12 22:40:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	400 K	0	5
2017-07-12 22:00:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	399 K	0	5
2017-07-12 22:30:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	401 K	0	5
2017-07-12 23:10:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	400 K	0	5
2017-07-12 22:20:00	kb_GetTopicByMultiCondition	[redacted]	SELECT TOP (@PageSize) A.[Guid] FROM dbo.topics AS A WITH (NOLOCK) INNER JOIN dbo.topics AS B WITH (NOLOCK) ON A.id = B.id WHERE B.[is_active] = 1 AND B.[model_id] = @Serialld ORDER BY B.[id] DESC , B.[haowen_votes] DESC , B.[posts_count] DESC	1 K	390 K	0	5


超时
Zoom Out
Last 3 hours
Refresh every 30m
Servername
All
dbname
All
超时查询
dev报表
使用说明
超时查询
count
70
60
50
40
30
20
10
0
14:20
14:30
14:40
14:50
15:00
15:10
15:20
15:30
15:40
15:50
16:00
16:10
16:20
16:30
16:40
16:50
17:00
17:10
total
148
35
25
25
21
21
19
16
logdatetime
servername
dbname
stmt
2017-07-16 16:15:23
c4dbsrv80\ucardb2
2017-07-16 16:15:23
2017-07-16 16:15:18
2017-07-16 16:15:18
SELECT a.carId FR
2017-07-16

邮件报警示例

复制延迟

[redacted]

收件人： dbas

agent_id：344
delay_minute：2882
delay_cmds：5906536

agent_id：408
delay_minute：2857
delay_cmds：1114846

agent_id：454
delay_minute：2885
delay_cmds：69744

作业错误(死锁牺牲作业周累计)@[redacted]

[redacted]

收件人： dbas

jobname：[redacted]
endtime：2017-07-09 13:32:25
duration：00:02:25
schedule：every 10 minute(s) between 0:00 and 23:59
message：已以用户 SCOM\ClusService 的身份执行。 事务(进程 ID 427)与另一个进程被死锁在 锁 资源上，并且已被选
(错误 1205). 该步骤失败。

jobname：[redacted]
endtime：2017-07-12 10:09:00
duration：00:28:00
schedule：every 7 minute(s) between 7:00 and 15:00
message：已以用户 SCOM\clusservice 的身份执行。 事务(进程 ID 514)与另一个进程被死锁在 锁 资源上，并且已被选
(错误 1205). 该步骤失败。

jobname：[redacted]
endtime：2017-07-12 10:09:00
duration：00:28:00
schedule：every 34 minute(s) between 15:00 and 18:00
message：已以用户 SCOM\clusservice 的身份执行。 事务(进程 ID 514)与另一个进程被死锁在 锁 资源上，并且已被选
(错误 1205). 该步骤失败。

发现长时间运行作业报警

长时间运行作业报警 [[redacted]]

收件人：张[redacted]; dbas

通知作业负责人
抄送DBA

此邮件已经归档。 查看原始项

服务器:C5[redacted]

作业名称:[redacted]
作业开始时间:04/02/2017 03:00:01
作业已执行时间:3小时38分钟37秒
作业历史平均执行时间:1小时52分钟56秒
作业描述:mailto:{[redacted]@bitauto.com}

largeIO@C[redacted]>>Ma[redacted]>>p_Get_MasterBandInSale, query_hash:0xD913529F78EB3FD6, 每秒io:278096

[redacted]

收件人： dbas

操作

此邮件已经归档。 查看原始项

```
SE[redacted]T DISTINCT m.* FROM MaiCheBase.dbo.CarMasterBrand_Full m WITH ( NOLOCK ) WHERE
m.M[redacted]MaiCheBase.dbo.CarSource s WITH (
NOLOCK ) [redacted] JOIN MaiCheBase [redacted] -inner
[redacted]dealerInfo [redacted].dealerId=d.dealerId INNER [redacted]Base.dbo.Deale[redacted]aleCity d
W[redacted]alerId = d.DealerId [redacted] = @cityId UNION SELECT
cc.M[redacted]erBrand[redacted]CK ) WHERE
cc.Busi[redacted]form )
```

