Apache Avro .NET

최흥배
https://github.com/jacking75/choiHeungbae

Apache Avro ?

- Thrift, ProtocolBuf와 비슷한 데이터 직렬화 라이브러리.
- 여기에 RPC 호출 과 파일에 데이터 저장하는 기능이 있음.
- 지원 언어
  C/C++, Java, PHP, Python, Ruby
  MS에서 직접 닷넷 버전 구현
- Hadoop의 RPC를 Avro RPC로 교체
- 풍부한 데이터 구조
- 작고 빠른 바이너리 타입 지원
- 컨테이너에 파일을 영구적으로 저장
- 기본적으로 바이너리 구조의 데이터와 Json 구조의 데이터 통신 Schema 지원
- Thrift에 비해 개발이 활발하고 카산드라의 내부 인터페이스로도 사용

Thrift, ProtocolBuf와 비교

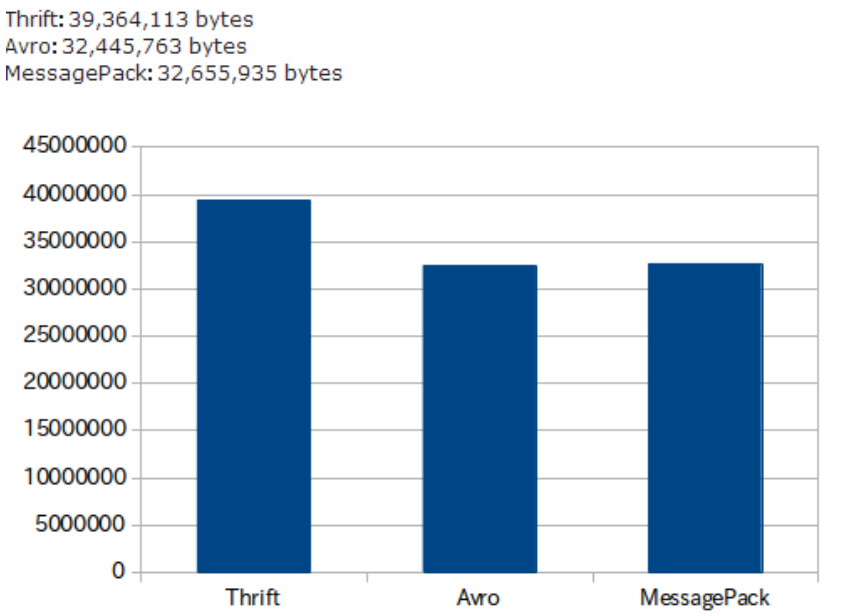
- Dynamic typing
  Json으로 하다보니 serialization/deserialization에 대해서 통신 가능

- Untagged data
  데이터 인코딩을 compact하게 하기 위해서 data 정도에 대한 필드정보가 없음

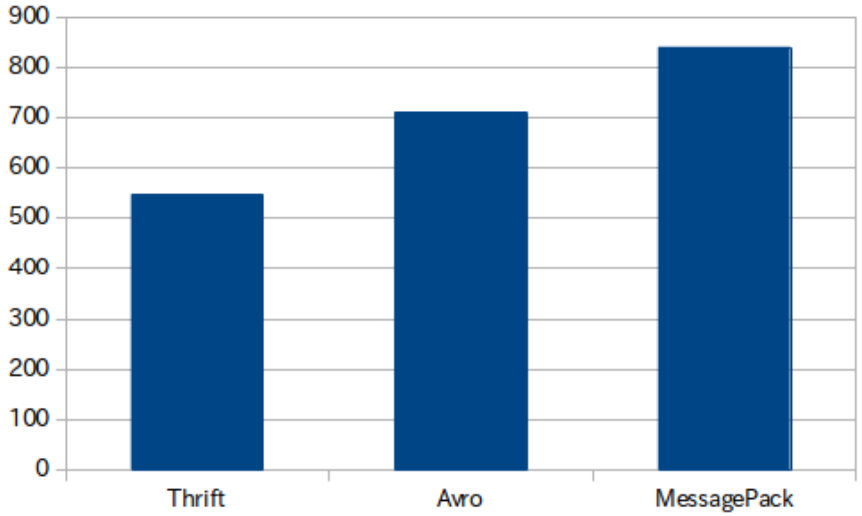
- No manually-assigned field IDs
  Thrift와 protocol buffer는 꼭 필요. (ordering이중요)

