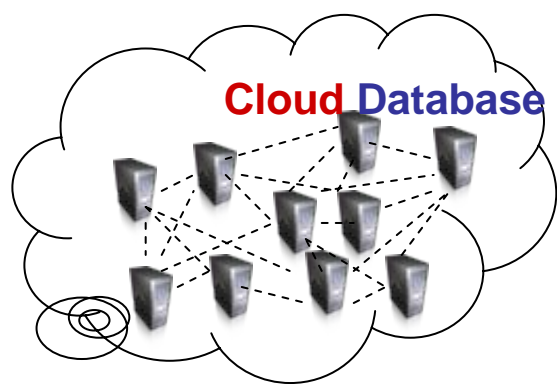


# 计算机领域最新技术报告



## 云数据库

林子雨

厦门大学计算机科学系

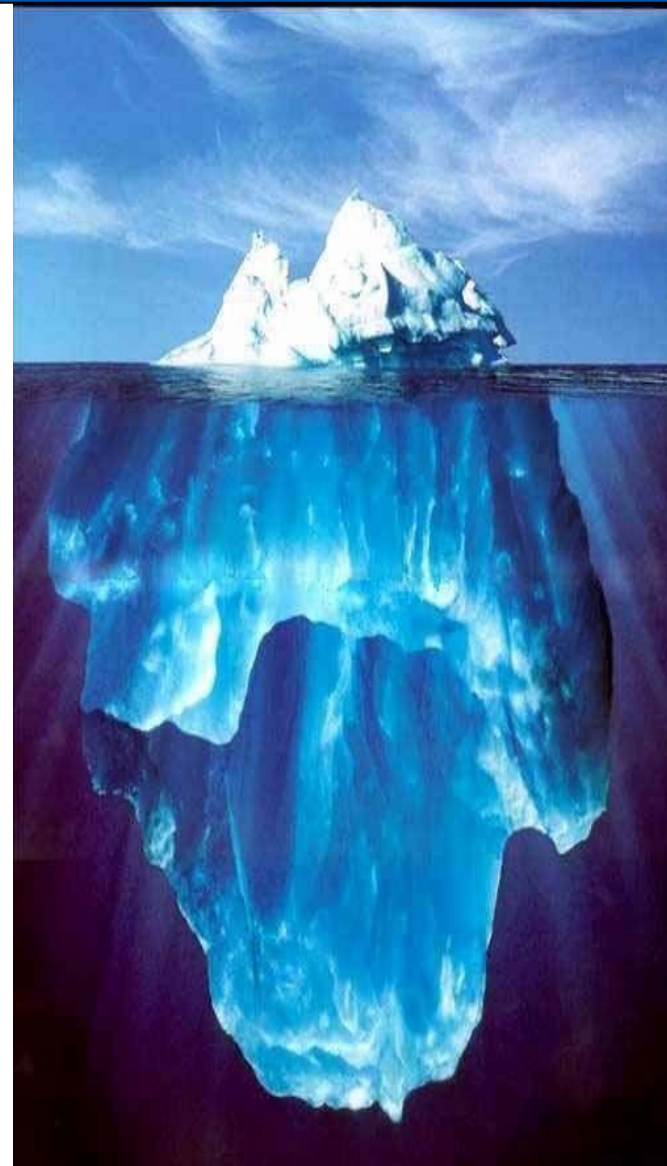
E-mail: [ziyulin@xmu.edu.cn](mailto:ziyulin@xmu.edu.cn) ▶▶





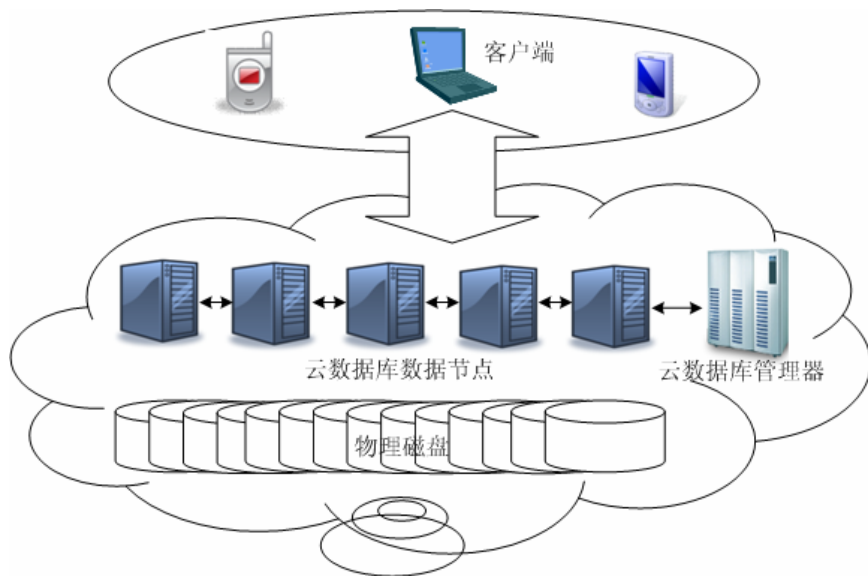
# 报告提要

- 云数据库概念和特点
- 云数据库与传统的分布式数据库
- 云数据库的影响
- 云数据库产品
- 云数据库领域的研究问题
- 参考文献
- 附件





# 云数据库概念和特点



云数据库应用示意图

在云数据库应用中，客户端不需要了解云数据库的底层细节，所有的底层硬件都已经被虚拟化，对客户端而言是透明的，它就像在使用一个运行在单一服务器上的数据库一样，非常方便容易，同时又可以获得理论上近乎无限的存储和处理能力。

## 云数据库概念

- 云数据库是部署和虚拟化在云计算环境中的数据库 [Yoon11]

## 云数据库特点

- 动态可扩展
- 高可用性
- 较低的使用代价
- 易用性
- 大规模并行处理

## Cloud Database

海量存储需求的必然选择



# 云数据库与传统的分布式数据库

## 分布式数据库概念

**分布式数据库**是计算机网络环境中各场地或节点上的数据库的逻辑集合。逻辑上它们属于同一系统，而物理上它们分散在用计算机网络连接的多个节点/场地，并统一由一个分布式数据库管理系统管理[ZhengYG99]。

## 云数据库和分布式数据库的共同点

云数据库和传统的分布式数据库有着相似的地方，比如，都把数据放到不同的节点上。

## 云数据库和分布式数据库的区别

分布式数据库在可扩展性方面是无法和云数据库相比的：

- ❑ 由于需要考虑数据同步和分区失败等开销，前者随着节点的增加，会导致DDB性能快速下降。
- ❑ 而云数据库则具有很好的可扩展性，因为后者在设计的时候，就已经避免了许多会影响到可扩展性的因素，比如采用更加简单的数据模型、对元数据和应用数据进行分离以及放松对一致性的要求等等[ChenCJOVWX10]

