

Hadoop Ecosystem



PIG



- PIG Latin – high level language
- Runs a map reduce program
- Users can create their own user defined functions
- Mainly used for ETL
- Powerful transformation capabilities

Running pig

- Local mode : `pig -x local`
- Grunt shell : `pig`
- Script : `pig myscript.pig`

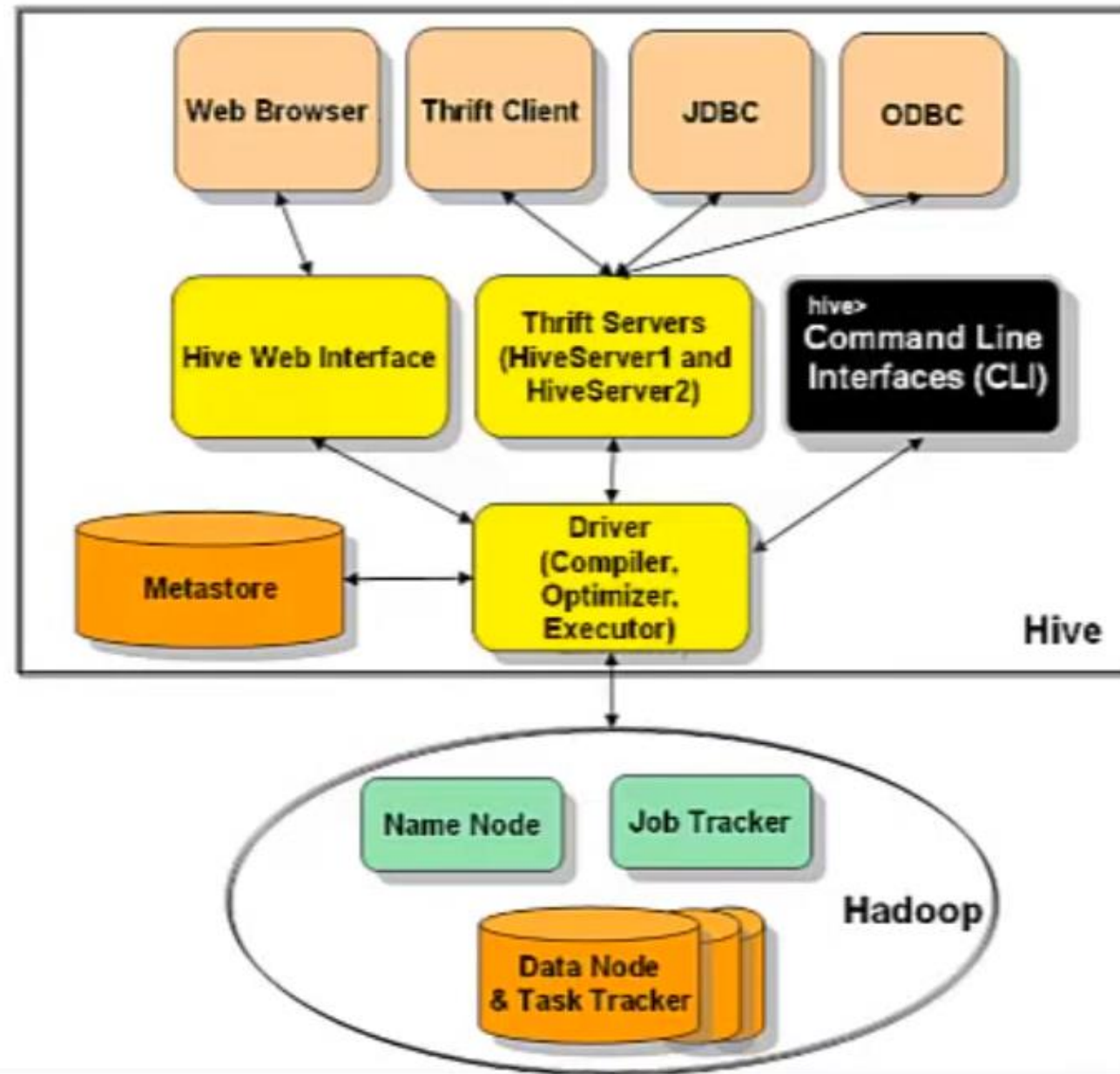
HIVE



- Hadoop warehouse (DW built on top of hadoop)
- Developed at facebook
- Currently used at fb for reporting dashboards and adhoc analysis
- Facilitates easy data summarisation, ad-hoc queries analysis of large datasets
- Provides a SQL interface (HQL)
- Runs as map reduce jobs in background

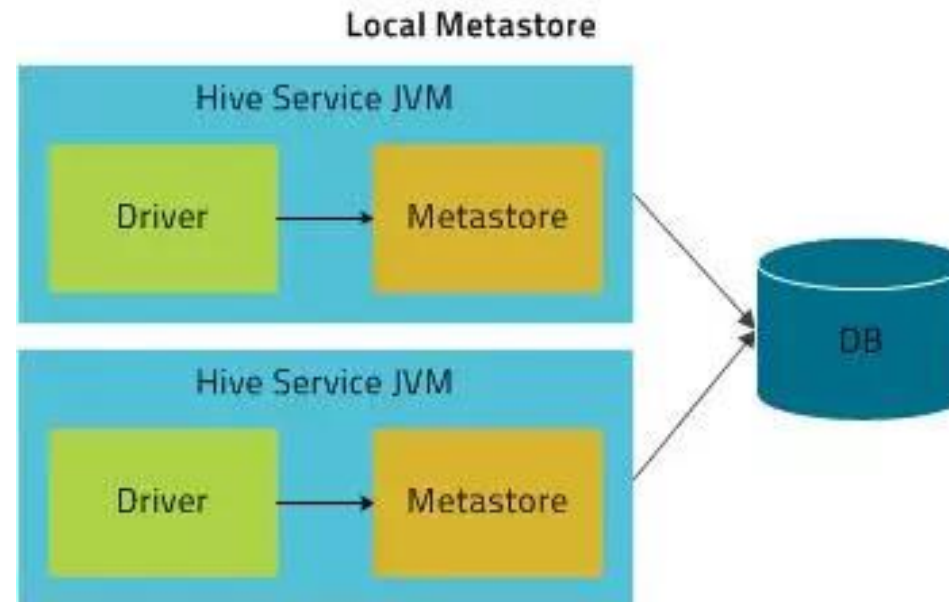
