

Windows Azure存储

演讲者
职位
公司



议程



Windows Azure 存储

Blob 存储

驱动器

表

队列

Windows Azure 存储

在云中的存储

高扩展，高可靠和高可用性
在任何地方任何时间访问
只为使用付费

通过RESTful的Web Service

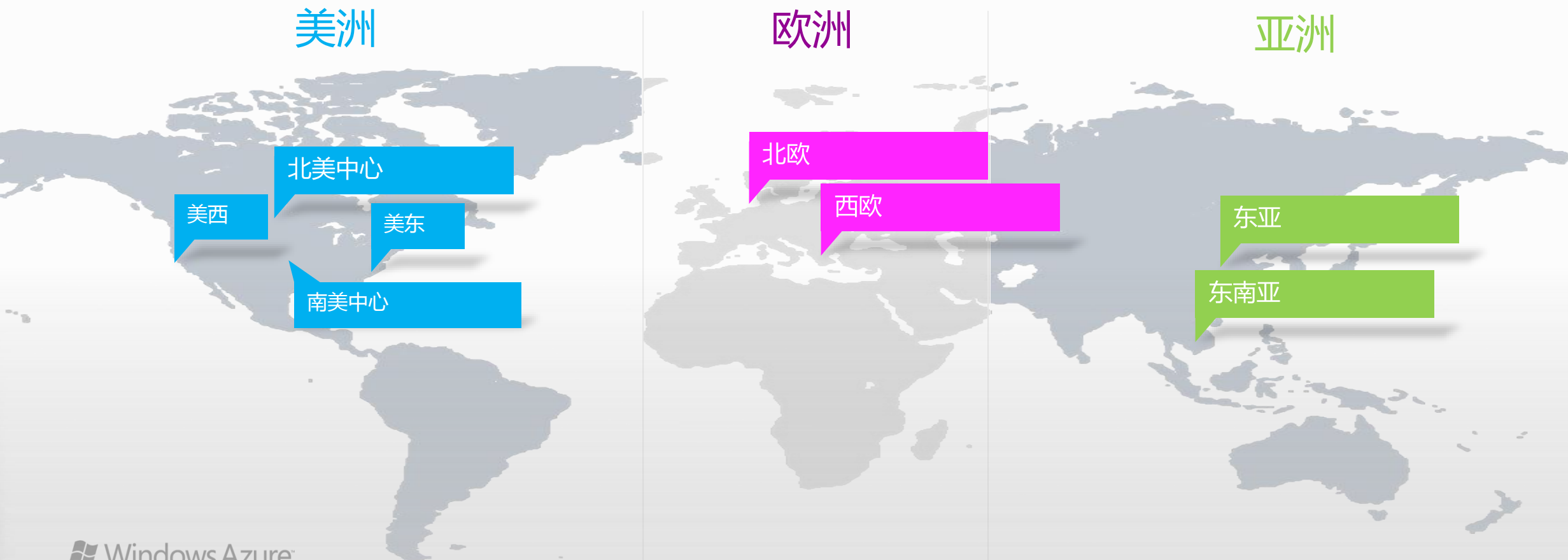
从 Windows Azure Compute 使用
从互联网上的任何地点使用



Windows Azure 存储账户

使用特殊的全球唯一账户名

可以选择地理位置来存储



Windows Azure 存储账户

支持CDN的账户

通过24个全球的CDN节点分发Blobs

通过计算账户共同定位存储账户

明确定义或者使用地缘组

账户拥有2个独立的512bit共享密钥

每个账户100 TBs

新的特性



地理复制 存储分析

日志: 提供存储账户请求的跟踪信息
数据: 提供关键能力和关键统计信息
Blobs, Tables, and Queues

对Blob改进的HTTP头信息

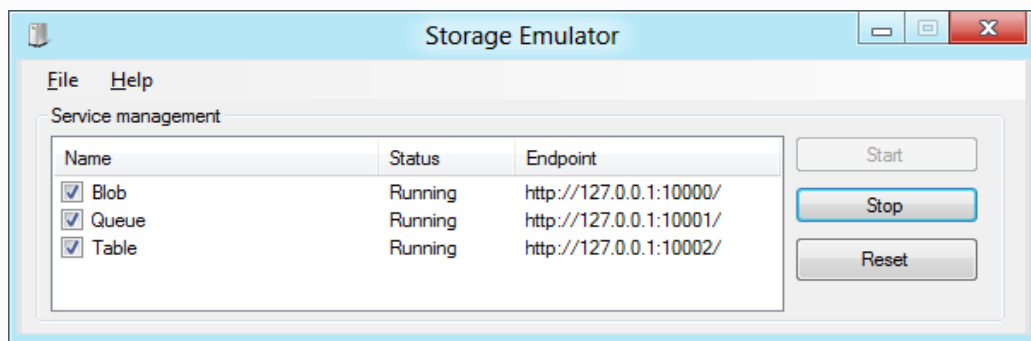
在Development Fabric中的存储

提供本地“Mock” 存储

模拟在云中的存储

允许离线开发

需要Express 2005/2008 或更新



云和模拟器存储有一些区别:

<http://msdn.microsoft.com/en-us/gg433135>

对开发人员一个好的方法:

测试部署前期的代码，将存储先放到云上
使用Dev Fabric连接到云上的存储
最后把计算放到云上

存储客户端API

本文中会包括底层的RESTful API

能够通过任何HTTP客户端调用
e.g. Flash, Silverlight, etc...