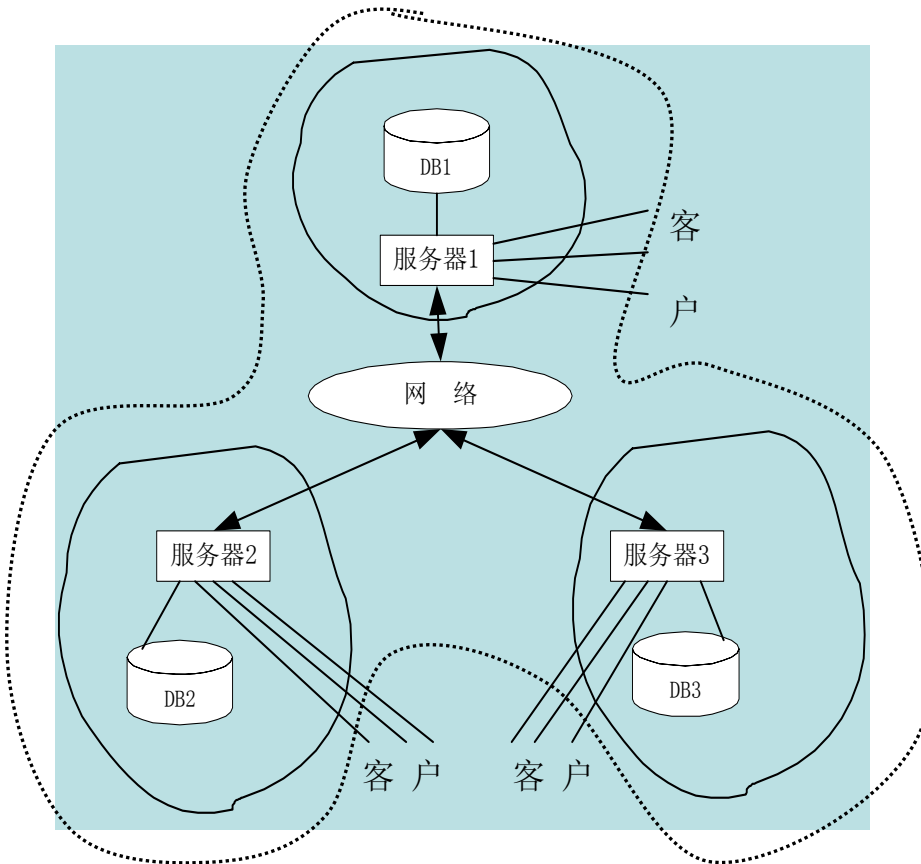
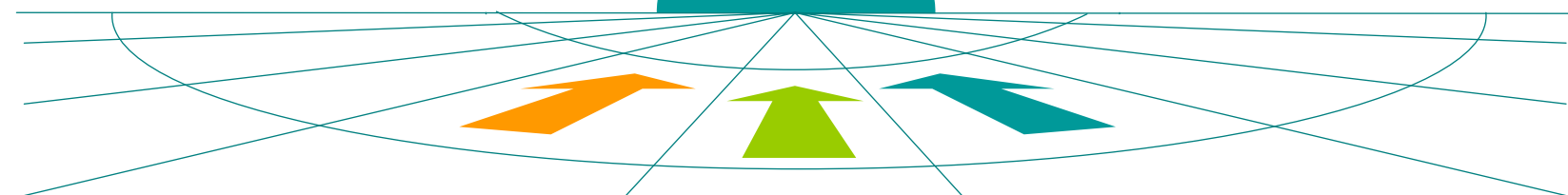


图 分布式数据库系统示意图



# 云数据库的影响

## Cloud 影响 Database



### 1、极大地改变企业管理数据的方式

- Forrester Research 分析师 Noel Yuhanna 指出，18% 的企业正在把目光投向云数据库。

- 中小企业会更多地采用云数据库产品，但是，对于大企业而言，云数据库并非首选，因为大企业通常自己建造数据中心。

### 2、催生新一代的数据库技术

- 第一代是20世纪70年代的早期关系数据库
- 第二代是80到90年代的更加先进的关系模型
- 第三代的数据库技术，要求数据库能够灵活处理各种类型的数据，而不是强制让数据去适应预先定制的数据结构。

- 从数据模型设计方式来看，已经有些产品（比如SimpleDB、HBase、Dynamo、BigTable）放弃传统的行存储方式，而采用键/值存储，从而可以在分布式的云环境中获得更好的性能。

### 3、数据库市场份额面临重新分配

- 此前，Teradata、Oracle、IBM DB2、Microsoft SQL Server、Sybase 等传统数据库厂商垄断市场

- Amazon 和 Google 扮演引领者角色

- 新的云数据库厂商开始出现 Vertica 和 EnterpriseDB



# 云数据库产品



- 传统的数据库厂商：Teradata、Oracle、IBM DB2和Microsoft SQL Server；
- 涉足数据库市场的云供应商：Amazon、Google和Yahoo；
- 新兴小公司：Vertica、LongJump和EnterpriseDB。

---

| 企业           | 产品  |
|--------------|---|
| Amazon       | SimpleDB、RDS  |
| Google       | BigTable、FusionTable、GoogleBase                     |
| Microsoft    | Microsoft SQL Azure                                 |
| Oracle       | Oracle Cloud  |
| Yahoo!       | PNUTS   |
| Vertica      | Analytic Database v3.0 for the Cloud                |
| EnterpriseDB | Postgres Plus in the Cloud                          |
| 开源项目         | HBase、Hypertable                                    |
| 其他           | EnterpriseDB、FathomDB、ScaleDB、Objectivity/DB、M/DB:X |



# 云数据库产品

## 云数据库市场先行者

- 提供著名的S3存储服务 and EC2 计算服务，提供基于云的数据库服务SimpleDB
- Amazon EC2应用托管服务可以部署多种数据库产品，如SQL Server、Oracle 11g、MySQL和IBM DB2等数据库平台

## 云数据库市场主力军

- Google BigTable是一种满足弱一致性要求的大规模数据库系统
- Google开发的另一款云计算数据库产品是Fusion Tables，采用了基于数据空间的技术

### 云数据库产品

Amazon

Google

开源

Microsoft

## 云数据库市场重要参与者

- HBase[CryansAA08]和Hypertable利用开源MapReduce平台Hadoop提供了类似于BigTable的可伸缩数据库实现
- 甲骨文开源数据库产品BerkelyDB也提供了云计算环境中的实现

## 云数据库市场主力军

- SQL Azure可以允许用户通过网络在云中创建、查询和使用SQL SERVER数据库
- 属于关系型数据库
- 支持云中的事务（局部事务）
- 支持存储过程



# 云数据库领域的研究问题







# 云数据库领域的研究问题-数据模型

## 键/值模型

### BigTable

一个BigTable实际上就是一个稀疏的、分布的、永久的多维排序图，它采用行键（row key）、列键（column key）和时间戳（timestamp）对图进行索引。图中的每个值都是未经解释的字节数组。

#### 行键

■BigTable在行键上根据字典顺序对数据进行维护。对于一个表而言，行区间是根据行键的值进行动态划分的。每个行区间称为一个Tablet，它是负载均衡和数据分发的基本单位，这些Tablet会被分发到不同的数据服务器上。

#### 列键

■被分组成许多“列家族”的集合，它是基本的访问控制单元。存储在一个列家族当中的所有数据，通常都属于同一种数据类型，这通常意味着具有更高的压缩率。数据可以被存放到列家族的某个列键下面，但是，在把数据存放到这个列家族的某个列键下面之前，必须首先创建这个列家族。在创建完成一个列家族以后，就可以使用同一个家族当中的列键。