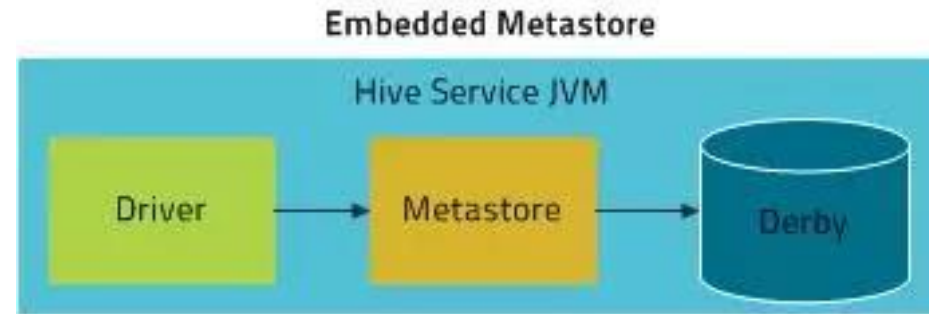
- **Metastore** - The Metastore stores the system catalog and metadata about tables, columns, partitions and so on.
- The metastore makes mapping file structure to a tabular form possible in Hive
- **Hcatalog** - HCatalog is built on top of the Hive metastore and incorporates Hive's DDL. Hcatalog provides read and write interfaces for Pig and MapReduce and uses Hive's command line interface for issuing data definition and metadata exploration commands.
- HCatalog makes it easier for users of Pig, MapReduce, and Hive, to read and write data on the grid
- **Warehouse Directory** – In hive your data is stored in the warehouse directory as configured in the hive-site.xml

