맛보기

Apache Avro

2011.9.29
1.5.4 기준

김용환
knight76.tistory.com

이름의 기원

- Avro는 영국 비행기 제조회사
- 1910년~1963년 (M&A되면서 사라짐)

함 해보기

설치

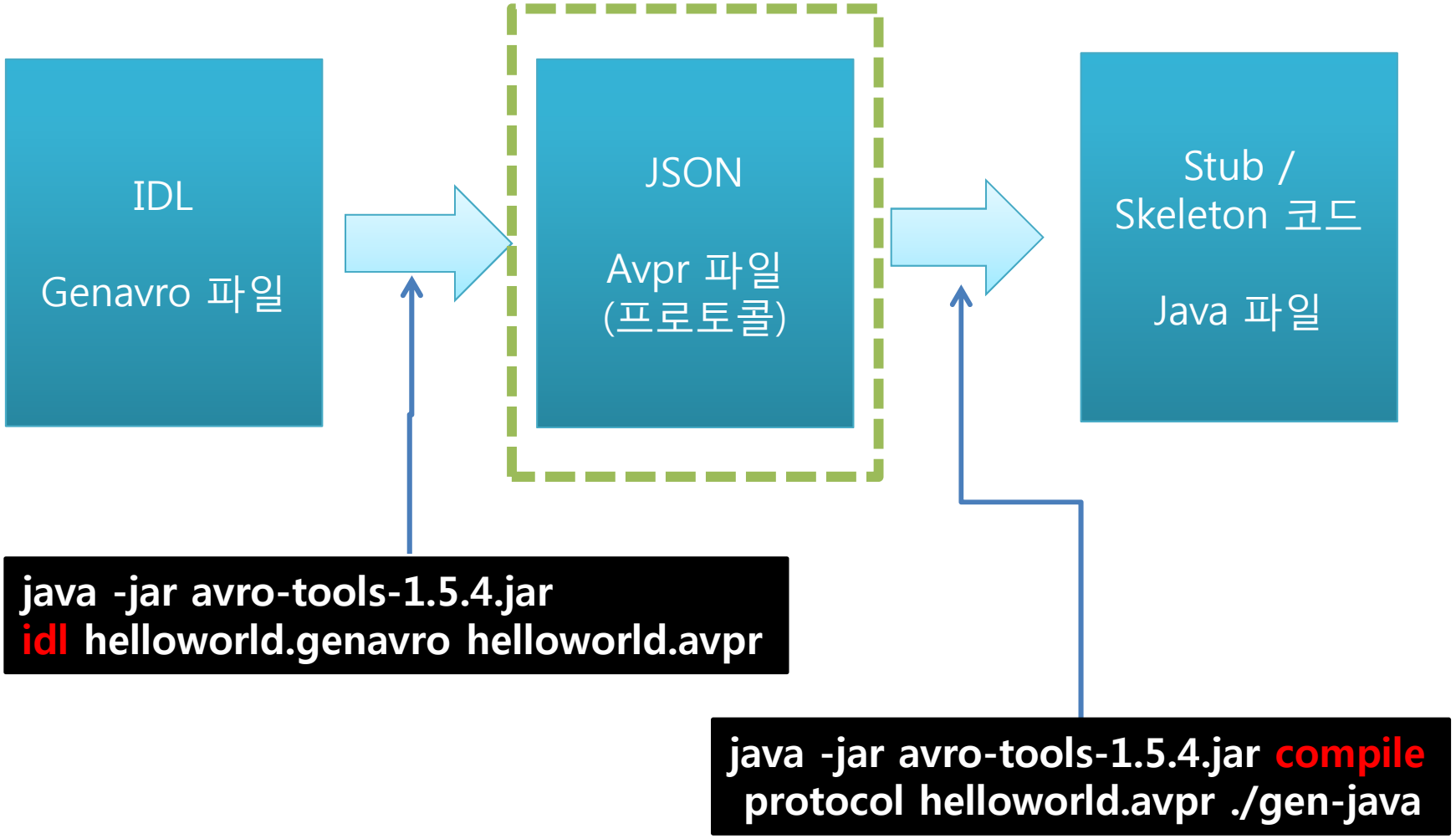
- 자바 버전은 소스 설치가 필요 없다.