#### 时间戳

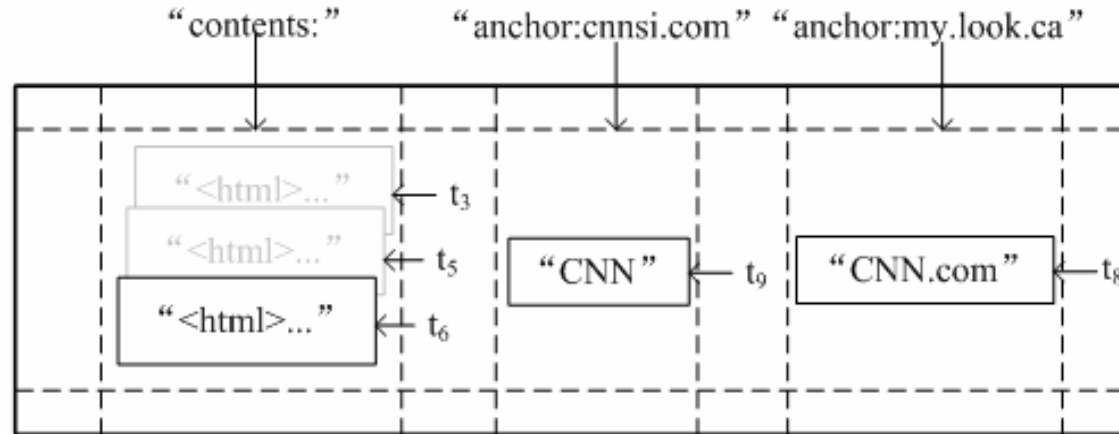
■在BigTable中的每个单元格当中，都包含相同数据的多个版本，这些版本采用时间戳进行索引。BitTable时间戳是64位整数。一个单元格的不同版本是根据时间戳降序的顺序进行存储的，这样，最新的版本可以被最先读取。



# 云数据库领域的研究问题-数据模型

## BigTable

“com.cnn.www”



| Row Key     | Timestamp | Column Family |     |                  |     |                   |     |
|-------------|-----------|---------------|-----|------------------|-----|-------------------|-----|
|             |           | contents:     | ... | anchor:cnnsi.com | ... | anchor:my.look.ca | ... |
| com.cnn.www | t9        |               |     | CNN              |     |                   |     |
|             | t8        |               |     |                  |     | CNN.com           |     |
|             | t7        | ...           |     | ...              |     | ...               |     |
|             | t6        | <html>...     |     |                  |     |                   |     |
|             | t5        | <html>...     |     |                  |     |                   |     |
|             | t4        | ...           |     | ...              |     | ...               |     |
|             | t3        | <html>...     |     |                  |     |                   |     |