5# 标准化

标准化参考

实例配置	最大并行度、并行开销阈值等
Windows配置	锁定内存页、执行卷维护任务、磁盘簇大小等
TempDB	个数和增长
数据库配置	数据大小和增长、兼容级别、日志延迟写、自动更新统计信息等
通用脚本	一些维护脚本、查看当前DB状态脚本、备份还原脚本等
扩展事件等	捕捉超时、捕捉慢查询、审核等
备份机制	根据RTO/RPO选择策略

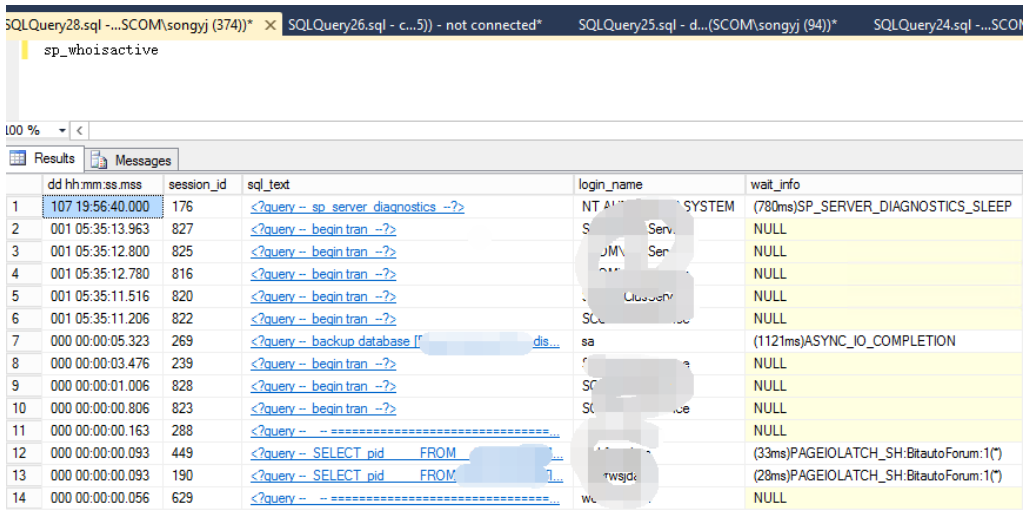
标准化优势

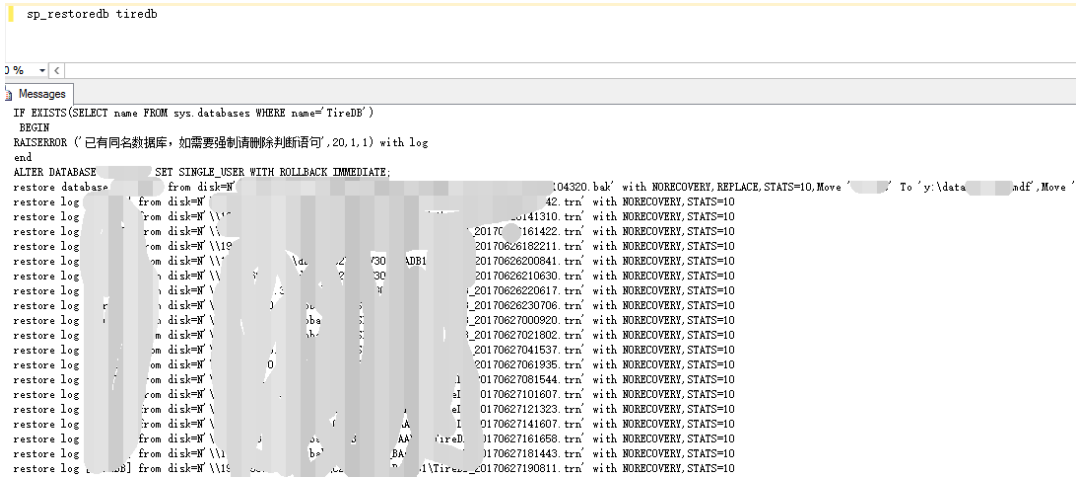
- 避免遗漏设置导致的问题
- 避免没必要的人工成本
- 降低运维水平需求，初级运维人员可以利用现有脚本完成高级功能