http://ftp.daum.net/apache//avro/avro-1.5.4/java/

avro-mapred-1.5.4.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-maven-plugin-1.5.4-javadoc.jar	08-Sep-2011	06:44	46K	Java-Apache (old)
avro-maven-plugin-1.5.4-javadoc.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-maven-plugin-1.5.4-sources.jar	08-Sep-2011	06:44	14K	Java-Apache (old)
avro-maven-plugin-1.5.4-sources.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-maven-plugin-1.5.4.jar	08-Sep-2011	06:44	18K	Java-Apache (old)
avro-maven-plugin-1.5.4.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-tools-1.5.4-javadoc.jar	08-Sep-2011	06:44	97K	Java-Apache (old)
avro-tools-1.5.4-javadoc.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-tools-1.5.4-nodeps.jar	08-Sep-2011	06:44	44K	Java-Apache (old)
avro-tools-1.5.4-nodeps.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-tools-1.5.4-sources.jar	08-Sep-2011	06:44	31K	Java-Apache (old)
avro-tools-1.5.4-sources.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)
avro-tools-1.5.4.jar	08-Sep-2011	06:44	7.8M	Java-Apache (old)
avro-tools-1.5.4.jar.asc	08-Sep-2011	06:46	198	Java-Apache (old)

(소스 분석을 위해서는
http://ftp.daum.net/apache//avro/avro-1.5.4/avro-src-1.5.4.tar.gz 다운)

테스트 시나리오


IDL
Genavro 파일
JSON
Avpr 파일
(프로토콜)
Stub /
Skeleton 코드
Java 파일
java -jar avro-tools-1.5.4.jar
idl helloworld.genavro helloworld.avpr
java -jar avro-tools-1.5.4.jar compile
protocol helloworld.avpr ./gen-java

Genavro 파일

```
@namespace("test")
protocol HelloService {

        string hello(string greeting);

        record Result {
                int cmdNumber;
                string return;
        }

        error CmdException {
                string message;
        }

        Result cmd(int cmdNumber, string param) throws CmdException;
}
```

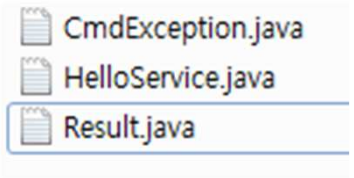
Avpr 파일
(java -jar avro-tools-1.5.4.jar idl helloworld.genavro helloworld.avpr)

```
{
  "protocol" : "HelloService",
  "namespace" : "test",
  "types" : [ {
    "type" : "record",
    "name" : "Result",
    "fields" : [ {
      "name" : "cmdNumber",
      "type" : "int"
    }, {
      "name" : "return",
      "type" : "string"
    } ]
  }, {
    "type" : "error",
    "name" : "CmdException",
    "fields" : [ {
      "name" : "message",
      "type" : "string"
    } ]
  } ],
```

```
"messages" : {
    "hello" : {
      "request" : [ {
        "name" : "greeting",
        "type" : "string"
      } ],
      "response" : "string"
    },
    "cmd" : {
      "request" : [ {
        "name" : "cmdNumber",
        "type" : "int"
      }, {
        "name" : "param",
        "type" : "string"
      } ],
      "response" : "Result",
      "errors" : [ "CmdException" ]
    }
  }
}
```

java 파일

java -jar avro-tools-1.5.4.jar compile
protocol helloworld.avpr ./gen-java



CmdException.java (C:₩a₩gen-java₩test) - GVIM1

```
/**
 * Autogenerated by Avro
 *
 * DO NOT EDIT DIRECTLY
 */
package test;
@SuppressWarnings("all")
public class CmdException extends org.apache.avro.specific.Specif
icExceptionBase implements org.apache.avro.specific.SpecificRecor
d {
  public static final org.apache.avro.Schema SCHEMA$ = org.apache
.avro.Schema.parse("{₩"type₩":₩"error₩",₩"name₩":₩"CmdException₩"
,₩"namespace₩":₩"test₩",₩"fields₩":[{₩"name₩":₩"message₩",₩"type₩
"":₩"string₩"}]}");
  public java.lang.CharSequence message;
  public org.apache.avro.Schema getSchema() { return SCHEMA$; }
  // Used by DatumWriter.  Applications should not call.
  public java.lang.Object get(int field$) {
    switch (field$) {
    case 0: return message;
    default: throw new org.apache.avro.AvroRuntimeException("Bad
index");
    }
  }
  // Used by DatumReader.  Applications should not call.
  @SuppressWarnings(value="unchecked")
  public void put(int field$, java.lang.Object value$) {
    switch (field$) {
    case 0: message = (java.lang.CharSequence)value$; break;
    default: throw new org.apache.avro.AvroRuntimeException("Bad
index");
    }
  }
}
```