SDK中的Client API 、
Microsoft.WindowsAzure.StorageClient

提供一个封装REST service的接口

在多种语言中的存储的库

C#/.NET

Python

Ruby

Perl

JavaScript (Node)

Java

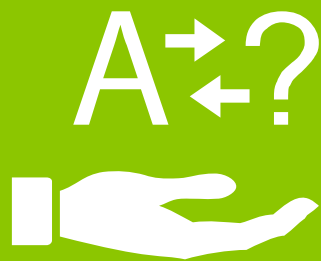
PHP

Erlang

Common LISP

Objective-C

C#/VB on Windows Phone 7



存储安全性

Windows Azure 存储提供了简单的安全性来调用存储服务

HTTPS 端口

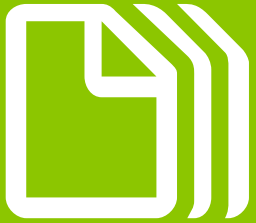
对于授权的操作需要数字签名请求

每个账户有两个 512bit 对称密钥

能够独立生成

通过共享的访问签名实现更多颗粒的安全性

Windows Azure Storage 抽象



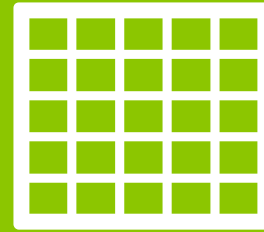
Blobs

简单命名的文件和元数据



Drives

可靠的NTFS 盘，基于 Blobs.



Tables

结构化的存储。一张表是实体的集合，一个实体是属性的集合。



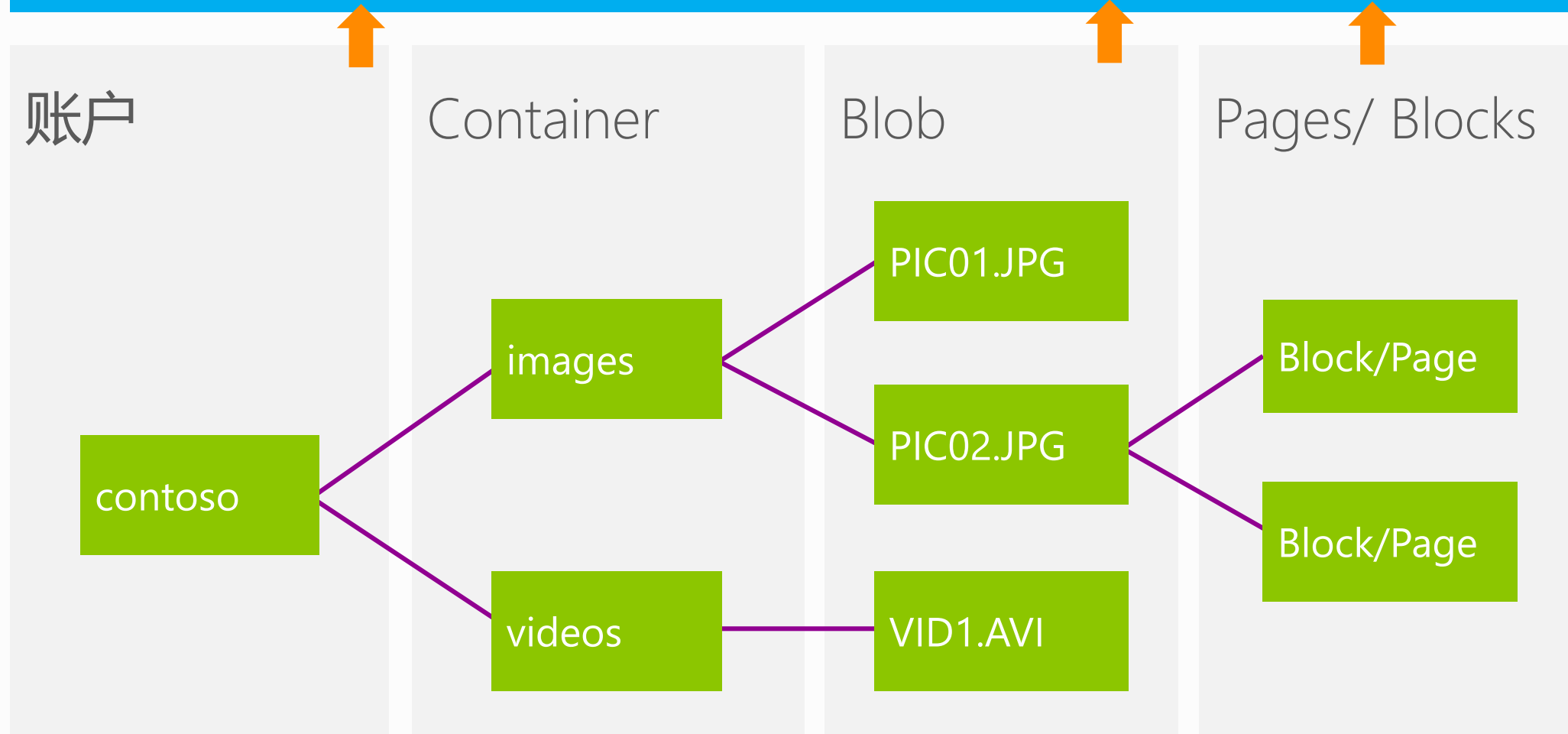
Queues

可靠的消息存储和发送

Blob 存储

Blob Storage 概念

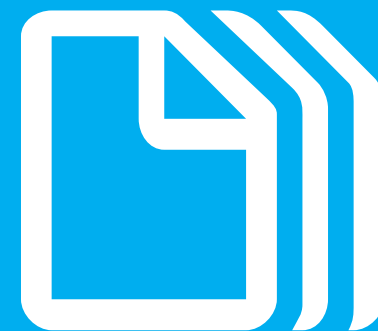
`http://<account>.blob.core.windows.net/<container>/<blobname>`