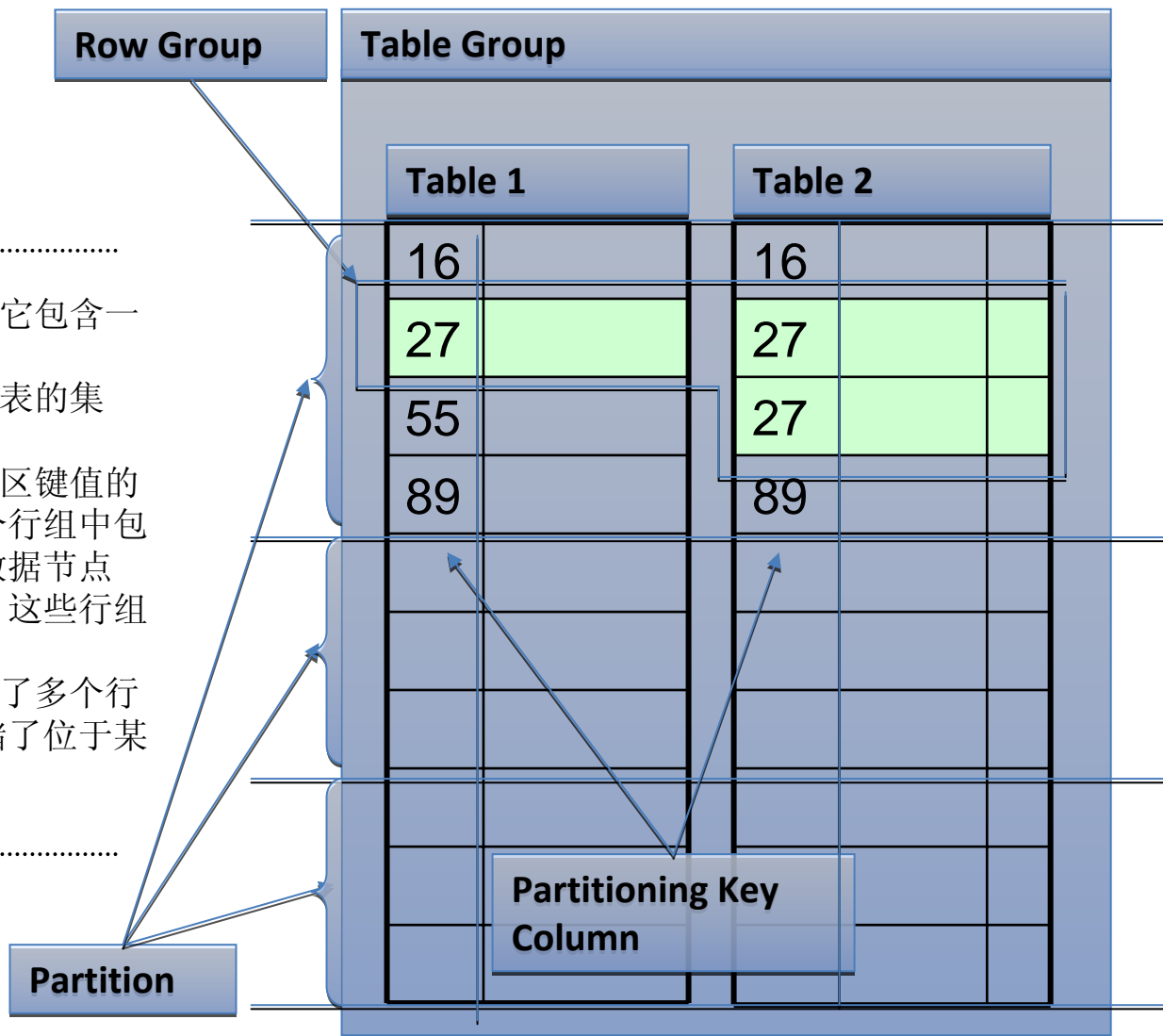


# 云数据库领域的研究问题-数据模型



## 关系模型

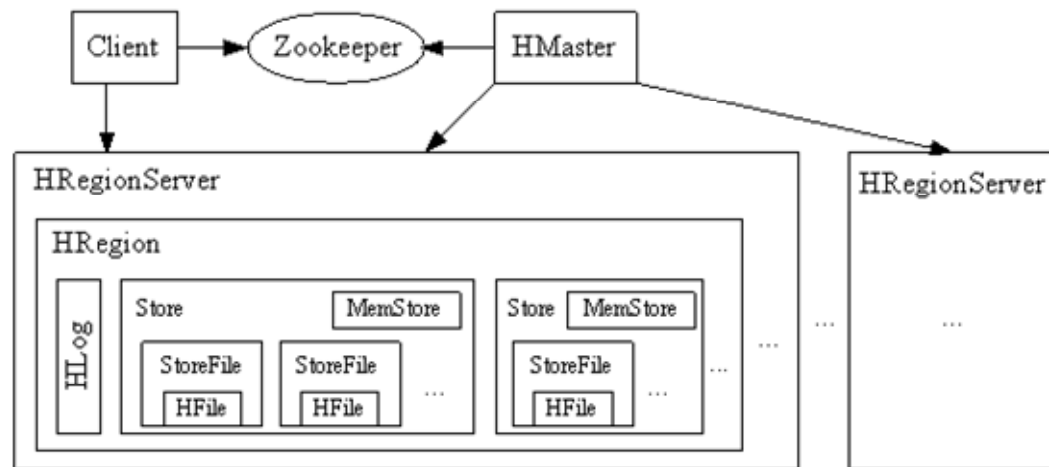
- **表**: 一个表是一个逻辑关系，它包含一个分区键，用来对表进行分区。
- **表组**: 具有相同分区键的多个表的集合，称为表组。
- **行组**: 在表组中，具有相同分区键值的多个行的集合，称为行组。一个行组中包含的行，总是被分配到同一个数据节点上。每个表组会包含多个行组，这些行组会被分配到不同的数据节点上。
- **数据分区**: 一个数据分区包含了多个行组。因此，每个数据节点都存储了位于某个分区键值区间内的所有行。





# 云数据库领域的研究问题-体系架构

## HBase 体系架构



HBase作为BigTable的一个开源实现，基本采用了和BigTable类似的架构。HBase体系架构中包括Client、Zookeeper、HMaster、HRegionServer和Store，具体功能如下：

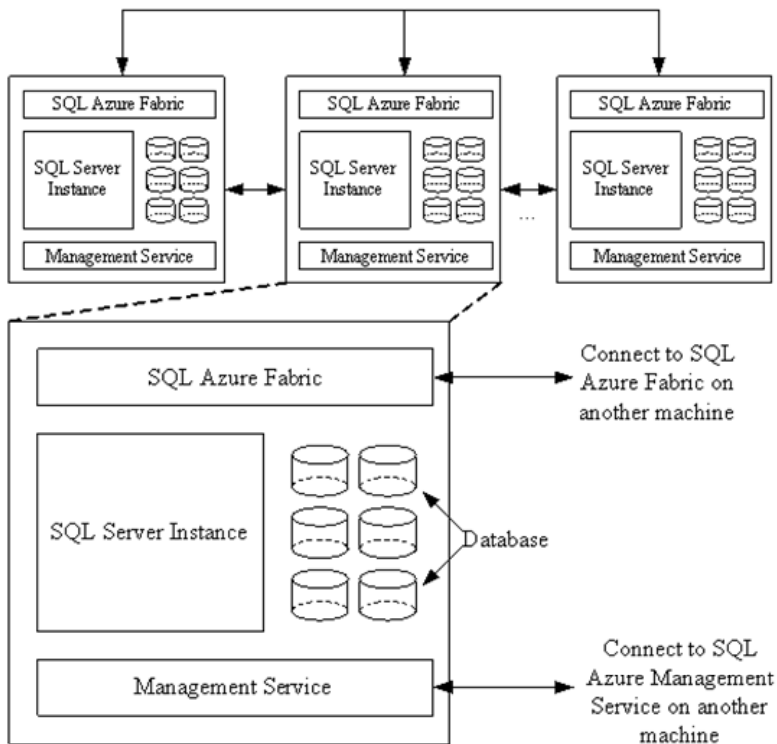
- Client: 访问HBase的接口；
- Zookeeper: 存储了HBase的数据库模式和所有HRegion的寻址入口、实时监控HRegionServer的状态；
- HMaster: 管理用户对Table的增、删、改、查操作，管理HRegionServer的负载均衡，调整Region分布等；
- HRegionServer: 负责响应用户I/O请求，向HDFS文件系统中读写数据，是HBase中最核心的模块；
- Store: 是HBase存储的核心，由两部分组成：MemStore和StoreFiles。用户写入的数据首先会放入MemStore，当MemStore满了以后会被存放到一个StoreFile中，StoreFile将被存放在HDFS文件系统的HFile中。



# 云数据库领域的研究问题-体系架构

## SQL Azure

### 体系架构



■ SQL Azure的体系架构中包含了一个虚拟机簇，可以根据工作负载的变化，动态增加或减少虚拟机的数量。

■ 每台虚拟机(SQL Server VM)安装SQL Server 2008数据库管理系统，以关系模型存储数据，通常一份数据库会被散存储到3至5台SQL Server VM中。

■ 每台SQL Server VM同时安装了SQL Azure Fabric和SQL Azure Management Service，后者负责对每个数据库间的数据复写工作，以保障SQL Azure的基本高可用性要求。

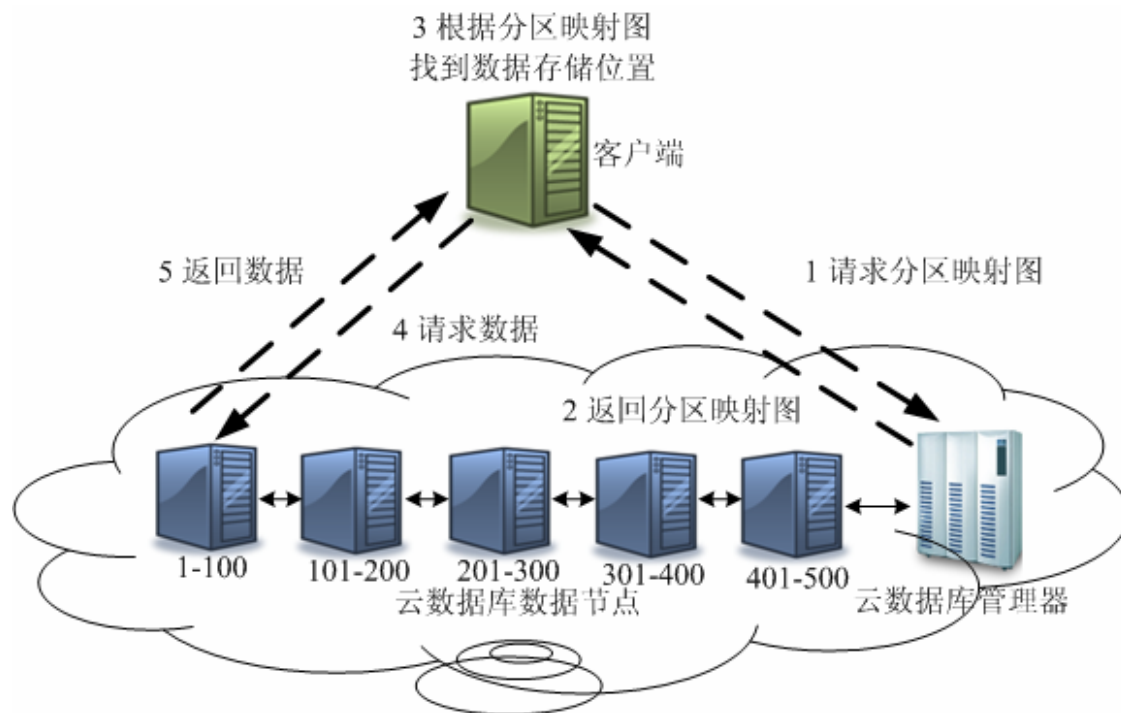
■ 不同SQL Server VM内的SQL Azure Fabric和Management Service会彼此交换监控信息，以保持整体服务的可监控性。

## Microsoft

Be what's next.