27,1 모두

HelloService.java (C:₩a₩gen-java₩test) - GVIM2

```
/**
 * Autogenerated by Avro
 *
 * DO NOT EDIT DIRECTLY
 */
package test;

@SuppressWarnings("all")
public interface HelloService {
  public static final org.apache.avro.Protocol PROTOCOL = org.apa
che.avro.Protocol.parse("{₩"protocol₩":₩"HelloService₩",₩"namespa
ce₩":₩"test₩",₩"types₩":[{₩"type₩":₩"record₩",₩"name₩":₩"Result₩"
,₩"fields₩":[{₩"name₩":₩"cmdNumber₩",₩"type₩":₩"int₩"},{₩"name₩":
₩"return₩",₩"type₩":₩"string₩"}]},{₩"type₩":₩"error₩",₩"name₩":₩"
CmdException₩",₩"fields₩":[{₩"name₩":₩"message₩",₩"type₩":₩"strin
g₩"}]}],₩"messages₩":{₩"hello₩":{₩"request₩":[{₩"name₩":₩"greetin
g₩",₩"type₩":₩"string₩"}],₩"response₩":₩"string₩"},₩"cmd₩":{₩"req
uest₩":[{₩"name₩":₩"cmdNumber₩",₩"type₩":₩"int₩"},{₩"name₩":₩"par
am₩",₩"type₩":₩"string₩"}],₩"response₩":₩"Result₩",₩"errors₩":[₩"
CmdException₩"]}}}");
  java.lang.CharSequence hello(java.lang.CharSequence greeting) t
hrows org.apache.avro.AvroRemoteException;
  test.Result cmd(int cmdNumber, java.lang.CharSequence param) th
rows org.apache.avro.AvroRemoteException, test.CmdException;

  @SuppressWarnings("all")
  public interface Callback extends HelloService {
    public static final org.apache.avro.Protocol PROTOCOL = test.
HelloService.PROTOCOL;
    void hello(java.lang.CharSequence greeting, org.apache.avro.i
pc.Callback<java.lang.CharSequence> callback) throws java.io.IOEx
ception;
    void cmd(int cmdNumber, java.lang.CharSequence param, org.apa
che.avro.ipc.Callback<test.Result> callback) throws java.io.IOExc
eption;
  }
}
```

20,1 모두

Result.java (C:₩a₩gen-java₩test) - GVIM3

```
/**
 * Autogenerated by Avro
 *
 * DO NOT EDIT DIRECTLY
 */
package test;
@SuppressWarnings("all")
public class Result extends org.apache.avro.specific.Spec
ificRecordBase implements org.apache.avro.specific.Specif
icRecord {
  public static final org.apache.avro.Schema SCHEMA$ = or
g.apache.avro.Schema.parse("{₩"type₩":₩"record₩",₩"name₩"
:₩"Result₩",₩"namespace₩":₩"test₩",₩"fields₩":[{₩"name₩":
₩"cmdNumber₩",₩"type₩":₩"int₩"},{₩"name₩":₩"return₩",₩"ty
pe₩":₩"string₩"}]}");
  public int cmdNumber;
  public java.lang.CharSequence return$;
  public org.apache.avro.Schema getSchema() { return SCHE
MA$; }
  // Used by DatumWriter.  Applications should not call.
  public java.lang.Object get(int field$) {
    switch (field$) {
    case 0: return cmdNumber;
    case 1: return return$;
    default: throw new org.apache.avro.AvroRuntimeExcepti
on("Bad index");
    }
  }
  // Used by DatumReader.  Applications should not call.
  @SuppressWarnings(value="unchecked")
  public void put(int field$, java.lang.Object value$) {
    switch (field$) {
    case 0: cmdNumber = (java.lang.Integer)value$; break;
    case 1: return$ = (java.lang.CharSequence)value$; bre
ak;
    default: throw new org.apache.avro.AvroRuntimeExcepti
on("Bad index");
    }
  }
}
```