Blob 细节

主要 Web Service
操作

PutBlob
GetBlob
DeleteBlob
CopyBlob
SnapshotBlob
LeaseBlob



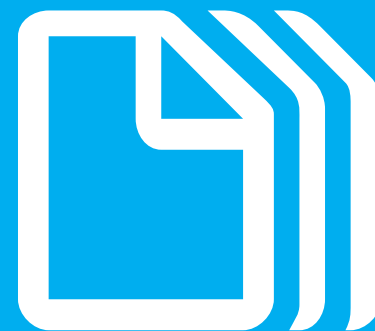
Blob 细节

相关的 Blob 元数据

标准的HTTP元数据/头
(缓存控制, 内容编码, 内容类型等)

源数据是<name, value> 对,
每个Blob最多8KB

要么做为PubBlob的一部分,
或者独立

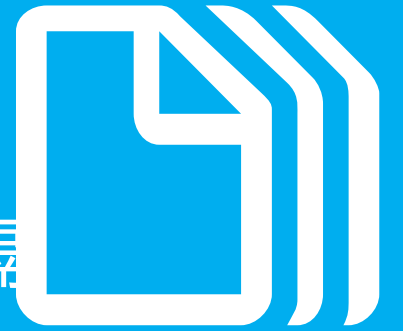


Blob 细节

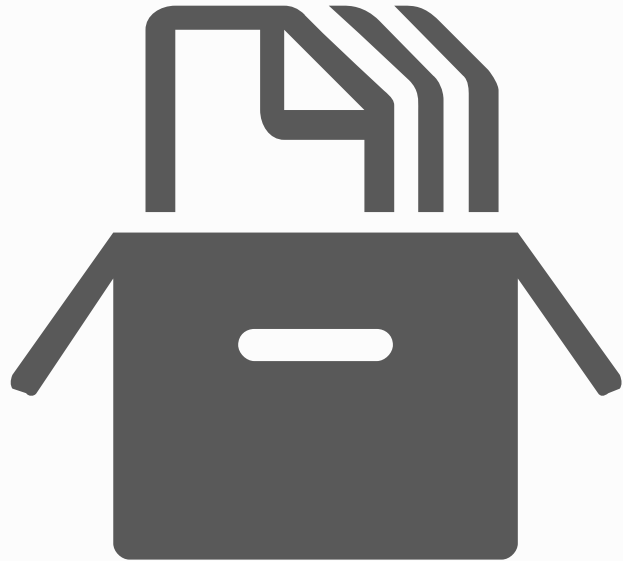
Blob 总是
通过名字
访问

名字可以包括 '/' 或者其他分隔符

e.g. /<container>/myblobs/blob.jpg



Blob Containers



每个账户多个Containers

特殊的\$root container

Blob Container

一个container包含blob的集合
在container级别设置访问策略
和Container相关的元数据

列出container里面所有的blob

包括Blob Metadata 和MD5

没有查询. i.e. 没有WHERE MetadataValue = ?

Blobs 带宽

一个分区

目标是60MB/s 每个Blob

遍历Blobs

获取Blob的操作需要参数

前缀
分隔符
包括= (snapshots, metadata etc...)

```
http://adventureworks.blob.core.windows.net/  
Products/Bikes/SuperDuperCycle.jpg  
Products/Bikes/FastBike.jpg  
Products/Canoes/Whitewater.jpg  
Products/Canoes/Flatwater.jpg  
Products/Canoes/Hybrid.jpg  
Products/Tents/PalaceTent.jpg  
Products/Tents/ShedTent.jpg
```

分页

Blob的列表可以分页

要么设置maxresults 或者;
超过默认值maxresults (5000)

```
http://.../products?comp=list&prefix=Canoes&max  
results=2
```

```
<Blob>Canoes/Whitewater.jpg</Blob>  
<Blob>Canoes/Flatwater.jpg</Blob>  
<NextMarker>MarkerValue</NextMarker>
```

Blob服务之旅

演示



两种类型的Blob

块Blob

用于流访问

每个blob由一个块的序列组成

每个块通过块Id区分

每个blob最大200GB

通过Etags优化并发

页Blob

用于随机读写访问

每个blob由页的数组组成

每个页通过offset来区分

每个blob最大 1TB

通过租用 (leases)优化

上传 Block Blob

上传一个大的blob



好处

有效延续和重试
并行，顺序无关

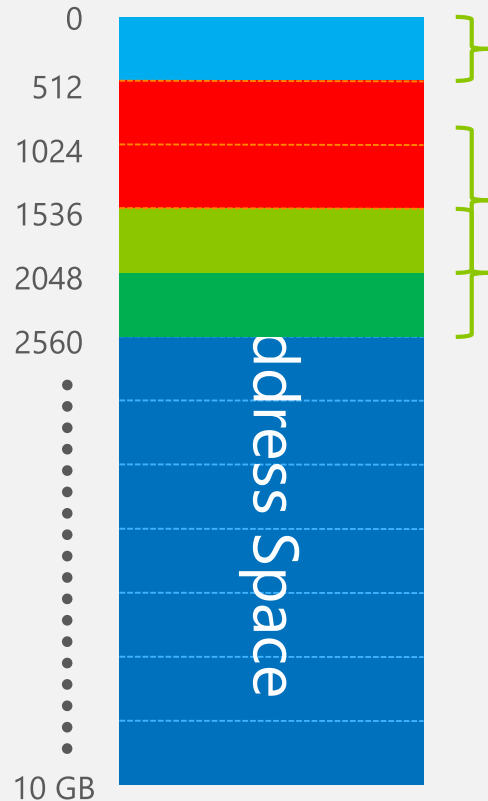
THE BLOB

```
blobName = "TheBlob.wmv";  
PutBlock(blobName, blockId1, block1Bits);  
PutBlock(blobName, blockId2, block2Bits);  
.....  
PutBlock(blobName, blockIdN, blockNBits);  
PutBlockList(blobName,  
              blockId1,...,blockIdN);
```