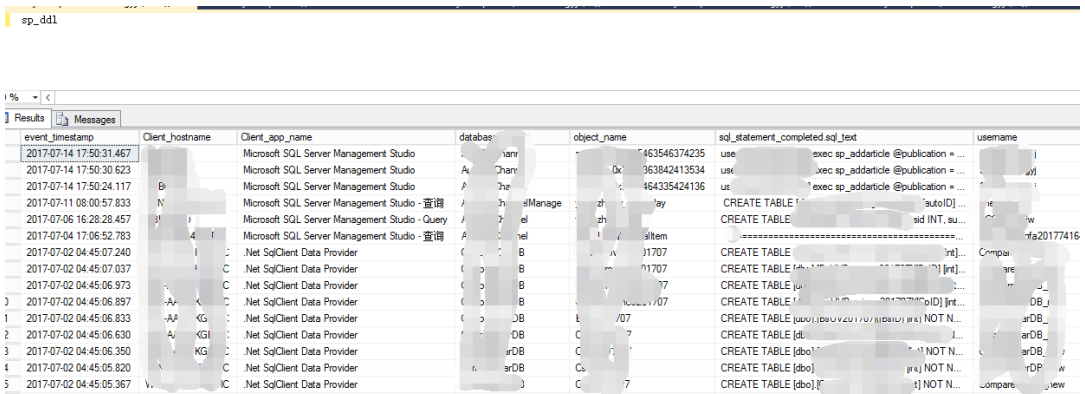
标准化实现机制

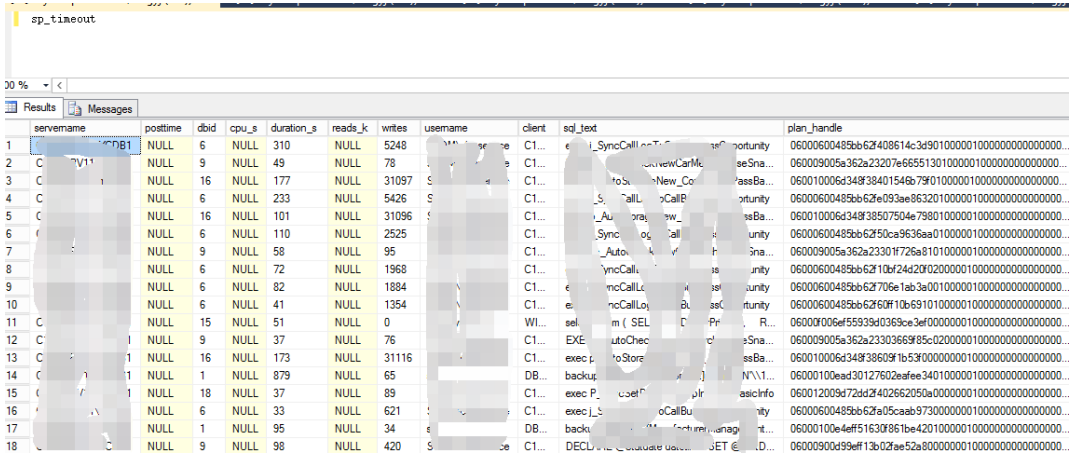
- 虚拟化使用模板完成标准化（Hyper-V/Vmware）
- 实体机使用初始化脚本（PowerShell/Python等）
- 统一部署脚本将新实例的相关元数据存入统一优化流程