# 云数据库领域的研究问题-体系架构



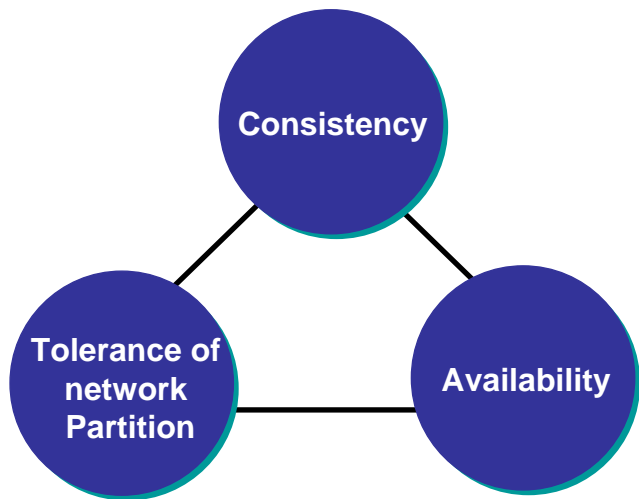
## 数据访问方法

- ❑ 1、客户端首先向管理器请求一份分区映射图
- ❑ 2、管理器向客户端发送分区映射图
- ❑ 3、客户端在映射图中根据键值找到所需数据的存储位置
- ❑ 4、客户端到指定的数据节点请求数据
- ❑ 5、由该数据节点把数据返回给客户端

实际上，为了改进性能，同时也为了避免管理器的性能瓶颈，通常会在客户端缓存常用的分区映射图，这样，客户端在很多情况下不用与管理器交互就可以直接访问相应的数据节点。



# 云数据库领域的研究问题-事务一致性



## CAP理论的启示

- CAP理论告诉我们，一个分布式系统不可能满足一致性，可用性和分区容错性这三个需求，最多只能同时满足两个。
- 在一个系统中，可以对某些数据做到CP，对另一些数据做到AP，就算是对同一个数据，调用者可以指定不同的算法，某些算法可以做到CP，某些算法可以做到AP。

当处理CAP的问题时，可以有4个选择：



## BASE: Basically Available, Soft-state, Eventual Consistency

- BASE模型反ACID模型，完全不同ACID模型，牺牲高一致性，获得可用性或可靠性
- BASE思想主要强调基本的可用性，如果你需要高可用性，也就是纯粹的高性能，那么就要以一致性或容错性为牺牲



# 云数据库领域的研究问题-事务一致性

很多现有的云数据库产品在实现时，为了获得CAP理论中的“可用性”和“分布性”，都放松了对事务ACID四性的要求。

■Google BigTable就弱化了事务的原子性要求，只支持单行事务，可以允许对存储在某个行键下面的数据执行原子的“读-修改-写”操作。

■Microsoft SQL Azure没有实现通用的事务，只支持局部事务，不支持分布式事务。

■Amazon SimpleDB放松了对事务的一致性和隔离性的要求，转而采用“最终一致性”（用户B虽然不能立即看到用户A修改的信息，但是，用户B最终会看到用户A的修改信息），使得所有副本不必都获得数据副本的当前最新值。但是，SimpleDB可以支持强一致读以及基于条件更新或者删除的乐观锁机制，并提供了简单的SQL Select子集支持。Amazon Dynamo也采用了最终一致性。

■Yahoo! PNUTS考虑到大部分WEB应用对一致性并没有非常严格的要求，因此，在设计上放弃了对强一致性的追求（采用时间线一致性），转而追求更高的可用性、容错能力和更快的响应等。

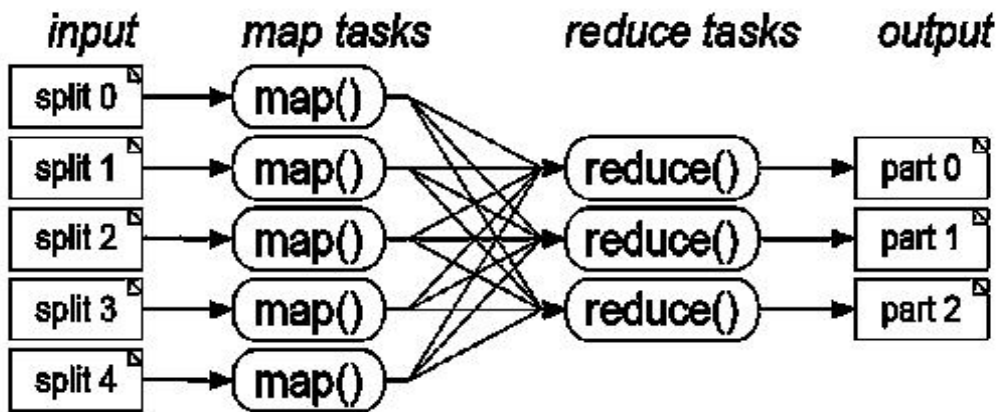




# 云数据库领域的研究问题-编程模型

## MapReduce

- Map/Reduce计算流程





# 云数据库领域的研究问题-编程模型

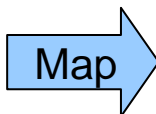
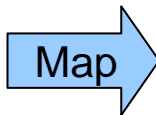
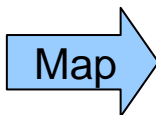
## MapReduce实例分析: WordCount

### Input

1, "Hello World Bye World"

2, "Hello Hadoop Bye Hadoop"

3, "Bye Hadoop Hello Hadoop"