Hive Components

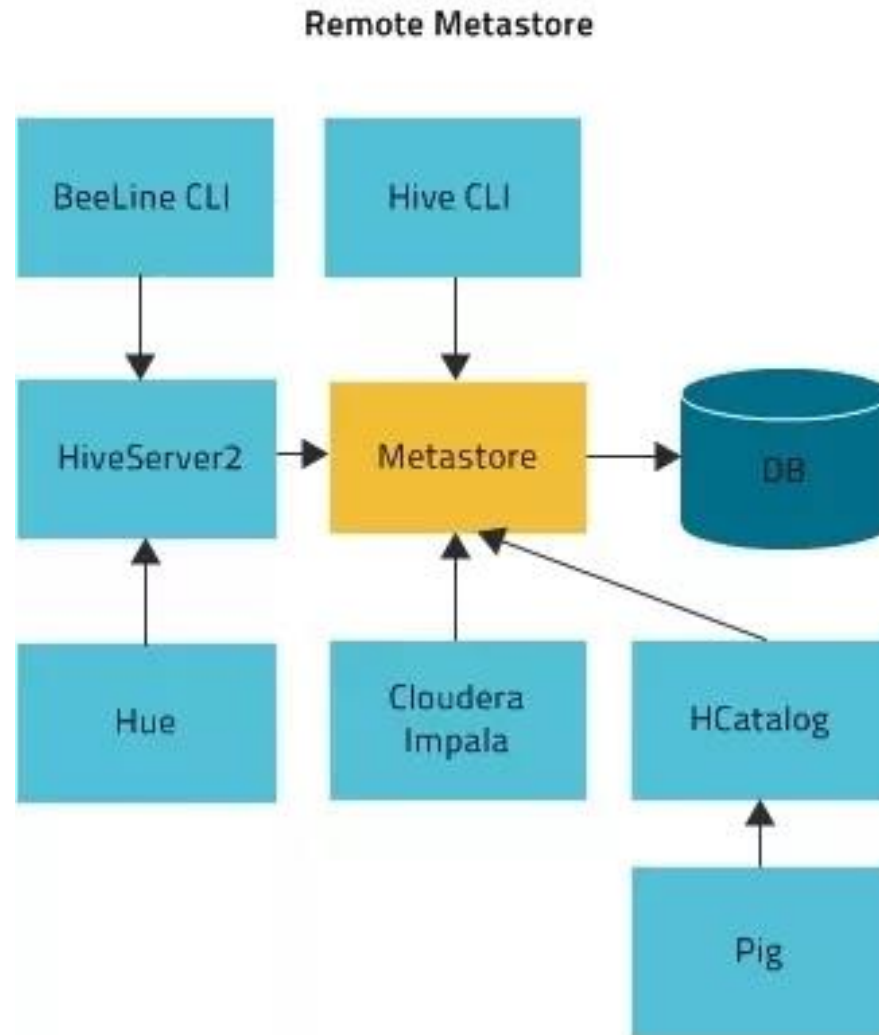


© 2013

Hive metastore deployment



Hive metastore deployment



Sqoop