TheBlob.wmv

Windows Azure
Storage

页Blob – 随机读/写



创建MyBlob

定义Blob 大小= 10 Gbytes

稀疏存储 - 只为存储的页付费

固定页大小 = 512 bytes

随机访问操作

PutPage[512, 2048)

PutPage[0, 1024)

ClearPage[512, 1536)

PutPage[2048, 2560)

GetPageRange[0, 4096) 返回有效数据范围:

[0, 512) , [1536, 2560)

GetBlob[1000, 2048) 返回

All 0 for first 536 bytes

Next 512 bytes are data stored in [1536, 2048)

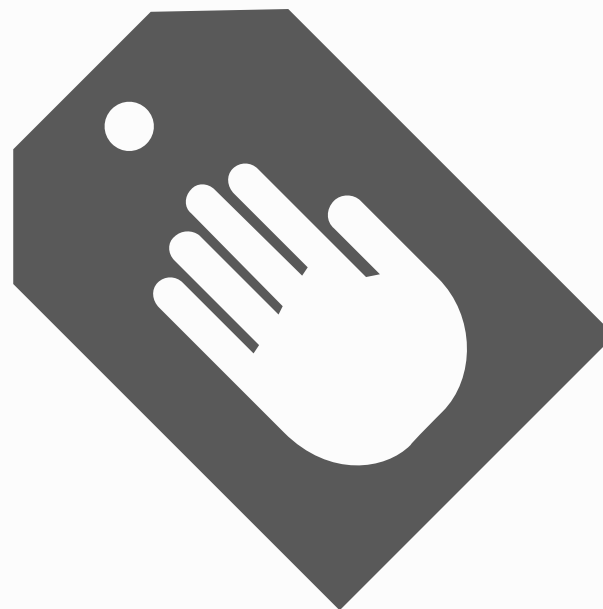
共享访问签名

良好的访问权限 blobs 和 containers
和存储Key一起签名的URL— 允许提升权限废止

使用短时间和重新提交
使用可删除的container级别的策略

2种方法

Ad-hoc
基于策略




Ad Hoc 签名

创建短时间的共享访问签名

Signed resource Blob or Container
AccessPolicy Start, Expiry and Permissions
Signature HMAC-SHA256 of above fields

用例

Single use URLs
E.g. 提供一个URL给Silverlight 客户端来上传文件



http://...blob.../pics/image.png?
sr=c&st=2009-02-09T08:20Z&se=2009-02-10T08:30Z&sp=w
&sig= dD80ihBh5jfNpymO5Hg1ldiJIEvHcJpCMiCMnN%2fRnbl%3d

基于策略的签名

创建Container级别的策略

定义StartTime, ExpiryTime, Permissions

创建共享访问签名URL

Signedresource Blob or Container

Signedidentifier Optional pointer to container policy

Signature HMAC-SHA256 of above fields

用例

提供可撤销的权限给特殊的用户/组

撤销: 删除或更新container策略



The diagram shows a signed URL with three blue arrows pointing to its parts: one to the resource path, one to the container policy, and one to the signature.

```
http://...blob.../pics/image.jpg?  
sr=c&si=MyUploadPolicyForUserID12345  
&sig=dD80ihBh5jfNpymO5Hg1ldiJIEvHcJpCMiCMnN%2fRnbl%3d
```

内容分发网络(CDN)

高带宽的全球blob内容分发

24 全球位置 (US, Europe, Asia, Australia and South America), 还在增加
无论用户在地球的什么地方, 无论存储账户在哪里, 体验完全相同

Blob service URL vs. CDN URL:

Windows Azure Blob URL: <http://images.blob.core.windows.net/>

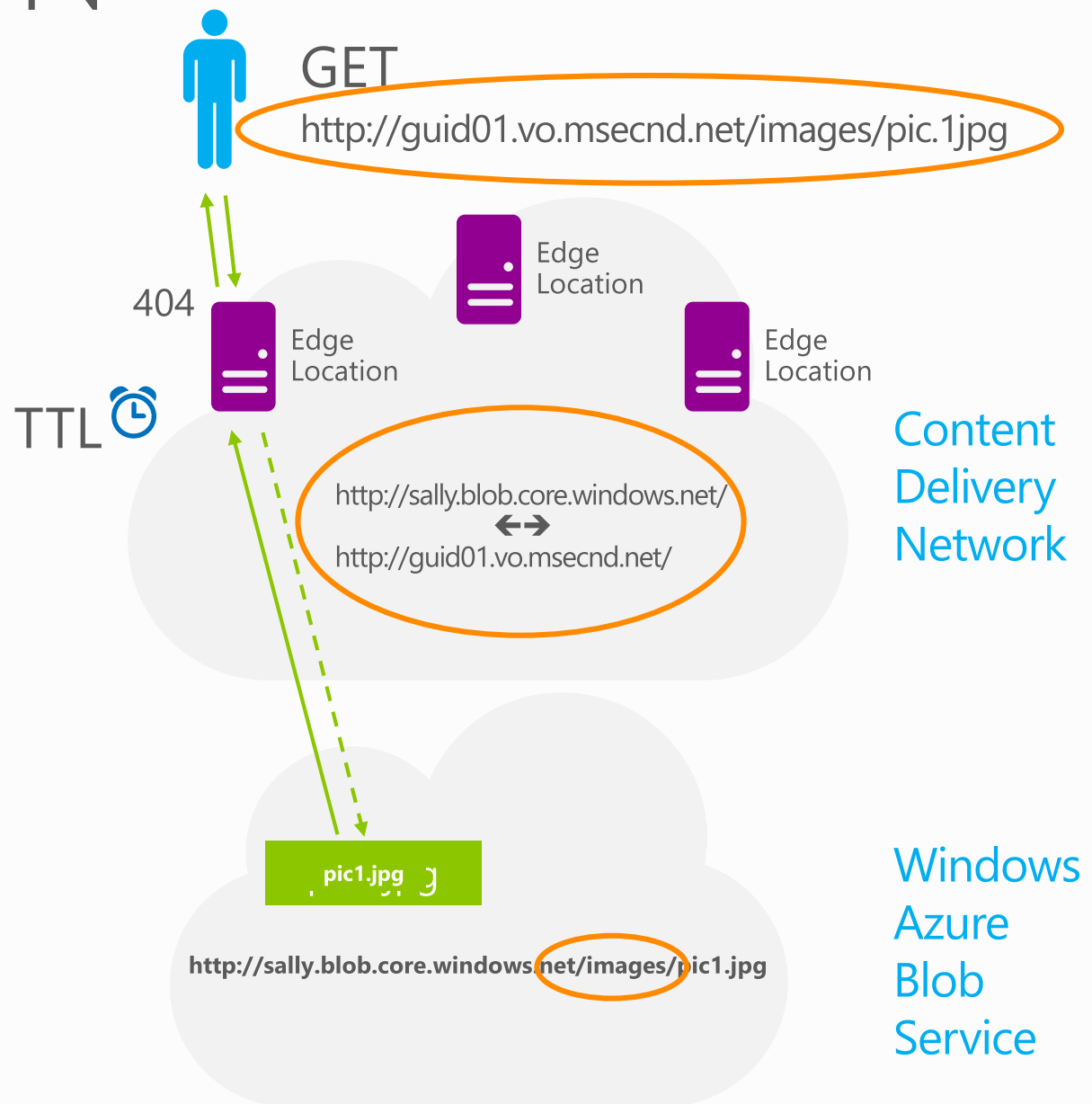
Windows Azure CDN URL: <http://<id>.vo.msecnd.net/>

Custom Domain Name for CDN: <http://cdn.contoso.com/>

Windows Azure CDN

开始CDN:

通过Dev Portal注册CDN
将Container设置为公共



驱动器



Windows Azure 驱动器

可靠的NTFS盘

使用现有的NTFS API访问网络挂载的驱动器
使用.NET 的 System.IO

好处

使用NTFS更容易地将现有的应用迁移到云上
实例回收，数据仍然存在不丢失

Windows Azure盘是一个NTFS的VHD Page Blob。

通过网络把Page Blob映射为一个NTFS驱动器
对于读操作有本地的缓存
所有的没有缓存的写都写到Page Blob中。

Windows Azure 驱动器功能

一个实例能最多动态加载16个驱动器
通过标准的BlobUI远程访问

不能远程挂载驱动器

能够把VHD上传到Page Blob，然后挂载为驱动器

能够把VHD下载到本地挂载

同时只能有一个实例可以读/写

使用只读的快照给多个实例读

驱动器细节

通过驱动器API，不是REST调用来操作驱动器

创建驱动器

在Blob存储中创建一个新的NTFS格式化的VHD

MountDrive/UnmountDrive

把VHD挂载到一个新的实例上，变成一个新的盘符

把VHD卸载

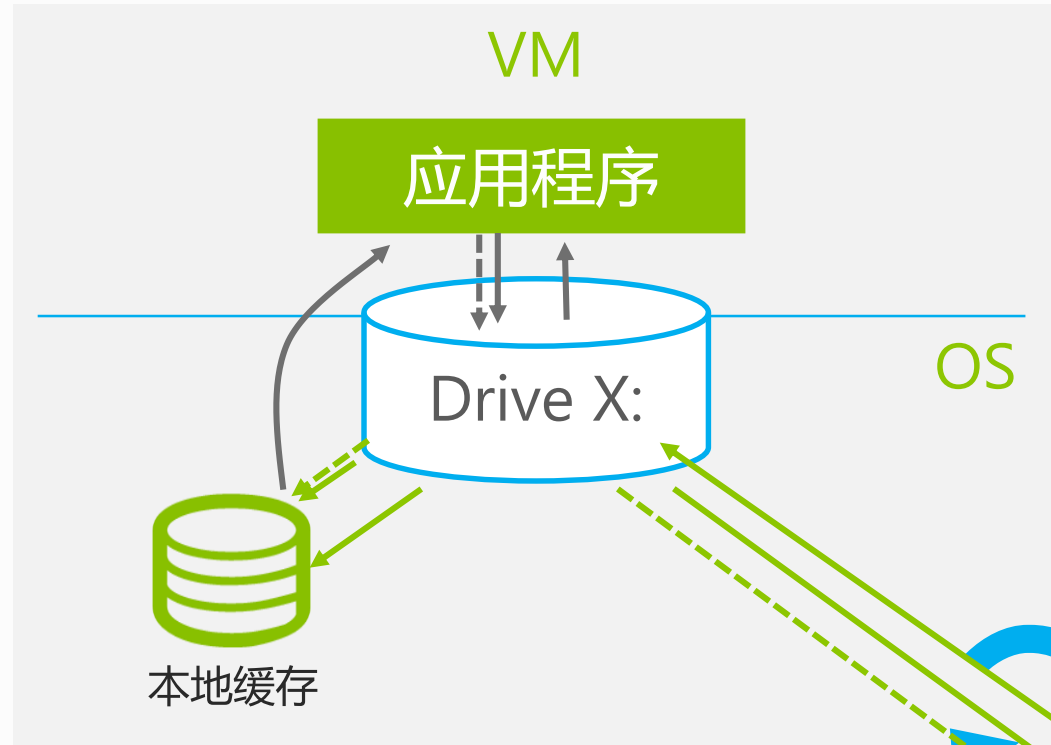
获取挂在的驱动器

列出所有挂载的驱动器，相关的blob和盘符

Drive镜像

创建盘符的镜像

Windows Azure 驱动器如何工作



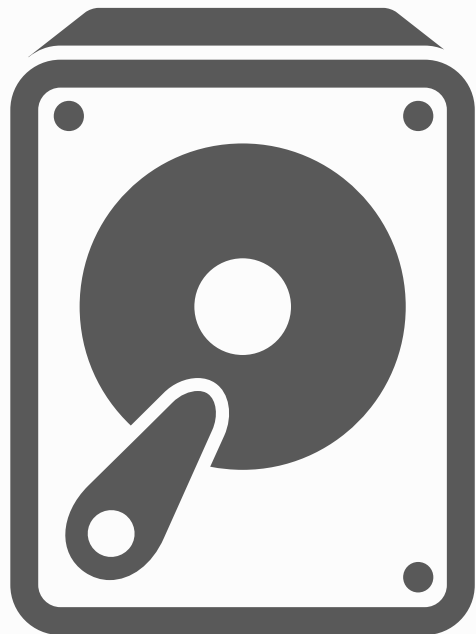
驱动器是一个在blob服务中格式化好的page blob
挂载包含了blob 租用
挂载定义了本地存储的数量作为缓存
NTFS 刷新/无缓存的写操作在返回应用之前提交到blob存储
NTFS 读可以来自于本地缓存或者blob存储（缓存失效）

Windows Azure
Blob 服务

Cloud Drive Client Library 举例

- `CloudStorageAccount`
`CloudStorageAccount.FromConfigurationSetting("CloudStorageAccount");`
- `//Initialize the local cache for drives mounted by this role instance`
`CloudDrive.InitializeCache(localCacheDir, cacheSizeInMB);`
- `//Create a cloud drive (PageBlob)`
`CloudDrive drive = account.CreateCloudDrive(pageBlobUri);`
`drive.Create(1000 /* sizeInMB */);`
- `//Mount the network attached drive on the local file system`
`string pathOnLocalFS = drive.Mount(cacheSizeInMB, DriveMountOptions.None);`
- `//Use NTFS APIs to Read/Write files to drive`
- `...`
- `//Snapshot drive while mounted to create backups`
`Uri snapshotUri = drive.Snapshot();`
- `//Unmount the drive`
`drive.Unmount();`

Drives故障转移



必须执行NTFS Flush 命令保持数据

使用System.IO.Stream.Flush()

