HBase应用方案，主要内容包括四个方面：HBase实际应用中的性能优化方法，HBase性能监视，在HBase之上构建SQL引擎，构建HBase二级索引。

首先看一下HBase实际应用中的性能优化方法。第一个可以优化的是行键，一般按字典序存储，在实际应用中，我们一般希望经常使用的数据存储在一起，最近可能会被访问的数据也放在一起，这样可能会与时间有关系，因此可以考虑将时间戳作为行键的一部分，重新定义一个行键，比如用Long.MAX_VALUE – timestamp，这样能保证新写入的数据在读取时可以很快被命中。

第二个可以优化的是缓存，也就是InMemory。在创建表时，直接把表缓存到Region服务器中，通过调用HColumnDescriptor.setInMemory(true)，保证了高速和命中率。第三个可以优化的是设置最大版本，创建表时通过HColumnDescriptor.setMaxVersions(int maxVersions)设置表中数据的最大版本，保存最新版本的数据只需设置setMaxVersions(1)。第四个可以优化的是数据的生命周期，可以通过HColumnDescriptor.setTimeToLive(int timeToLive)来设置，一旦超过生命期的数据会自动被删除，比如存储近两天的数据可以设置setTimeToLive(2 * 24 * 60 * 60)。

下面我们来看一下怎么去监视你Hbase的性能，就是我们有时候需要经常的通过一些可视化的方式去实时的去监控，我们去辨识到底底层一些运作性能如何？我们可以通过这么几种工具去查看一下Hbase的性能，一个是master status，一个是Ganglia,还有两个分别是OpenTSDB,Ambari。

这个master status呢，是Hbase自身带的这么一款工具，你通过界面就可以查询，是他通过web界面的方式可以查询底层的一些运行状态，直接在浏览器中就可以输入这个地址就可以查看了。

Ganglia是伯克利发起的一个开源的集群监视项目，也是用于监控系统性能的，也支持对Hbase进行性能监控。

OpenTSDB可以从大规模的集群当中获取相关的性能参数进行存储，然后以可视化或web化的方式提供给管理员，让他们掌握Hbase的当前性能。

另一个工具Ambari 的作用就是创建、管理、监视 Hadoop 的集群，那么我们HBase也是这个集群一部分，他当然可以对他进行监视，所以我们说你在具体应用当中，可以通过上面几种工具，对HBase的性能进行相关的监视。

我们很多人都习惯用sql语句去查询，那我能不能说也通过sql来查询Hbase数据库的数据呢，答案是可以的，就是我们实际上现在已经有了一些产品，帮你在Hbase上去构建一个sql引擎，你直接用sql语句就可以去查询Hbase当中相关的数据，有这么几个原因，就是我们用这种方法去查,一个是比较容易使用，因为毕竟大部分的现代人员对这个比较熟悉，但对于Hbase相对比较陌生，所以我们用这种SQL架构在上面就比较容易使用，另外也可以减少代码量，如果直接用Hbase上那个接口去查询数据的话，你可能要编写非常多代码，但sql语句非常简洁啊，它是这种非过程性的语言，你只要写sql语句，它自动帮你执行相关的操作，这样可以减少代码量。典型方案有两种，一种是Hive整个Hbase，另一个是Phoenix。

首先我们看第1种，在Hive的0.6.0版本就已经开始具备了和HBase的整合功能，也就是说他们两者接口互相通信就可以实现对外界访问了。那么第二个就是Phoenix，Phoenix是指saas服务的一个供应商，就是Salesforce公司开源的一个项目叫phoenix，它是构建在Apache，Hbase上一个SQL中间层，就通过这个产品，允许开发者在APP上面去执行sql查询，所以典型呢，你可以通过这两种方式，当然现在可能还有其他方式，大家可以自己去了解一下。

我们再来看一下构建Hbase二级索引问题，二级索引就是辅助索引，在关系数据库当中大家是比较熟悉的，比如可以为学生表建立一个学号主索引叫做primary key,还可以对姓名、成绩等建立二级的辅助索引，在关系数据库中就是通过这种方式来建立多级索引。但对于Hbase来讲，是不支持对各个点建立多索引的，它只有一个针对行键的索引，并且访问表中的行时，原生的只有三种方式：通过单个行键访问，通过一个行键的区间来访问，再就是全表扫描。但在实际应用中，由于要加速数据分析，往往又需要二级索引，对不同的列去构建索引，而原生的Hbase又不支持，怎么办呢？是这样解决的，Hbase在0.92版本之后引入一个新特性Coprocessor，通过这个特性可以帮忙二级索引的构建，如：华为公司的Hindex二级索引，Redis,solr。

Coprocessor构建二级索引，它提供了两个实现方法，一个endpoint,相当于关系型数据库的存储过程，完成一些事务处理，另外一个是observer,相当于触发器，满足相应条件后触发相关事务，如插入数据前后可以做一些事务处理，同时也写入索引表。这样会有两个表，一个是主表，另一个是索引表，这样一来会导致一个缺点，就是耗时是双倍的，对HBase的集群的压力也是双倍的。但也有一个优点就是不需要改动Hbase，也不需要上层应用做什么事情，就可以构建二级索引。

华为公司的Hindex二级索引是纯Java编写的，支持多个表、多个列的索引，还支持基于部分列值的索引。

还有一个就是Hbase+Redis方案, 由于Coprocessor建索引时耗时是双倍的，这对性能不利，为了解决这个问题，引入Redis做客户端缓存，也就是构建的索引先放在redis中，由Redis来管理这个索引，而不是直接写Hbase的主表和索引表，索引实时更新到Redis中，并且定期将redis中的索引更新到Hbaser 底层索引表中，这就减少了更新索引代价的开销。

还有一种方案就是Solr+Hbase方案，Solr是一个高性能，基于Lucene的全文搜索服务器。我们可以输入一个关键字查询Hbase数据，solr可以快速找到相关的行键,并在Solr建立索引，是一款非常优秀的全文搜索引擎。

安装：

请参考课本Hbase章后实验指导以及厦门大学大数据实验室网站的指导博客，完成Hbase数据库的安装和配置，并练习使用Hbase数据库操作命令，本次实验自行联系，不布置作业。