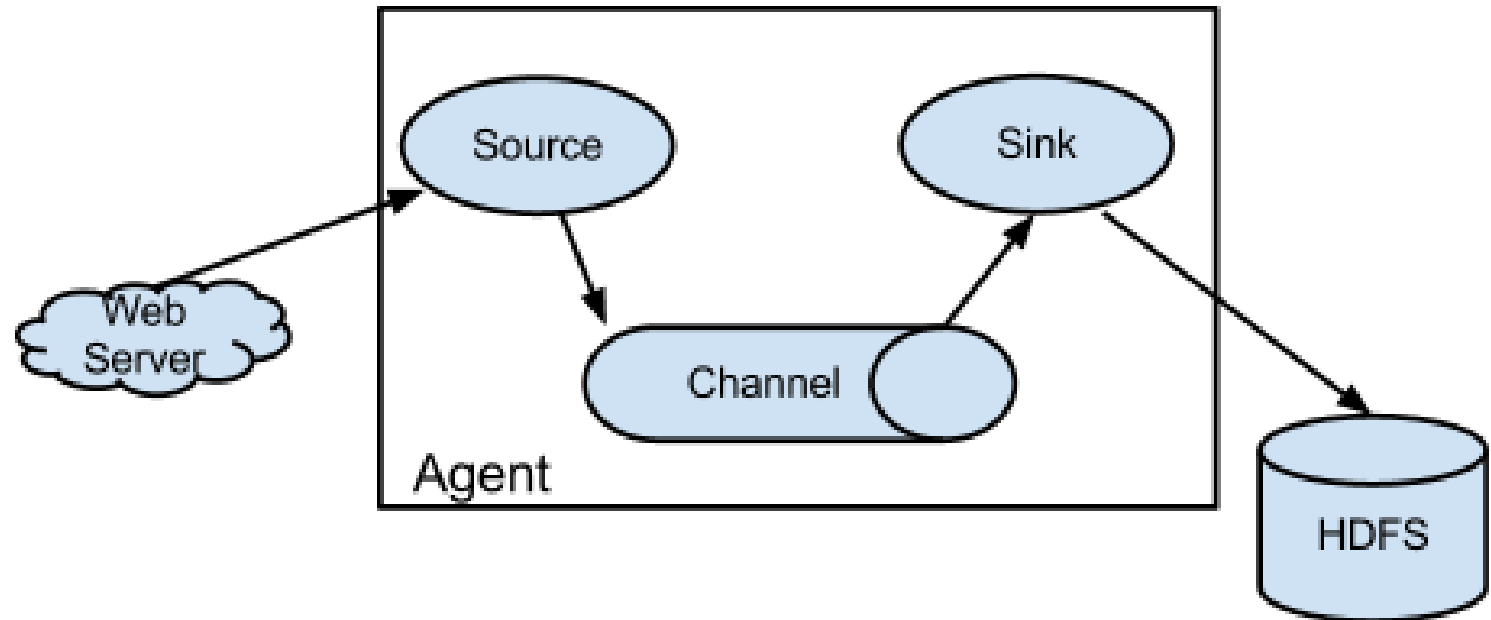
- Apache Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
- Uses map reduce to import and export data