19,5 모두

HelloService.java

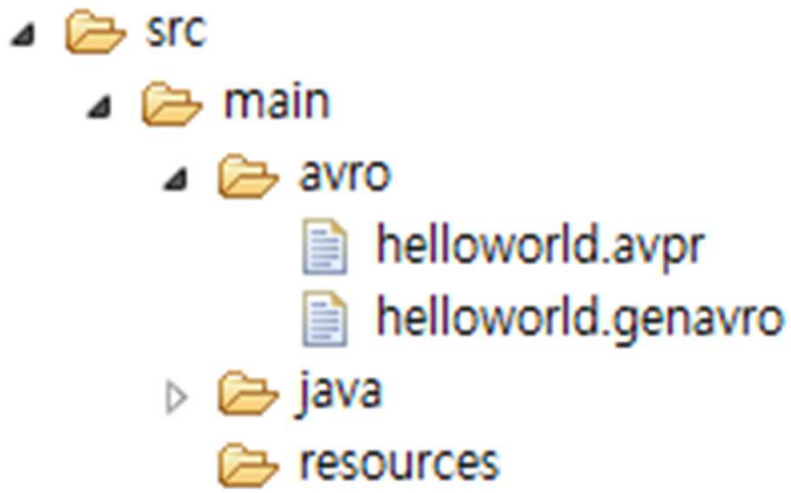
```
package test;

@SuppressWarnings("all")
public interface HelloService {
  public static final org.apache.avro.Protocol PROTOCOL =
org.apache.avro.Protocol.parse("{₩"protocol₩":₩"HelloService₩",₩"namespace₩":₩"test₩",₩"types
₩":[{₩"type₩":₩"record₩",₩"name₩":₩"Result₩",₩"fields₩":[{₩"name₩":₩"cmdNumber₩",₩"type
₩":₩"int₩"},{₩"name₩":₩"return₩",₩"type₩":₩"string₩"}]},{₩"type₩":₩"error₩",₩"name₩":₩"Cmd
Exception₩",₩"fields₩":[{₩"name₩":₩"message₩",₩"type₩":₩"string₩"}]}],₩"messages₩":{₩"hello
₩":{₩"request₩":[{₩"name₩":₩"greeting₩",₩"type₩":₩"string₩"}],₩"response₩":₩"string₩"},₩"cm
d₩":{₩"request₩":[{₩"name₩":₩"cmdNumber₩",₩"type₩":₩"int₩"},{₩"name₩":₩"param₩",₩"type
₩":₩"string₩"}],₩"response₩":₩"Result₩",₩"errors₩":[₩"CmdException₩"]}}}");

  java.lang.CharSequence hello(java.lang.CharSequence greeting) throws
org.apache.avro.AvroRemoteException;
  test.Result cmd(int cmdNumber, java.lang.CharSequence param) throws
org.apache.avro.AvroRemoteException, test.CmdException;

  @SuppressWarnings("all")
  public interface Callback extends HelloService {
    public static final org.apache.avro.Protocol PROTOCOL = test.HelloService.PROTOCOL;
    void hello(java.lang.CharSequence greeting,
org.apache.avro.ipc.Callback<java.lang.CharSequence> callback) throws java.io.IOException;
    void cmd(int cmdNumber, java.lang.CharSequence param,
org.apache.avro.ipc.Callback<test.Result> callback) throws java.io.IOException;
  }
```

Eclipse-maven 프로젝트생성

- genavro와 json으로 생성된 avpr 파일 생성
- pom.xml 파일에 java stub/skeleton 만드는 avro-maven-plugin plugin 추가

Eclipse

- src/main/avro 디렉토리에 avpr, genavro 파일 위치



Maven 프로젝트

- Pom.xml

```
<plugin>
<groupId>org.apache.avro</groupId>
<artifactId>avro-maven-plugin</artifactId>
<version>1.5.4</version>
<executions>
<execution>
<id>schemas</id>
<phase>generate-sources</phase>
<goals>
<goal>schema</goal>
<goal>protocol</goal>
<goal>idl-protocol</goal>
</goals>
<configuration>
<sourceDirectory>src/main/avro/</sourceDirectory>
<outputDirectory>src/main/java/</outputDirectory>
<testSourceDirectory>src/test/avro/</testSourceDirectory>
<testOutputDirectory>src/test/java/</testOutputDirectory>
</configuration>
</execution>
</executions>
</plugin>
```