### Output.Collecter

<Hello,1>  
<World,1>  
<Bye,1>  
<World,1>

<Hello,1>  
<Hadoop,1>  
<Bye,1>  
<Hadoop,1>

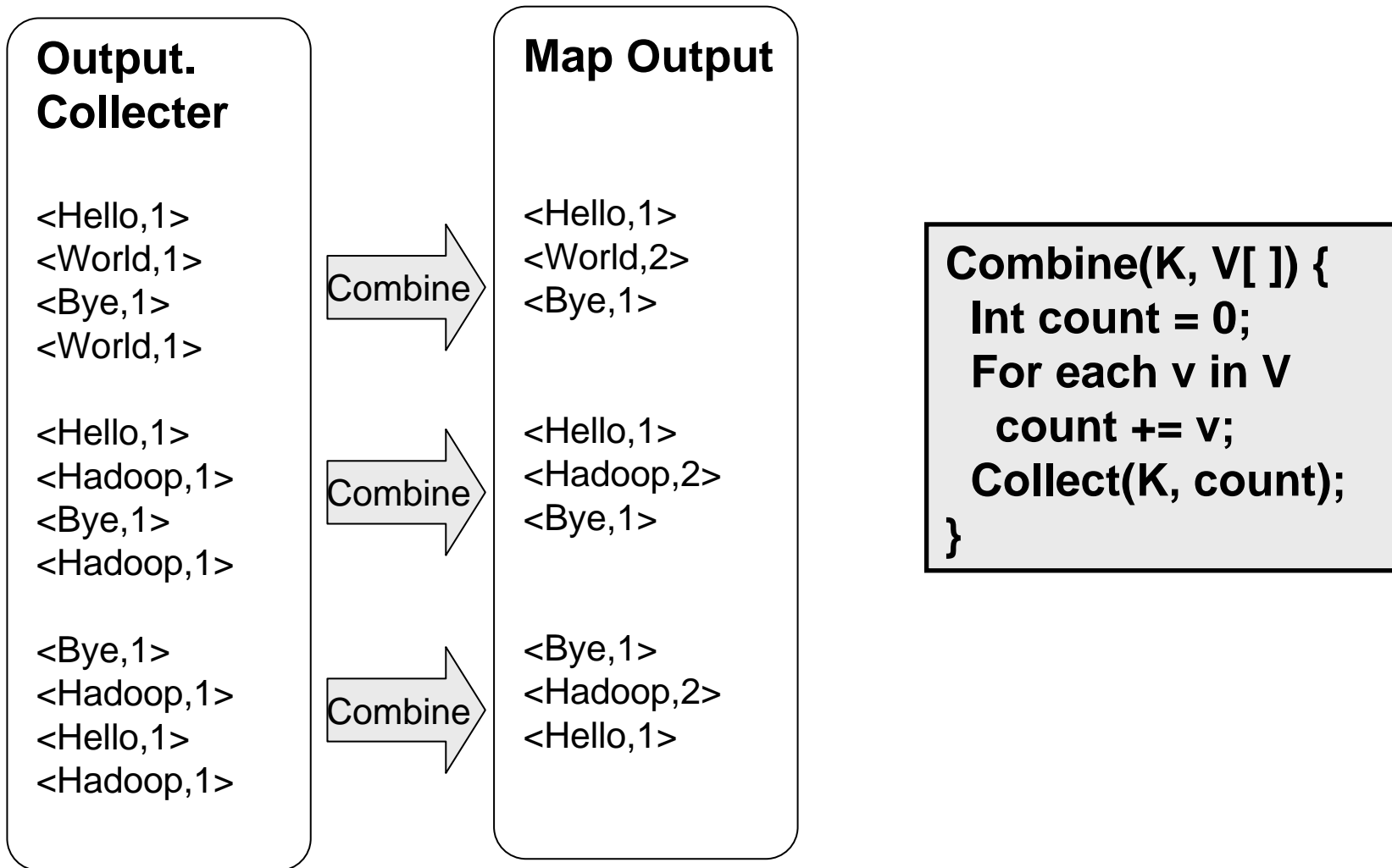
<Bye,1>  
<Hadoop,1>  
<Hello,1>  
<Hadoop,1>

```
Map(K, V) {  
  For each word w in  
  V  
    Collect(w, 1);  
}
```