Flume



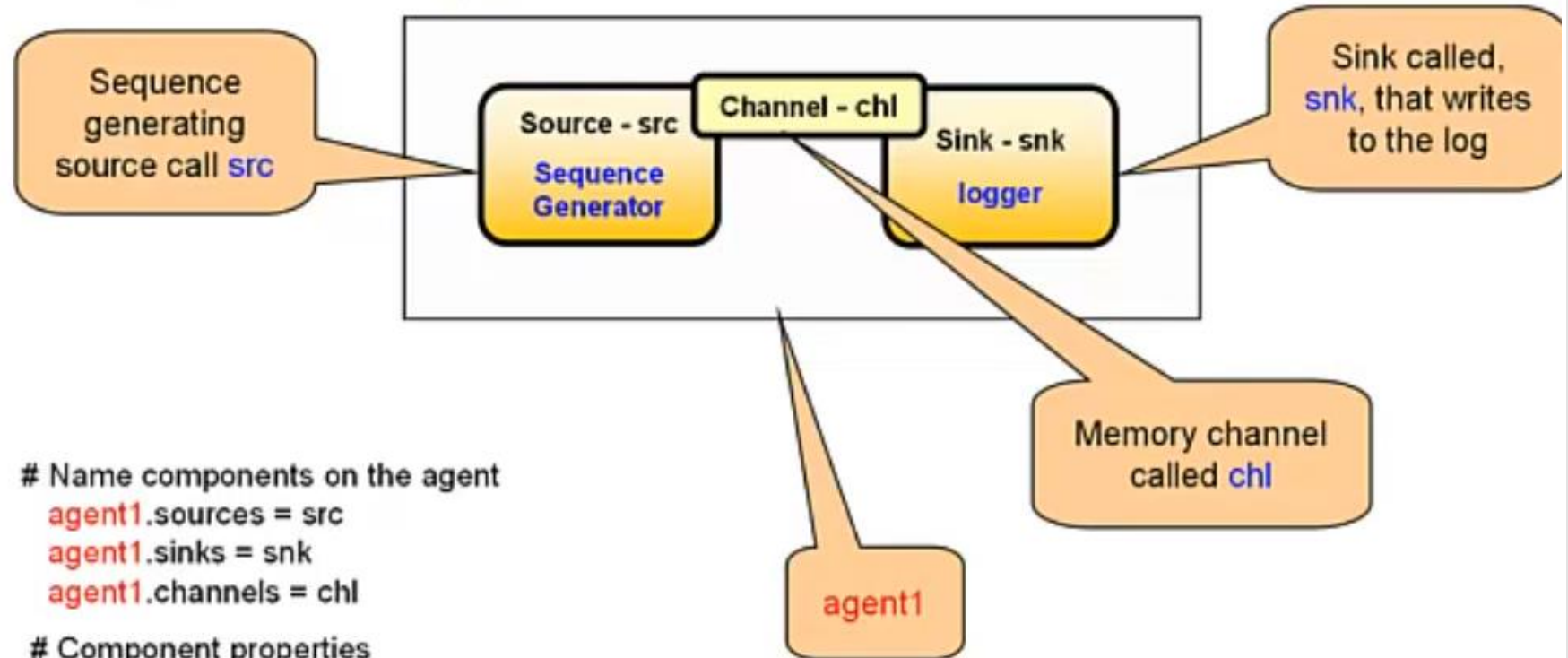
- Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
- Built on concept of flows
- It has a simple and flexible architecture based on streaming data flows.

Flume architecture



Flume configuration

Configuration example



Name components on the agent

```
agent1.sources = src  
agent1.sinks = snk  
agent1.channels = chl
```

Component properties

```
agent1.sources.src.type = seq  
agent1.sinks.snk.type = logger  
agent1.channels.chl.type = memory  
agent1.channels.chl.capacity = 100
```

Bind the components

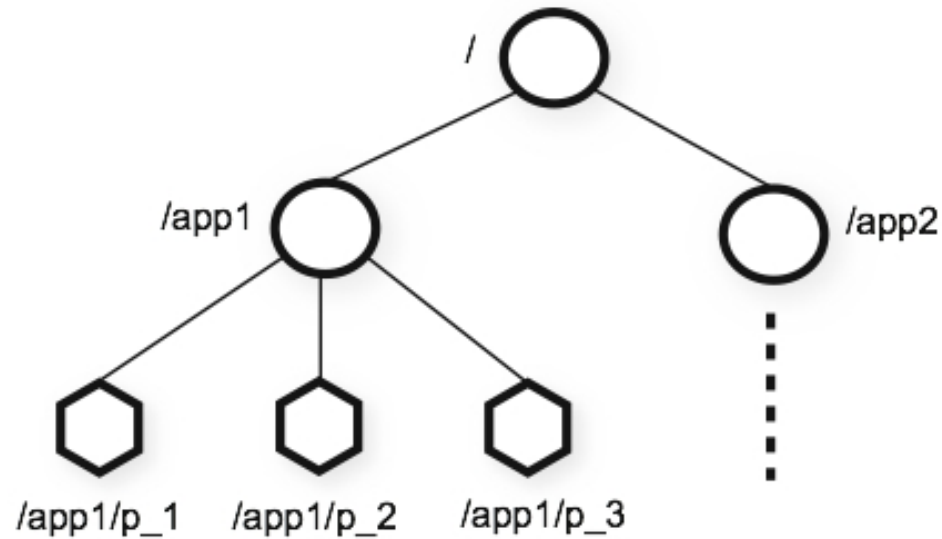
```
agent1.sources.src.channels = chl  
agent1.sinks.snk.channel = chl
```