Exception 발생시

```
C:₩a>java -jar avro-tools-1.5.4.jar idl helloworld.genavro helloworld.avpr
Exception in thread "main" org.apache.avro.compiler.idl.ParseException:
Undefine
d name '.String', at line 4, column 9
        at org.apache.avro.compiler.idl.Idl.error(Idl.java:48)
        at org.apache.avro.compiler.idl.Idl.ReferenceType(Idl.java:750)
        at org.apache.avro.compiler.idl.Idl.Type(Idl.java:670)
        at org.apache.avro.compiler.idl.Idl.ResultType(Idl.java:821)
        at org.apache.avro.compiler.idl.Idl.MessageDeclaration(Idl.java:565)
        at org.apache.avro.compiler.idl.Idl.ProtocolBody(Idl.java:342)
        at org.apache.avro.compiler.idl.Idl.ProtocolDeclaration(Idl.java:206)
        at org.apache.avro.compiler.idl.Idl.CompilationUnit(Idl.java:84)
        at org.apache.avro.tool.IdlTool.run(IdlTool.java:65)
        at org.apache.avro.tool.Main.run(Main.java:74)
        at org.apache.avro.tool.Main.main(Main.java:63)
```

Avro tool

- **compile** : Generates Java code for the given schema.
- fragtojson : Renders a binary-encoded Avro datum as JSON.
- fromjson : Reads JSON records and writes an Avro data file.
- **genavro** : Generates a JSON schema from a GenAvro file
- getschema : Prints out schema of an Avro data file.
- induce : Induce a schema/protocol from Java class/interface.
- jsontofrag : Renders a JSON-encoded Avro datum as binary.
- rpcreceive : Opens an HTTP RPC Server and listens for one message.
- rpcsend : Sends a single RPC message.
- tojson : Dumps an Avro data file as JSON, one record per line.

Features

특징

- a data serialization system.
  - Protocol buffer와 Thrift는 이기종간의 통신 솔루션인 데 반해, Avro는 serialization에 집중된 통신 솔루션.
- 특징
  - Schema language, cross language
  - Rich data structures.
  - A compact, fast, binary data format.
  - A container file, to store persistent data.
  - Remote procedure call (RPC).
    - 원래 없었는데. 1.3.0부터 지원 (ipc 패키지 추가됨)
  - Simple integration with dynamic languages.
- Language : C, C++, Java, PHP, Python, Ruby

특징

- Main developer and created by
  - Doug Cutting
- Doug Cutting said "**Avro will replace Hadoop's existing RPC**"
  - http://www.cloudera.com/blog/2009/11/avro-a-new-format-for-data-interchange/

특징

- Avro is replacing Thrift as the RPC client for interacting with Cassandra.
- Avro is a sub-project of the Apache Hadoop project
- dynamic data serialization library that has an advantage over Thrift in that it does not require static code generation.
- org.apache.cassandra.thrift.CassandraServer => org.apache.cassandra.avro.CassandraServer