使用租用保护读/写

1分钟租用过期

通过Windows Azure OS驱动维护

在RoleEntryPoint.OnStop卸载

失败时

租用会在1分钟后过期

在新的实例上重新mount

表



表存储概念

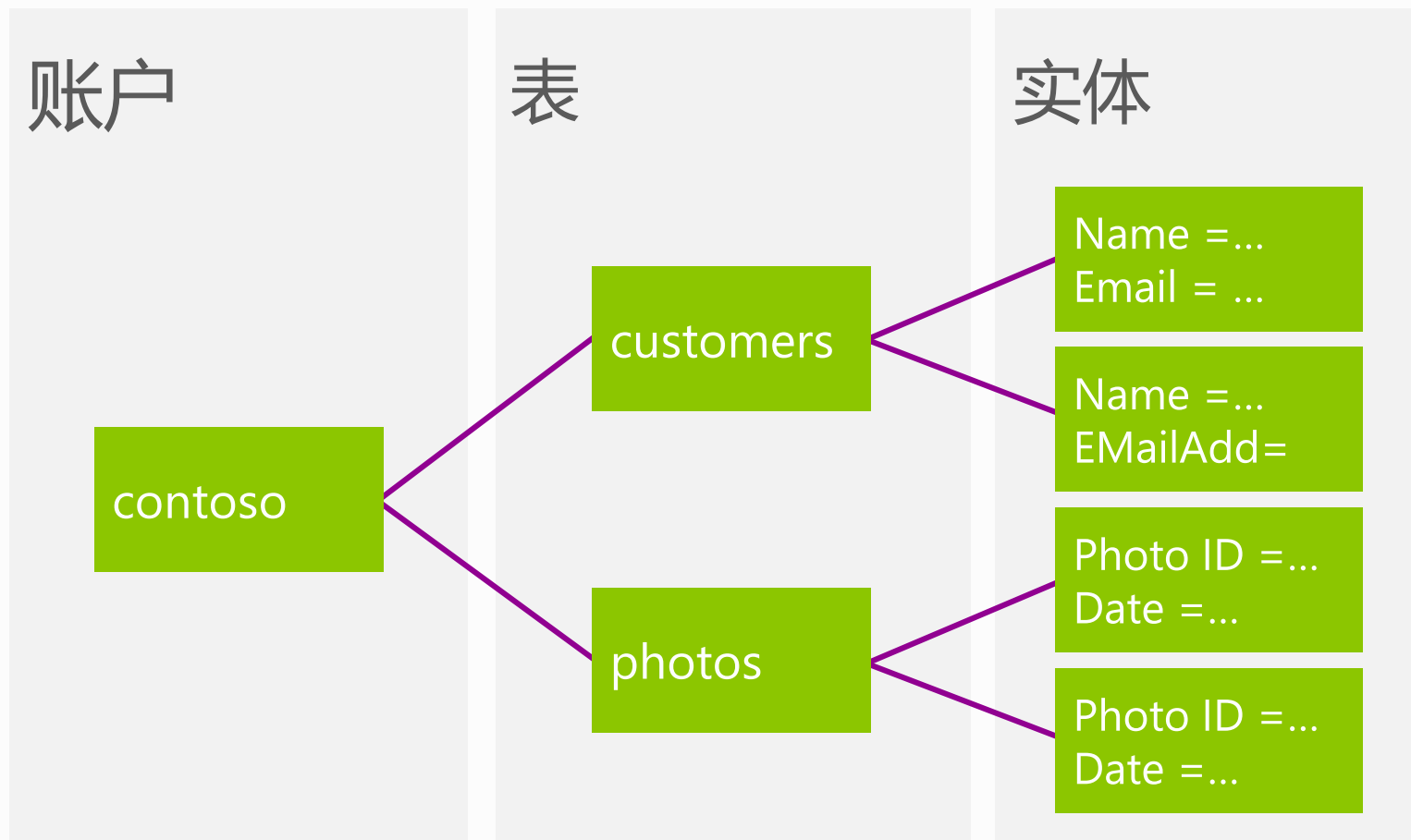


Table 细节



不是RDBMS!
Table

Create, Query, Delete
表可以有元数据



实体

Insert

Update

Merge – 部分更新

Replace – 更新全部实体

Upsert

Delete

Query

Entity Group Transaction

多个CUD 操作在一个原子事务中

Entity 属性

实体可以拥有最多255个属性

每个实体1MB

对每个实体的强制属性

PartitionKey & RowKey (只有索引的属性)

唯一实体区分
定义排序顺序

时间戳
优化并发
作为Etag暴露

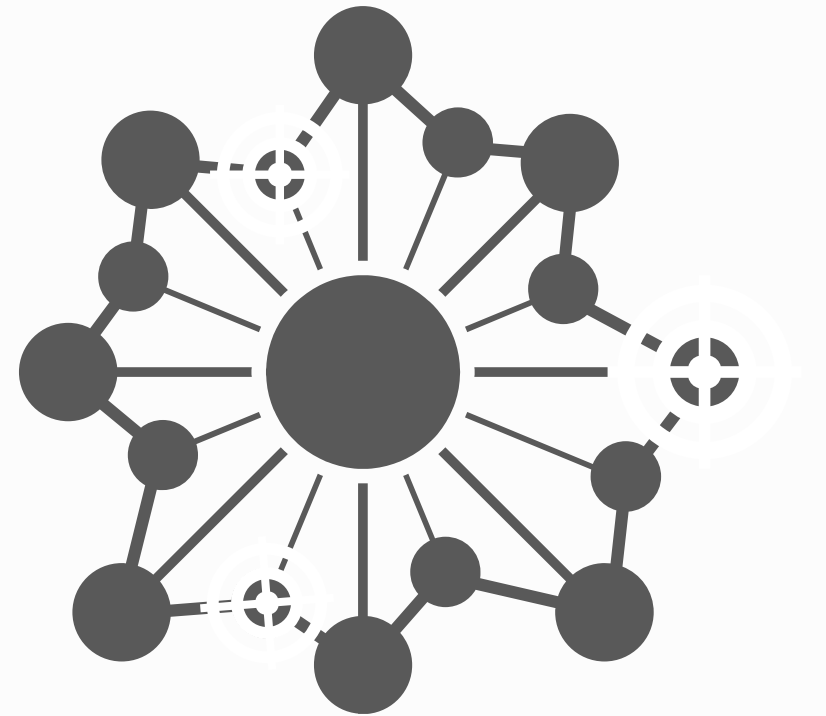
对其他属性没有固定的schema

每个属性存为 <name, typed value> 对

没有表的schema

属性可以是标准 .NET 类型

String, binary, bool, DateTime, GUID, int, int64, and double



没有固定Schema



FIRST	LAST	BIRTHDATE	FAV SPORT
Wade	Wegner	2/2/1981	Canoeing
Nathan	Totten	3/15/1965	
Nick	Harris	May 1, 1976	

查询

?\$filter=Last eq 'Wegner'

	FIRST	LAST	BIRTHDATE
	Wade	Wegner	2/2/1981
	Nathan	Totten	3/15/1965
	Nick	Harris	May 1, 1976