标准化脚本示例


SQLQuery28.sql -...SCOM\songyj (374))*
SQLQuery26.sql - c...5)) - not connected*
SQLQuery25.sql - d...(SCOM\songyj (94))*
SQLQuery24.sql -...SCOM
sp_whoisactive
Results
Messages
dd hh:mm:ss.mss | session_id | sql_text | login_name | wait_info
1 | 107 19:56:40.000 | 176 | <?query -- sp_server_diagnostics --?> | NT AU...SYSTEM | (780ms)SP_SERVER_DIAGNOSTICS_SLEEP
2 | 001 05:35:13.963 | 827 | <?query -- begin tran --?> | S...Serv | NULL
3 | 001 05:35:12.800 | 825 | <?query -- begin tran --?> | ...DM\...Ser | NULL
4 | 001 05:35:12.780 | 816 | <?query -- begin tran --?> | ...OM... | NULL
5 | 001 05:35:11.516 | 820 | <?query -- begin tran --?> | ...ClassSer... | NULL
6 | 001 05:35:11.206 | 822 | <?query -- begin tran --?> | SC...ce | NULL
7 | 000 00:00:05.323 | 269 | <?query -- backup database [...dis... | sa | (1121ms)ASYNC_IO_COMPLETION
8 | 000 00:00:03.476 | 239 | <?query -- begin tran --?> | ...e | NULL
9 | 000 00:00:01.006 | 828 | <?query -- begin tran --?> | SC... | NULL
10 | 000 00:00:00.806 | 823 | <?query -- begin tran --?> | SC...ce | NULL
11 | 000 00:00:00.163 | 288 | <?query -- -- ==========...... | | NULL
12 | 000 00:00:00.093 | 449 | <?query -- SELECT pid FROM ...... | ... | (33ms)PAGEIOLATCH_SH:BitautoForum:1(")
13 | 000 00:00:00.093 | 190 | <?query -- SELECT pid FROM ...]... | rwsjd... | (28ms)PAGEIOLATCH_SH:BitautoForum:1(")
14 | 000 00:00:00.056 | 629 | <?query -- -- ==========...... | wo... | NULL


sp_restoredb tiredb
Messages
IF EXISTS(SELECT name FROM sys.databases WHERE name='TireDB')
BEGIN
RAISERROR ('已有同名数据库，如需要强制请删除判断语句',20,1,1) with log
end
ALTER DATABASE ... SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
restore database ... from disk=N'...04320.bak' with NORECOVERY,REPLACE,STATS=10,Move '...' To 'y:\data...mdf',Move '...
restore log ... from disk=N'...42.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\\...20141310.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\\...20170...161422.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\\19...20170626182211.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\\...ADB1...20170626200841.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\\...20170626210630.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170626220617.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170626230706.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627000920.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627021802.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627041537.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627061935.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627081544.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627101607.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627121323.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627141607.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627161658.trn' with NORECOVERY,STATS=10
restore log ... from disk=N'\...20170627181443.trn' with NORECOVERY,STATS=10
restore log ...] from disk=N'\...20170627190811.trn' with NORECOVERY,STATS=10


sp_ddl
Results
Messages
event_timestamp | Client_hostname | Client_app_name | database | object_name | sql_statement_completed.sql_text | username
2017-07-14 17:50:31.467 | ... | Microsoft SQL Server Management Studio | ... | ...463546374235 | use... exec sp_addarticle @publication = ... | ...
2017-07-14 17:50:30.623 | ... | Microsoft SQL Server Management Studio | ... | ...0x...363842413534 | use... exec sp_addarticle @publication = ... | ...
2017-07-14 17:50:24.117 | ... | Microsoft SQL Server Management Studio | ... | ...464335424136 | use... exec sp_addarticle @publication = ... | ...
2017-07-11 08:00:57.833 | ... | Microsoft SQL Server Management Studio - 查询 | ... | ... | CREATE TABLE ... [autoID] ... | ...
2017-07-06 16:28:28.457 | ... | Microsoft SQL Server Management Studio - Query | ... | ... | CREATE TABLE ... sid INT, su... | ...
2017-07-04 17:06:52.783 | ... | Microsoft SQL Server Management Studio - 查询 | ... | ... | ... | ...fa20177416
2017-07-02 04:45:07.240 | ... | .Net SqlClient Data Provider | ... | ...01707 | CREATE TABLE ...nt]... | ...
2017-07-02 04:45:07.037 | ... | .Net SqlClient Data Provider | ... | ...01707 | CREATE TABLE ...[int]... | ...
2017-07-02 04:45:06.973 | ... | .Net SqlClient Data Provider | ... | ...07 | CREATE TABLE ... | ...
2017-07-02 04:45:06.897 | ... | .Net SqlClient Data Provider | ... | ...1707 | CREATE TABLE ...pID] [int]... | ...
2017-07-02 04:45:06.833 | ... | .Net SqlClient Data Provider | ... | .../07 | CREATE TABLE ...[int] NOT N... | ...
2017-07-02 04:45:06.630 | ... | .Net SqlClient Data Provider | ... | ...7 | CREATE TABLE [d... | ...
2017-07-02 04:45:06.350 | ... | .Net SqlClient Data Provider | ... | ... | CREATE TABLE [dbo]... NOT N... | ...
2017-07-02 04:45:05.820 | ... | .Net SqlClient Data Provider | ... | ... | CREATE TABLE [dbo]... NOT N... | ...
2017-07-02 04:45:05.367 | ... | .Net SqlClient Data Provider | ... | ...7 | CREATE TABLE [dbo]...t] NOT N... | ...