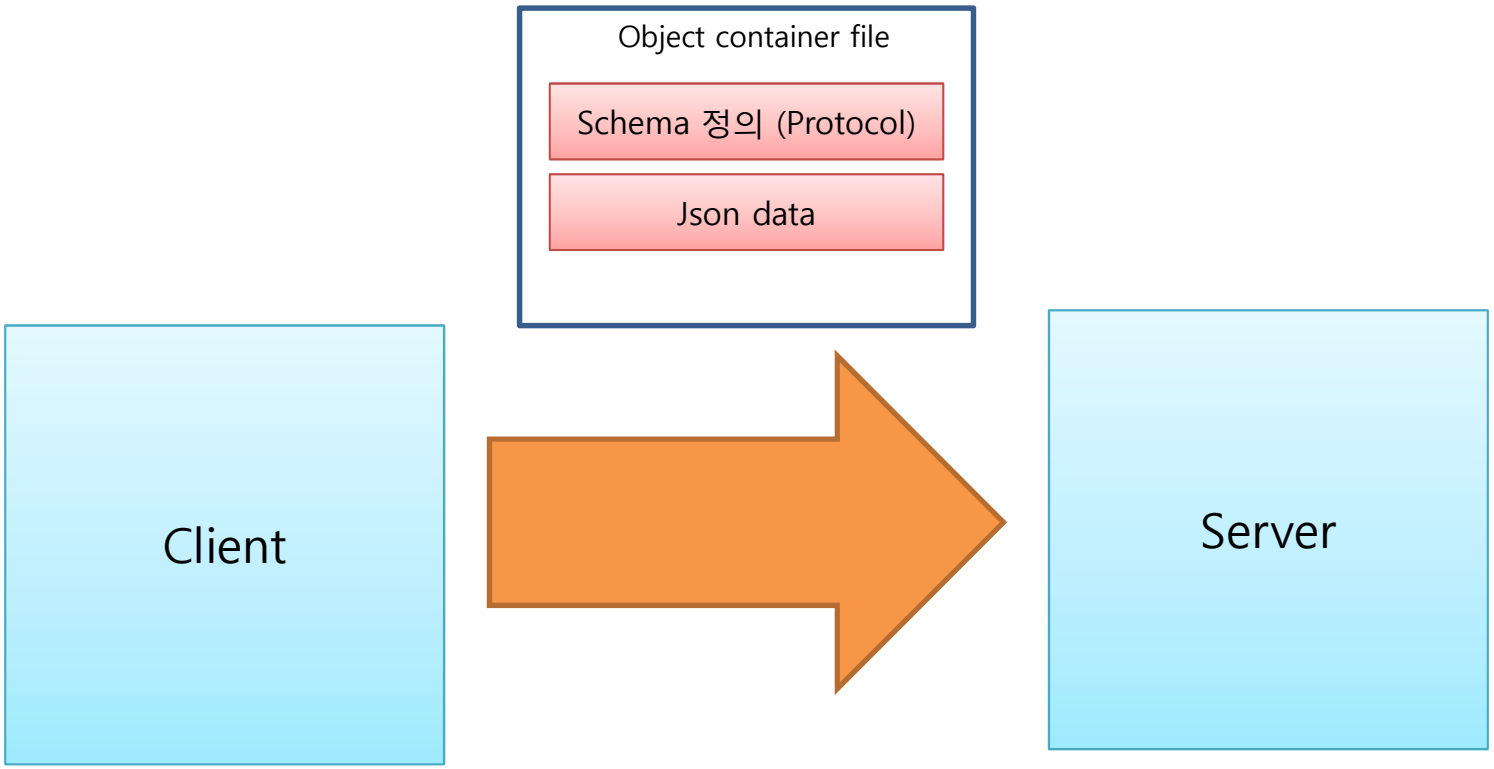
Type

- Primitive
  - string
  - bytes : byte 배열
  - int
  - long
  - float
  - double
  - Boolean
  - null
- Complex
  - record : struct 같은 개념
    - name : 레코드의 이름
    - doc : 스키마에 대한 설명
    - fields : 필드 정의, name, doc, type, default 속성 정의
  - array
    - Items
  - map
    - value
  - Union
    - name, size
  - fixed (고정길이)
  - enum
    - name, doc, symbols

Schema

- JSON 을 이용해서 구조를 정의 : Schema
- 전송 또는 저장 데이터의 포맷을 정의
- 전송시 스키마와 데이터를 차례로 전송, 저장시 파일 앞부분에 JSON형태로 스키마를 먼저 저장??
- 런타임 때, Schema 없이 파싱할 수 있음

데이터 전달



org.apache.avro.ipc

Interface Server

All Known Implementing Classes:

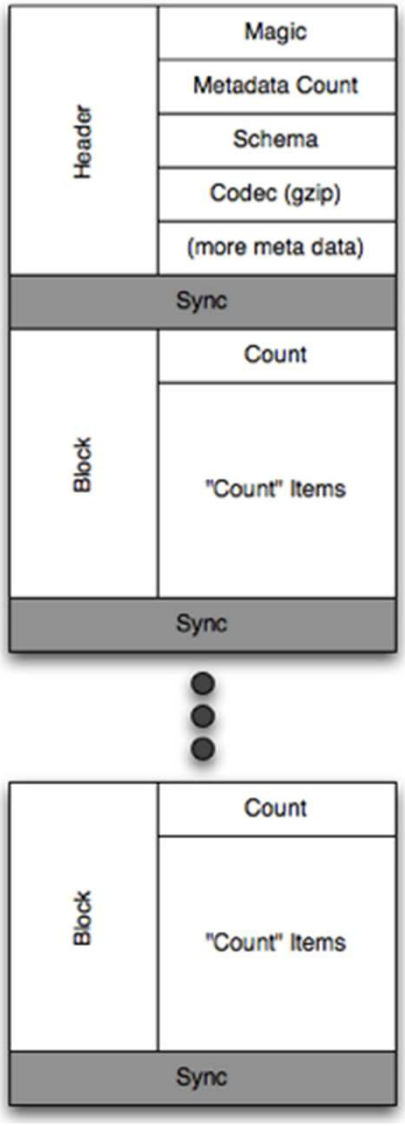
DatagramServer, HttpServer, NettyServer, SaslSocketServer, SocketServer

Object Container File

- Serialization의 비밀
- 데이터 스키마 정보와 데이터로 구성
- 데이터는 압축시켜 Block에 저장
- Sync Marker가 압축된 파일 중간에 추가

=> Hadoop Sequence File과 흡사한 구조

Object Container File



Hadoop Sequence File

- Hadoop sequence file

- Hadoop sequence file Header

Sequence File Header

4 bytes - 'SEQ' + VERSION
Text - Key Class Name
Text - Value Class Name
Boolean - Is Compressed
Boolean Is Block Compressed
Text - Compress Codec Class Name if (Is Compressed && version > CUSTOM_COMPRESS_VERSION)
Metadata (If version > VERSION_WITH_METADATA)
16 bytes - Sync (new UID() + '@' + time)

http://www.cloudera.com/blog/2011/01/hadoop-io-sequence-map-set-array-bloommap-files/

Hadoop Integration

1. Pig, Hive 와 같은 플랫폼에서 생성된 다양한 데이터를 처리 가능할 수 있음
2. Java api 한계를 극복, 다른 언어에서 포팅하고 구현할 수 있도록 함
3. 압축가능하고, 나눌 수 있는 구조
4. Shema는 sort order를 정의할 수 있음 ??

⇒Hadoop과 Avro가 쓰일 수 있도록 기대함

Apache avro api

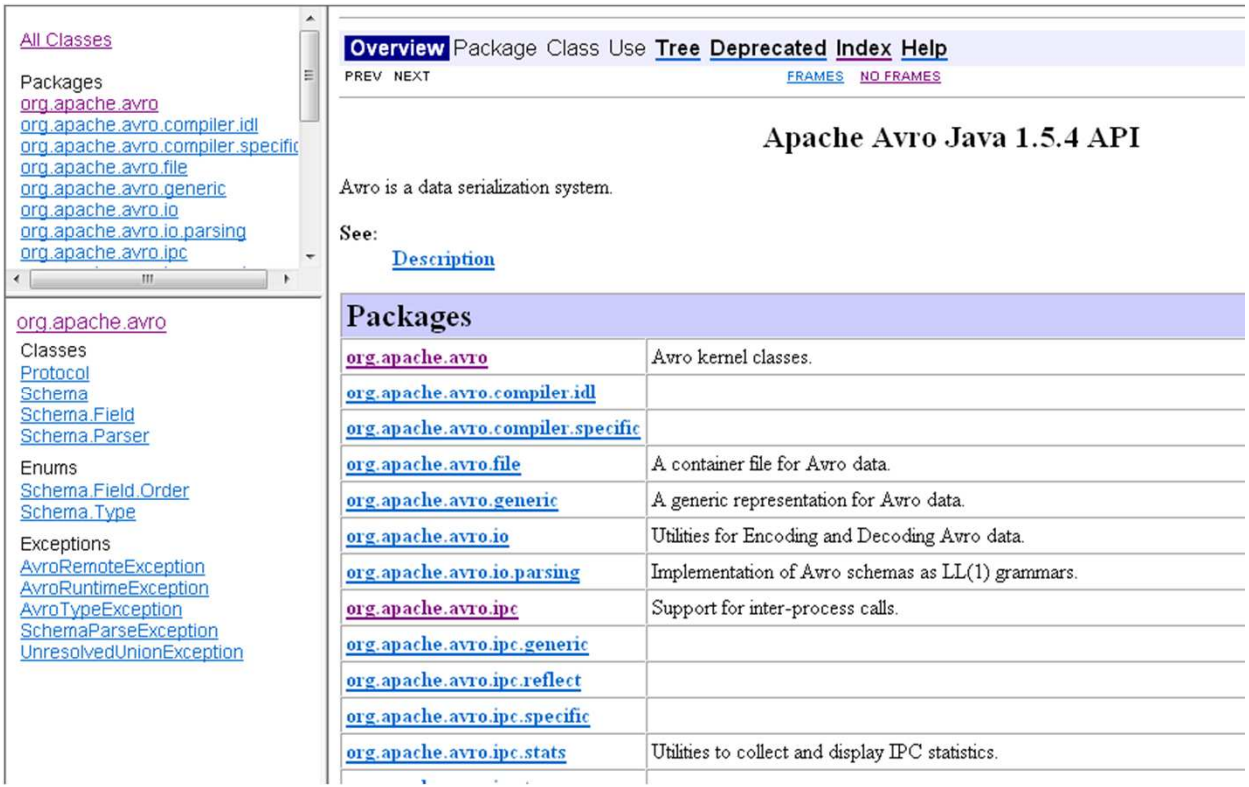
org.apache.avro.mapred	Run Hadoop MapReduce jobs over Avro data, with map and reduce functions written in Java.
org.apache.avro.mapred.tether	Run Hadoop MapReduce jobs over Avro data, with map and reduce functions run in a sub-process.