# 云数据库领域的研究问题-编程模型

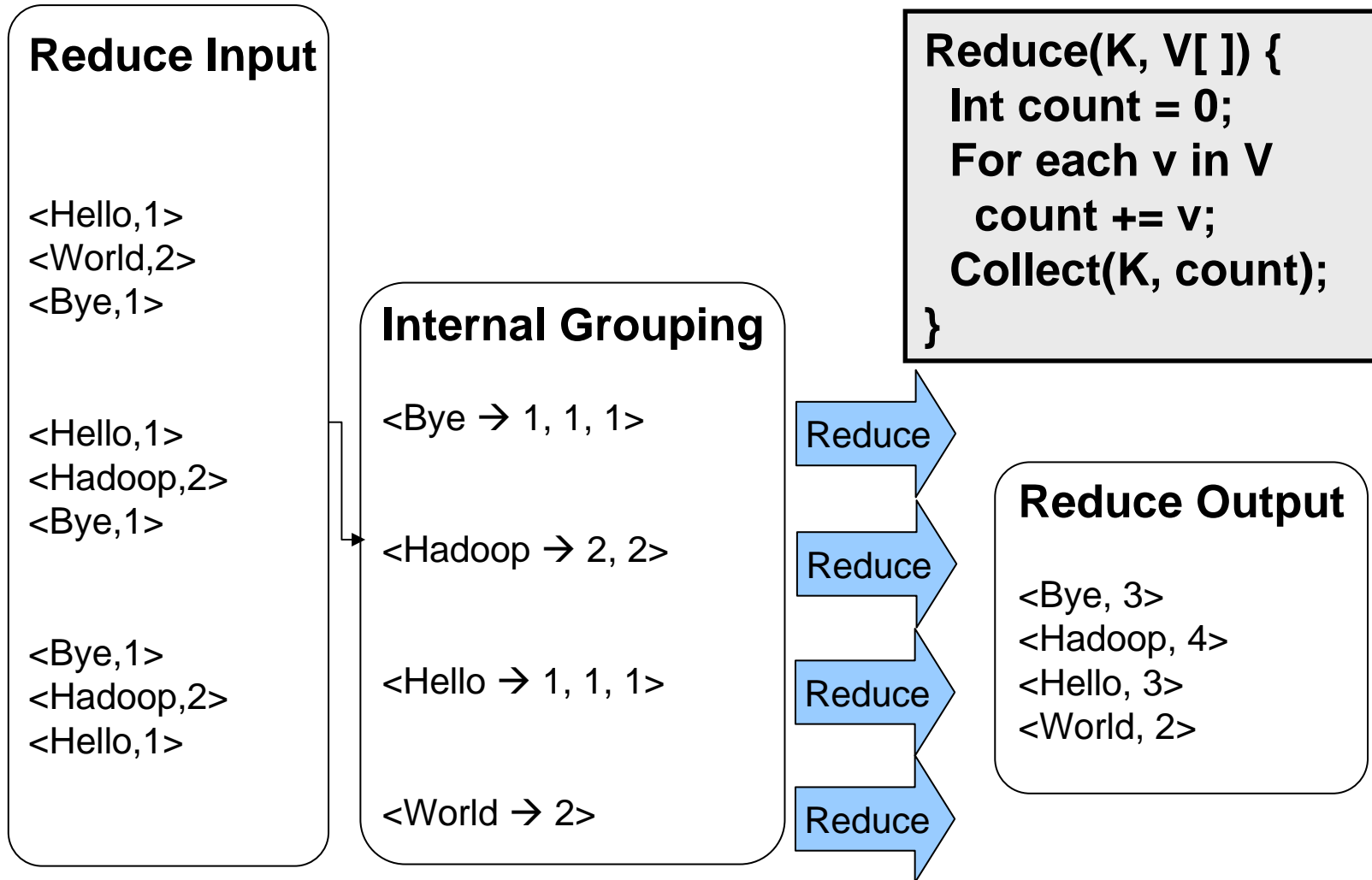
## MapReduce实例分析: WordCount





# 云数据库领域的研究问题-编程模型

## MapReduce实例分析: WordCount





# 云数据库领域的研究问题-编程模型

## MapReduce

在MapReduce环境下执行两个关系的联接操作

- ❑ 假设关系 $R(A,B)$ 和 $S(B,C)$ 都存储在一个文件中。
- ❑ 为了联接这些关系，必须把来自每个关系的各个元组都和一个key关联，这个key就是属性B的值。
- ❑ 可以使用一个Map进程集合，把来自R的每个元组 $(a,b)$ 转换成一个key-value对，其中的key就是b，值就是 $(a,R)$ 。注意，这里把关系R包含到value中，这样做使得我们可以在Reduce阶段，只把那些来自R的元组和来自S的元组进行匹配。
- ❑ 类似地，可以使用一个Map进程集合，把来自S的每个元组 $(b,c)$ ，转换成一个key-value对，key是b，value是 $(c,S)$ 。这里把关系名字包含在属性值中，可以使得在Reduce阶段只把那些来自不同关系的元组进行合并。
- ❑ Reduce进程的任务就是，把来自关系R和S的具有共同属性B值的元组进行合并。这样，所有具有特定B值的元组必须被发送到同一个Reduce进程。
- ❑ 假设使用k个Reduce进程。这里选择一个哈希函数h，它可以把属性B的值映射到k个哈希桶，每个哈希值对应一个Reduce进程。每个Map进程把key是b的key-value对，都发送到与哈希值 $h(b)$ 对应的Reduce进程。Reduce进程把联接后的元组 $(a,b,c)$ ，写到一个单独的输出文件中。



# 参考文献

- [ChenZ09]Chen K, Zheng WM. Cloud computing: System instances and current research. Journal of Software, 2009, 20(5):1337–1348.
- [DashKA09]Debabrata Dash, Verena Kantere, Anastasia Ailamaki: An Economic Model for Self-Tuned Cloud Caching. ICDE 2009:1687-1693.
- [LiuWW09]Qin Liu, Guojun Wang, Jie Wu: An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing. CSE 2009:715-720.
- [Hadoop]Hadoop, an open source implementing of MapReduce and GFS, <http://hadoop.apache.org>.
- [FengZZX11]Feng Dengguo, Zhang Min, Zhang Yan, Xu Zhen. Study on Cloud Computing Security.2011,22(1):71-83.
- [XuGDLS09]Meng Xu, Dan Gao, Chao Deng, Zhiguo Luo, Shaoling Sun: Cloud Computing Boosts Business Intelligence of Telecommunication Industry. CloudCom 2009:224-231.
- [QiQL09]Ji Qi, Ling Qian, Zhiguo Luo: Distributed Structured Database System HugeTable. CloudCom 2009:338-346.
- [DeWitt08]D. DeWitt and M. Stonebraker. MapReduce: A major step backwards. DatabaseColumn Blog. <http://www.databasecolumn.com/2008/01/mapreduce-a-major-step-back.html>.
- [HoldenKBI09]Edward P. Holden, Jai W. Kang, Dianne P. Bills, Mukhtar Ilyassov: Databases in the cloud: a work in progress. SIGITE Conference 2009:138-143.
- [AbouzeidBARS09]Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel J. Abadi, Alexander Rasin, Avi Silberschatz: HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. PVLDB 2(1):922-933 (2009).
- [AhrensA11]Maximilian Ahrens, Gustavo Alonso: Relational databases, virtualization, and the cloud. ICDE 2011:1254
- [AgrawalADE11]Divyakant Agrawal, Amr El Abbadi, Sudipto Das, Aaron J. Elmore: Database Scalability, Elasticity, and Autonomy in the Cloud - (Extended Abstract). DASFAA 2011:2-15
- [SoaresP11]Luís Soares, José Pereira: Improving the Scalability of Cloud-Based Resilient Database Servers. DAIS 2011:136-149.
- [IonRC11]Mihaela Ion, Giovanni Russello, Bruno Crispo: Enforcing Multi-user Access Policies to Encrypted Cloud Databases. POLICY 2011:175-177.
- [SimpleDB]Amazon Web Services. SimpleDB. Web Page. <http://aws.amazon.com/simpledb/>.



# 参考文献

- [ChangDGHWBCFG08]Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Michael Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber: Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst. (TOCS) 26(2) (2008).
- [Bain09]Tony Bain. Is the Relational Database Doomed? <http://www.readwriteweb.com/enterprise/2009/02/is-the-relational-database-doomed.php>..
- [Yoon11]Jong P. Yoon. Access Control and Trustiness for Resource Management in Cloud Databases. In the proceedings of Grid and Cloud Database Management, 109-131, 2011.
- [Wang08]Wang Zhongyuan. Massive storage leads to cloud database, China Information World, 2008.
- [ZhengYG99] Zheng Zhenmei, Yu Ge, Guo Min. Distributed Database, Science Press, 1999.
- [ChenCJOVWX10]Chun Chen, Gang Chen, Dawei Jiang, Beng Chin Ooi, Hoang Tam Vo, Sai Wu, Quanqing Xu: Providing Scalable Database Services on the Cloud. WISE 2010:1-19
- [Olofson06]C. Olofson. Worldwide RDBMS 2005 vendor shares. Technical Report 201692, IDC, May 2006.
- [Vertica2011]<http://www.vertica.com/wp-content/uploads/2011/01/VerticafortheCloudDatashet.pdf>
- [EnterpriseDB]<http://enterprisedb.com/solutions/postgres-plus-cloud-computing>
- [ChangDGHWBCFG06]Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Michael Burrows, Tushar Chandra, Andrew Fikes, Robert Gruber: Bigtable: A Distributed Storage System for Structured Data (Awarded Best Paper!). OSDI 2006:205-218.
- [GhemawatGL03]Ghemawat, S., Gobiuff, H., and Leung, S.-T. The Google file system. In Proc. of the 19th ACM SOSP (Dec.2003), pp. 29–43.
- [Burrows06]Burrows, M. The Chubby lock service for loosely coupled distributed systems. In Proc. of the 7th OSDI (Nov. 2006).
- [GonzalezHJLMSS10]Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen: Google fusion tables: data management, integration and collaboration in the cloud. SoCC 2010:175-180.
- [GonzalezHJLMSSG10]Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, Jonathan Goldberg-Kidon: Google fusion tables: web-centered data management and collaboration. SIGMOD 2010:1061-1066.
- [GonzalezHLMSSG10]Hector Gonzalez, Alon Y. Halevy, Anno Langen, Jayant Madhavan, Rod McChesney, Rebecca Shapley, Warren Shen, Jonathan Goldberg-Kidon: Socialising Data with Google Fusion Tables. IEEE Data Eng. Bull. (DEBU) 33(3):25-32 (2010).