sp_timeout
Results
Messages
servername | posttime | dbid | cpu_s | duration_s | reads_k | writes | username | client | sql_text | plan_handle
1 | ...DB1 | NULL | 6 | NULL | 310 | NULL | 5248 | ... | C1... | ...Sync...Call...portunity | 0600060048 5bb62f408614c3d901000001000000000000000...
2 | C...7V11 | NULL | 9 | NULL | 49 | NULL | 78 | ... | C1... | ...seSna... | 060009005a362a23207e665513010000010000000000000...
3 | C... | NULL | 16 | NULL | 177 | NULL | 31097 | S... | C1... | ...New_Co...assBa... | 060010006d348f38401546b79f0100000100000000000000...
4 | C... | NULL | 6 | NULL | 233 | NULL | 5426 | S... | C1... | ...portunity | 0600060048 5bb62fe093ae863201000001000000000000000...
5 | C... | NULL | 16 | NULL | 101 | NULL | 31096 | S... | C1... | ...ssBa... | 060010006d348f38507504e798010000010000000000000...
6 | ... | NULL | 6 | NULL | 110 | NULL | 2525 | ... | C1... | ...unity | 0600060048 5bb62f50ca9636aa0100000100000000000000...
7 | ... | NULL | 9 | NULL | 58 | NULL | 95 | ... | C1... | ...Sna... | 060009005a362a23301f726a8101000001000000000000000...
8 | ... | NULL | 6 | NULL | 72 | NULL | 1968 | ... | C1... | ...unity | 0600060048 5bb62f10bf24d20f0200000100000000000000...
9 | ... | NULL | 6 | NULL | 82 | NULL | 1884 | ... | C1... | ...unity | 0600060048 5bb62f706e1ab3a0010000010000000000000...
10 | ... | NULL | 6 | NULL | 41 | NULL | 1354 | ... | C1... | ...ortunity | 0600060048 5bb62f60ff10b691010000010000000000000...
11 | C... | NULL | 15 | NULL | 51 | NULL | 0 | ... | WI... | sel...m (SEL...Pr...R... | 06000f006ef55939d0369ce3ef0000000100000000000000...
12 | C... | NULL | 9 | NULL | 37 | NULL | 76 | ... | C1... | EXE...utoChec...eSna... | 060009005a362a23303669f85c0200000100000000000000...
13 | C... | NULL | 16 | NULL | 173 | NULL | 31116 | ... | C1... | exec p...oStora...ssBa... | 060010006d348f38609f1b53f0000000010000000000000...
14 | C... | NULL | 1 | NULL | 879 | NULL | 65 | ... | DB... | backup... | 06000100ead30127602eafee3401000001000000000000000...
15 | C... | NULL | 18 | NULL | 37 | NULL | 89 | ... | C1... | exec P...oInf...asicInfo | 060012009d72dd2f402662050a0000000100000000000000...
16 | ... | NULL | 6 | NULL | 33 | NULL | 621 | ... | C1... | exec j...oCallBu...nity | 0600060048 5bb62fa05caab9730000000100000000000000...
17 | ... | NULL | 1 | NULL | 95 | NULL | 34 | ... | DB... | backu...facture...nage...nt... | 06000100e4eff51630f861be42010000010000000000000...
18 | C... | NULL | 9 | NULL | 98 | NULL | 420 | S...ce | C1... | DECL...ET @...D... | 06000900d99eff13b02fae52a8000000010000000000000...