RPC

- Transports - currently only HTTP
- Handshake (exchange/verify protocols)
- Asynchronous/Synchronous

Java api

- http://avro.apache.org/docs/1.5.4/api/java/index.html



SocketServer Sample

```
public class Main {
    public static class MailImpl implements Mail {
        // in this simple example just return details of the message
        public Utf8 send(Message message) {
            return new Utf8("Sent message to " + message.to.toString()
                            + " from " + message.from.toString()
                            + " with body " + message.body.toString());
        }
      }

    private static SocketServer server;
    private static void startServer() throws IOException {
        // the server implements the Mail protocol (MailImpl)
        server = new SocketServer(new SpecificResponder(
                Mail.class, new MailImpl()), new InetSocketAddress(0));
    }

    public static void main(String[] args) throws IOException {
        if (args.length != 3) {
            System.out.println("Usage: <to> <from> <body>");
        }

        // usually this would be another app, but for simplicity
        startServer();

        // client code - attach to the server and send a message
        SocketTransceiver client =
            new SocketTransceiver(new InetSocketAddress(server.getPort()));
        Mail proxy = (Mail) SpecificRequestor.getClient(Mail.class, client);

        // fill in the Message record and send it
        Message message = new Message();
        message.to = new Utf8(args[0]);
        message.from = new Utf8(args[1]);
        message.body = new Utf8(args[2]);
        System.out.println("Result: " + proxy.send(message));

        // cleanup
        client.close();
        server.close();
    }
}
```

https://github.com/phunt/avro-rpc-quickstart/blob/master/src/main/java/example/Main.java

단점

- 버전이 바뀔 때마다 사용방법이 달라지거나 java api가 바뀌고 있는 중. (신버전 사용시 에러 발생있음)
- Json Schema를 사용하기 때문에 복잡한 데이터 구조 표현 방식의 어려움이 있을 수 있음.
  - Importing, polymorphism implementation, Nested extension…..

기대되는 점

- 웹 서버에 붙여서 바로 사용가능 (http/json)
- RpcSendTool/RpcReceiveTool 과 같이 검증 가능한 api 제공
  - 응답/요청에 대한 기대값을 테스트할 수 있음
- 많은 commiter를 흡수하려고 함

  예) Tracing api

Thrift, Protocol Buffer 비교

- Dynamic typing
  - Json으로 하다보니 serialization/deserialization에 대해서 통신 가능
- Untagged data
  - 데이터 인코딩을 compact하게 하기 위해서 data 정도에 대한 필드정보가 없음
- No manually-assigned field IDs
  - Thrift와 protocol buffer는 꼭 필요. (ordering이 중요)

끝