PartitionKey的目的

本地实体

相同partition的实体会存储在一起

有效地查询和本地缓存

所有查询中包含partition key

Entity Group Transactions

相同Partition中的多个Insert/Update/Delete操作在一个事务中

Table 扩展性

目标流量 – 500 tps/partition, 几千个tps/account

Windows Azure 监控分区的模式

自动负载均衡

每个分区能在不同的存储节点上操作

满足表的扩展需求

分区和分区范围

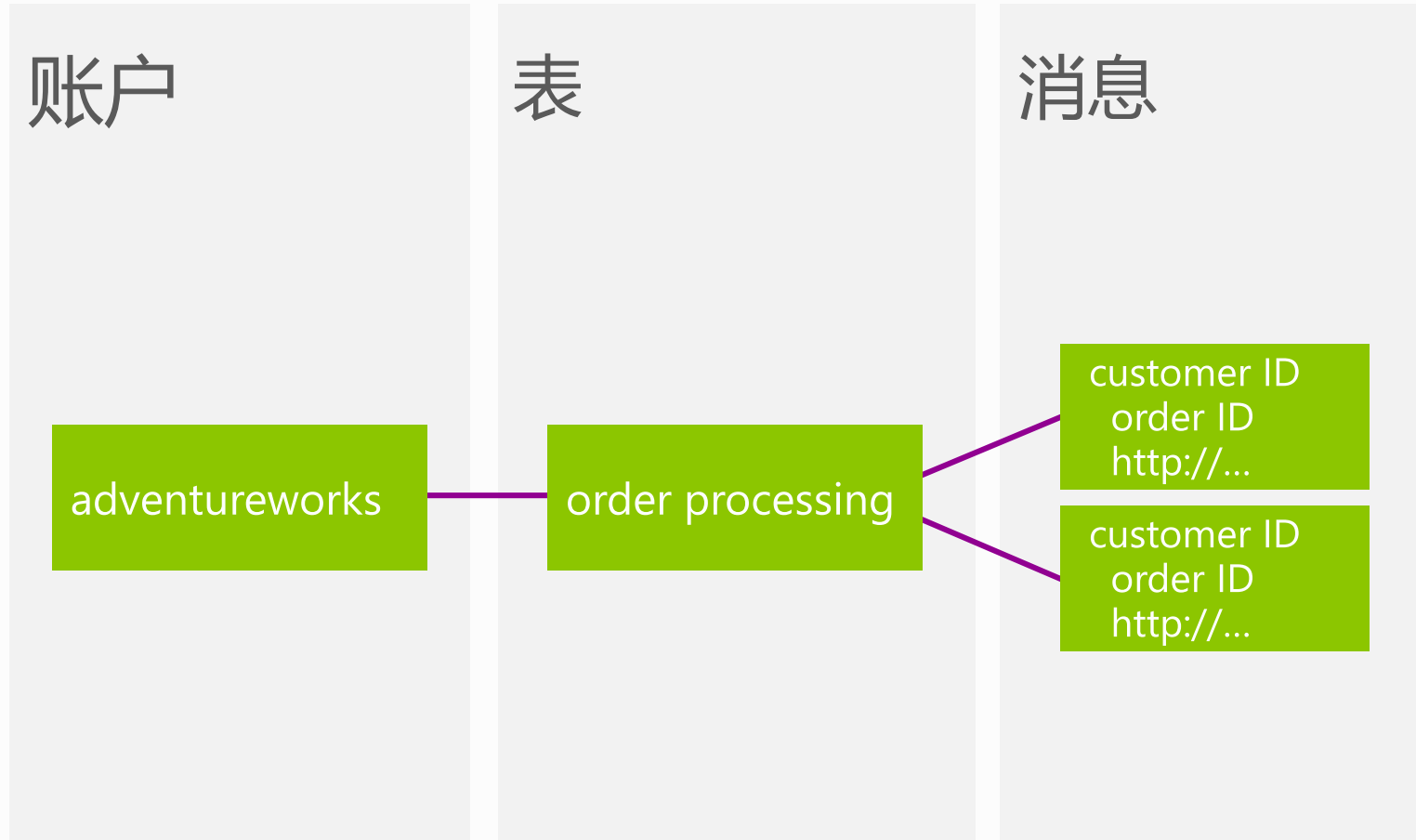


PARTITIONKEY (分类)	ROWKEY (标题)	时间戳	模型年份
Bikes	Super Duper Cycle	...	2009
Bikes	Quick Cycle 200 Deluxe	...	2007
...
Canoes	Whitewater	...	2009
Canoes	Flatwater	...	2006
Rafts	14ft Super Tourer	...	1999
...
Skis	Fabrikam Back Trackers	...	2009
...
Tents	Super Palace	...	2008
...
Tents	Super Palace	...	2008

队列



Queue Storage 概念



利用队列实现松耦合的工作流

在角色之间完成工作流

从队列中载入工作

生产者可以在放入队列后忘记它的存在

多个工作者可以消费队列

极端流量(> 500 tps)

使用多个队列

批读取消息

每个消息多个工作



队列细节

简单的异步分发队列

不限制队列长度

每个消息64kb

ListQueues - 列出账户下所有的队列

队列操作

CreateQueue

DeleteQueue

Get/Set Metadata

Clear Messages

队列细节

消息操作

PutMessage – 写入消息

GetMessages – 读取一条或多条消息并且隐藏他们

PeekMessages – 读取一条或多条消息但不隐藏他们

DeleteMessage – 永久从队列删除一条消息

UpdateMessage – 客户端更新租用和内容

队列的可靠分发

保证分发/处理消息（2步处理）

工作者读取消息并且标记为不可见，在一定的“不可见时间”
工作者在处理完成后删除消息
如果工作角色崩溃，消息重新变成可见

Windows Azure 存储总结

创建应用的数据抽象层的基础

Blobs: 文件和大对象

驱动器: 迁移应用程序中使用NTFS APIs

表: 大块需要扩展的结构化数据

队列: 可靠的消息分发

通过库简单使用





© 2012 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Translated to Chinese Simplified Version by Shanghai Yungoal Info Tech Co., Ltd. [YunGoal](#)