# 参考文献

- [CryansAA08]Jean-Daniel Cryans, Alain April, Alain Abran: Criteria to Compare Cloud Computing with Current Database Technology. IWSM/Metrikon/Mensura 2008:114-126.
- [DeanG04]Jeffrey Dean, Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters. OSDI 2004:137-150.
- [CooperRSSBJPWY08]B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. Pnuts: Yahoo!'s hosted data serving platform. PVLDB, 1(2):1277-1288, 2008.
- [CurinoJPMWMBZ11]Carlo Curino, Evan P. C. Jones, Raluca A. Popa, Nirmesh Malviya, Eugene Wu, Samuel Madden, Hari Balakrishnan, Nikolai Zeldovich: Relational Cloud: a Database Service for the cloud. CIDR 2011:235-240
- [DeCandiaHJKLPSVV07]Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Werner Vogels: Dynamo: amazon's highly available key-value store. SOSP 2007:205-220.
- [DasAA10]Sudipto Das, Divyakant Agrawal, Amr El Abbadi: G-Store: a scalable data store for transactional multi key access in the cloud. SoCC 2010:163-174.
- [HacigumusTHMPSCJ10]Hakan Hacigümüs, Jun'ichi Tatemura, Wang-Pin Hsiung, Hyun Jin Moon, Oliver Po, Arsany Sawires, Yun Chi, Hojjat Jafarpour: CloudDB: One Size Fits All Revived. SERVICES 2010:148-149
- [Stonebraker08]Michael Stonebraker: Technical perspective - One size fits all: an idea whose time has come and gone. Commun. ACM (CACM) 51(12):76 (2008)
- [SYBASE]David Loshin. Gaining the Performance Edge Using a Column-Oriented Database Management System. White paper of SYBASE Inc.
- [HBaseArchitecture]<http://hbase.apache.org/book.html#architecture>
- [SQLAzureArchitecture]<http://msdn.microsoft.com/en-us/library/windowsazure/ee336271.aspx>
- [Zookeeper]Zookeeper project, <http://hadoop.apache.org/zookeeper>.
- [Browne2009]Julian Browne. Brewer's CAP Theorem. January 11, 2009. <http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>.
- [GilbertL02]S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News, 33(2):51–59, 2002.
- [Abadi09]Daniel J. Abadi: Data Management in the Cloud: Limitations and Opportunities. IEEE Data Eng. Bull. (DEBU) 32(1):3-12 (2009) .





# 参考文献

- [IslamV10]Md. Ashfakul Islam, Susan V. Vrbsky: Tree-Based Consistency Approach for Cloud Databases. CloudCom 2010:401-404.
- [ChohanBKN11]Navraj Chohan, Chris Bunch, Chandra Krintz, Yoshihide Nomura: Database-Agnostic Transaction Support for Cloud Infrastructures. IEEE CLOUD 2011:692-699
- [KraskaHAK09]Tim Kraska, Martin Hentschel, Gustavo Alonso, Donald Kossmann: Consistency Rationing in the Cloud: Pay only when it matters. PVLDB 2(1):253-264 (2009)
- [LometFWZ09]David B. Lomet, Alan Fekete, Gerhard Weikum, Michael J. Zwilling: Unbundling Transaction Services in the Cloud. CIDR 2009.
- [AfratiU10]Foto N. Afrati, Jeffrey D. Ullman: Optimizing joins in a map-reduce environment. EDBT 2010:99-110.
- [PavloPRADMS09]Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden, Michael Stonebraker: A comparison of approaches to large-scale data analysis. SIGMOD 2009:165-178
- [AboulnagaSSMKK09]Ashraf Aboulnaga, Kenneth Salem, Ahmed A. Soror, Umar Farooq Minhas, Peter Kokosielis, Sunil Kamath: Deploying Database Appliances in the Cloud. IEEE Data Eng. Bull. (DEBU) 32(1):13-20 (2009).
- [DasNAA11]Sudipto Das, Shoji Nishimura, Divyakant Agrawal, Amr El Abbadi: Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud using Live Data Migration. PVLDB 4(8):494-505 (2011)
- [CecchetSS11]Emmanuel Cecchet, Rahul Singh, Upendra Sharma, Prashant J. Shenoy: Dolly: virtualization-driven database provisioning for the cloud. VEE 2011:51-62.
- [XiongCZMPH11]PengCheng Xiong, Yun Chi, Shenghuo Zhu, Hyun Jin Moon, Calton Pu, Hakan Hacigümüs: Intelligent management of virtualized resources for database systems in cloud environment. ICDE 2011:87-98
- [RogersPC10]Jennie Rogers, Olga Papaemmanouil, Ugur Çetintemel: A generic auto-provisioning framework for cloud databases. ICDE Workshops 2010:63-68
- [CooperSTRS10]Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears: Benchmarking cloud serving systems with YCSB. SoCC 2010:143-154
- [ShiMZHLW10]Yingjie Shi, Xiaofeng Meng, Jing Zhao, Xiangmei Hu, Bingbing Liu, Haiping Wang: Benchmarking cloud-based data management systems. CloudDB 2010:47-54
- [KossmannKL10]Donald Kossmann, Tim Kraska, Simon Loesing: An evaluation of alternative architectures for transaction processing in the cloud. SIGMOD 2010:579-590.
- [WangWGLO10]Jinbao Wang, Sai Wu, Hong Gao, Jianzhong Li, Beng Chin Ooi: Indexing multi-dimensional data in a cloud system. SIGMOD 2010:591-602.
- [ElmoreDAA11]Aaron J. Elmore, Sudipto Das, Divyakant Agrawal, Amr El Abbadi: Zephyr: live migration in shared nothing databases for elastic cloud platforms. SIGMOD 2011:301-312.



# 参考文献

- [WangLWG11]Hongzhi Wang, Jianzhong Li, Jinbao Wang, Hong Gao. Dirty Data Management in Cloud Database. In the proceedings of Grid and Cloud Database Management, 133-150, 2011.
- [ThakarSCT11]Ani Thakar, Alexander S. Szalay, Ken Church, Andreas Terzis: Large science databases - are cloud services ready for them? Scientific Programming (SP) 19(2-3):147-159 (2011)
- [ThakarS10]Ani Thakar, Alexander S. Szalay: Migrating a (large) science database to the cloud. HPDC 2010:430-434
- [Wood2010]Timothy Wood and Emmanuel Cecchet and K. K. Ramakrishnan and Prashant Shenoy and Jacobus Van Der Merwe and Arun Venkataramani. Disaster Recovery as a Cloud Service: Economic Benefits & Deployment Challenges. Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010,8-8.



# 附件：报告人简介



**林子雨**

男，1978年12月22日出生于浙江省温州市；小学、初中、高中就读于吉林省柳河县；1997年09月考入福州大学土木建筑工程系，2001年07月本科毕业并获得工学学士学位；2002年09月考入厦门大学信息科学技术学院计算机科学技术系攻读工学硕士学位；2005年09月考入北京大学信息科学技术学院计算机软件与理论专业攻读理学博士学位；2009年07月应聘于厦门大学计算机科学系，开始执教生涯.....



**2005.09~2009.07**  
博士



北京  
大学

厦  
门  
大  
学



福  
州  
大  
学



**2002.09~2005.07** 硕士  
**2009.07至今** 讲师



**1997.09~2001.07**  
本科

The background is a solid blue color with faint, light blue silhouettes of people. At the top, there are two groups of people: one on the left holding hands in a circle, and one on the right standing in a line. On the right side, there is a large silhouette of a person talking on a mobile phone. In the bottom left corner, there are silhouettes of two people, one of whom appears to be holding a phone to their ear.

# Thank You!

Department of Computer Science, Xiamen University, November 15, 2011