Zookeeper



- Apache ZooKeeper, is a highly available, high performance coordination service
- Apache ZooKeeper provides operational services for a Hadoop cluster. ZooKeeper provides a distributed configuration service, a synchronization service and a naming registry for distributed systems.
- Challenges with distributed systems,
 - Coordination
 - Partial failures

Data Model

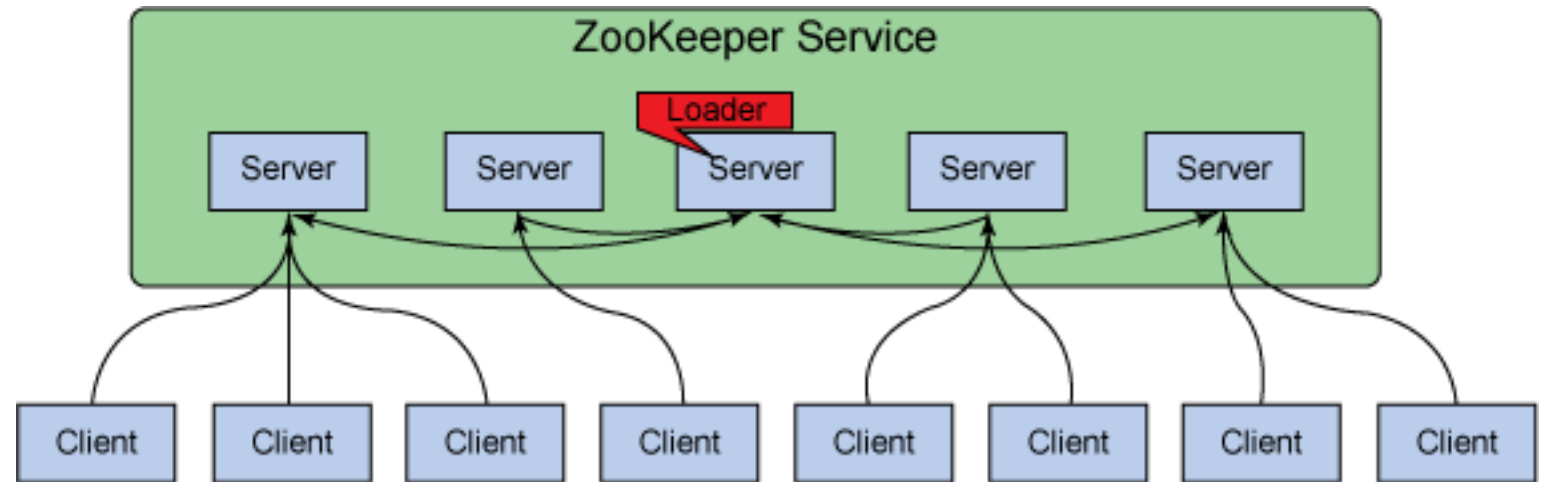


- Hierarchical namespace
- Each node in the namespace is called **znode**
- Every node in the namespace is identified by a path
- znode types – persistent and ephemeral
- Each znode has data and can also have children

Implementation

- Zookeeper service can run in two modes : standalone mode and replicated mode
- Implementation of replicated mode uses a protocol called **zab** that runs in two phases,
 1. Phase 1 : Leader Election
 2. Phase 2 : Atomic broadcast

Consistency



- Sequential consistency
- Atomicity
- Single system image
- Durability
- Timeliness

THANK YOU