6# 还原策略

还原策略

- 注意到我说的不是备份策略
- RTO（Recovery Time Object）
- RPO（Recovery Point Object）

备份类型

- 完整备份
- 增量备份
  - 差异备份（基于变更数据，还原快，无法还原到指定时间点）
  - 日志备份（基于）
- 当RTO和RPO要求很高的时候，就不能再基于冷备

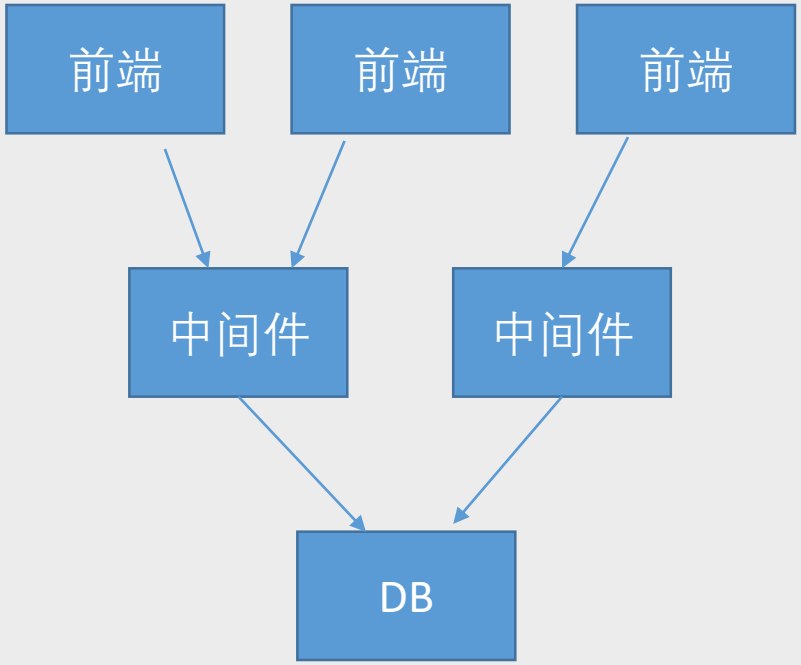
其他原则

- 数据库之间弱耦合
  - 避免数据库之间直接进行数据同步
  - 依赖中间层
- 业务无关数据库尽量拆分到不同机器
  - 单台大机器N个数据库 vs 单大宿主机多个虚拟机
- 按照业务要求，定期数据归档
- 尽量微服务，一个业务实体不应该在数据库中调用另一个业务实体

7# SQL书写原则

书写原则

- 简单
- DB做简单的CRUD
- 业务操作在DB之外



SQL 书写原则

General Guideline

参数化

数据类型统一，尽量避免对列做操作

避免使用任何ORM系统

简单、简单、简单，重要的事情说三遍

防止SQL注入

高并发下防止重编译（大量降低CPU）

SQL 书写原则

General Guideline

参数化

数据类型统一，尽量避免对列做操作

避免使用任何ORM系统

简单、简单、简单，重要的事情说三遍

数据类型转换可能导致低效的执行计划

列上数据类型转换导致无法使用Seek

数据类型不一致导致隐式转换

SQL 书写原则

General Guideline

参数化

数据类型统一，尽量避免对列做操作

避免使用任何ORM系统

简单、简单、简单，重要的事情说三遍

ORM对于跨数据库开发效率高

ORM往往导致复杂低效的SQL

SQL 书写原则

General Guideline

参数化

数据类型统一，尽量避免对列做操作

避免使用任何ORM系统

简单、简单、简单，重要的事情说三遍

任何复杂SQL都应该分解为简单SQL

尽量避免相关子查询

存储过程中有逻辑分支是灾难

多表Join将中间结果存入带索引的临时表

如果可能，使用微服务架构，不同实体的数据不应该在DB层面有关联

写在最后

- 尽量工程化、标准化主动预防问题
- 在问题出现征兆之前，主动解决
- 能够避免很多午夜Surprise

Q A