Thrift, Avro, MessagePack 성능 비교

- 자바를 기준으로 테스트
- 2014.03.01 테스트
  출처: http://symfoware.blog68.fc2.com/blog-entry-1292.html
  우편번호 약 12만건을 사용.
  소스: https://bitbucket.org/symfo/postgresql-map-sample/src



시리얼라이즈 용량 비교



시리얼라이즈 시간 비교(ms)

Avro 스키마 예

```
{"namespace": "org.jacob.avro",
 "type": "record",
 "name": "User",
 "fields": [
     {"name": "name", "type": "string"},
     {"name": "favorite_number",  "type": ["int", "null"]},
     {"name": "favorite_color", "type": ["string", "null"]}
 ]
}
```

GenAvro 파일로 스키마 파일 만들기

- GenAvro 파일을 이용하면 스키마 파일을 쉽게 만들 수 있다.
- 아래는 자바의 경우.
  http://blog.naver.com/cookatrice/50106992022

```
@namespace("avrotest")
protocol HelloService{
	record Greeting{
		string greetingMessage;
	}
	error GreetingException{
		string message;
	}
	Greeting hello(string greeting) throws GreetingException;
}
```

```
$java -jar avro-tools-1.4.1.jar idl hello.genavro hello.avpr
```

```
{
 "protocol" : "HelloService",
 "namespace" : "avrotest",
 "types" : [ {
  "type" : "record",
  "name" : "Greeting",
  "fields" : [ {
   "name" : "greetingMessage",
   "type" : "string"
  } ]
 }, {
  "type" : "error",
  "name" : "GreetingException",
  "fields" : [ {
   "name" : "message",
   "type" : "string"
  } ]
 } ],
 "messages" : {
  "hello" : {
   "request" : [ {
    "name" : "greeting",
    "type" : "string"
   } ],
   "response" : "Greeting",
   "errors" : [ "GreetingException" ]
  }
 }
}
```

Microsoftr Avro

- MS에서 직접 만든 Apache Avro의 닷넷 버전
- Azure용 Haddop인 Azure HDInsight를 위해 만든 것 같음.
- Apache Avro에 완벽하게 호환되고 고성능으로 만듬.
  단 RPC 기능은 지원하지 않음
- NuGet으로 설치 가능
  Install-Package Microsoft.Hadoop.Avro
- 공식 사이트
  https://hadoopsdk.codeplex.com/wikipage?title=Avro%20Library&referringTitle=Home
- 인메모리 식 트리를 구축으로 임의의 종류의 구조체를 시리얼화.
  인메모리 식 트리는 시리얼라이저가 네이티브한 성능을 발휘하도록 IL 코드로 컴파일 된다.

시리얼화 모드

- 리플렉션 모드
  시리얼라이저의 IL 코드는 .NET의 스키마 종류를 기초로 성능이 최대화 되도록 구축된다.

- 범용 레코드 모드
  데이터의 JSON 스키마는 실행시에 지정 할 수 있으므로 임의의 스키마로 동적 데이터 처리에 대응할 수 있다.

- 컨테이너 모드
  임베디드 스키마를 사용하여 포터블한 파일을 생성할 수 있다.
  파일 형식은 Avro 컨테이너 파일 사양과 호환성이 있으며 각종 플랫폼에서 사용할 수 있다.

예제: 리플렉션 모드

```
var avroSerializer = AvroSerializer.Create<SensorData>();
using (var buffer = new MemoryStream())
{
    var expected = new SensorData {
        Value = new byte[] { 1, 2, 3, 4, 5 },
        Position = new Location { Room = 243, Floor = 1 }
    };
    avroSerializer.Serialize(buffer, expected);
    buffer.Seek(0, SeekOrigin.Begin);
    var actual = avroSerializer.Deserialize(buffer);
    Assert.AreEqual(expected, actual);
}
```

http://blogs.msdn.com/resized-image.ashx/__size/426x0/__key/communityserver-blogs-components-weblogfiles/00-00-01-51-80/8372.001.png

참고

- Thrift & Avro RESEARCH
  http://koikebox.tistory.com/113
- Avro 1.7.6
  http://avro.apache.org/docs/current/spec.html
- Avro C++
  http://avro.apache.org/docs/current/api/cpp/html/index.html
- MS Avro API 레퍼런스 문서
  http://msdn.microsoft.com/en-us/library/azure/dn469975.aspx
- Overview 예제 코드
  http://code.msdn.microsoft.com/windowsazure/Serialize-data-with-the-86055923
- 커스텀 압축 코덱 사용 예제 코드
  http://code.msdn.microsoft.com/windowsazure/Serialize-data-with-the-67159111