

DataSet 연동(Java -->C#)방안 및 성능비교

Justin Song(mcsong@gmail.com)

1. 목적

기존에 사용하고 연동방식인 soap_http 방식과 soap_http 방식을 대체할 수 있는 json_http, json_http(rest), json_socket 방식에 대한 내용 및 성능비교를 통해서 WBC와의 연동 방안에 대한 가이드 제시를 목적으로 합니다.

2. 성능비교 및 분석

성능 비교는 Dual Core 2.53 Ghz, 2G Ram, Windows XP SP3에서 테스트를 진행하였습니다.

2.1 구간 및 데이터

2.1.1 구간

테스트 구간은 웹과 미들웨어 구간입니다. 웹 ↔ 미들웨어 ↔ DB의 구성에서 밑줄로 표시된 구간을 테스트 하였습니다.

2.1.2 데이터

```
DataSet ds = new DataSet();
DataTable dt = new DataTable();

dt.Columns.Add("serviceCode");
dt.Columns.Add("uuid");
dt.Columns.Add("product");
dt.Columns.Add("description");

for (int i = 0; i < 10; i++)
{
    DataRow row = dt.NewRow();
    row["serviceCode"] = "13";
    row["uuid"] = i + 1;
    row["product"] = "상품_" + i;
    row["description"] = "상품_" + i + "는 30일 정액제 입니다.";
    dt.Rows.Add(row);
}
ds.Tables.Add(dt);
```

2.2 성능비교

코드에서 본 것처럼, DataSet을 List<PurchaseModel>로 변환해서, Java 클라이언트에서 각각의 방식에 따

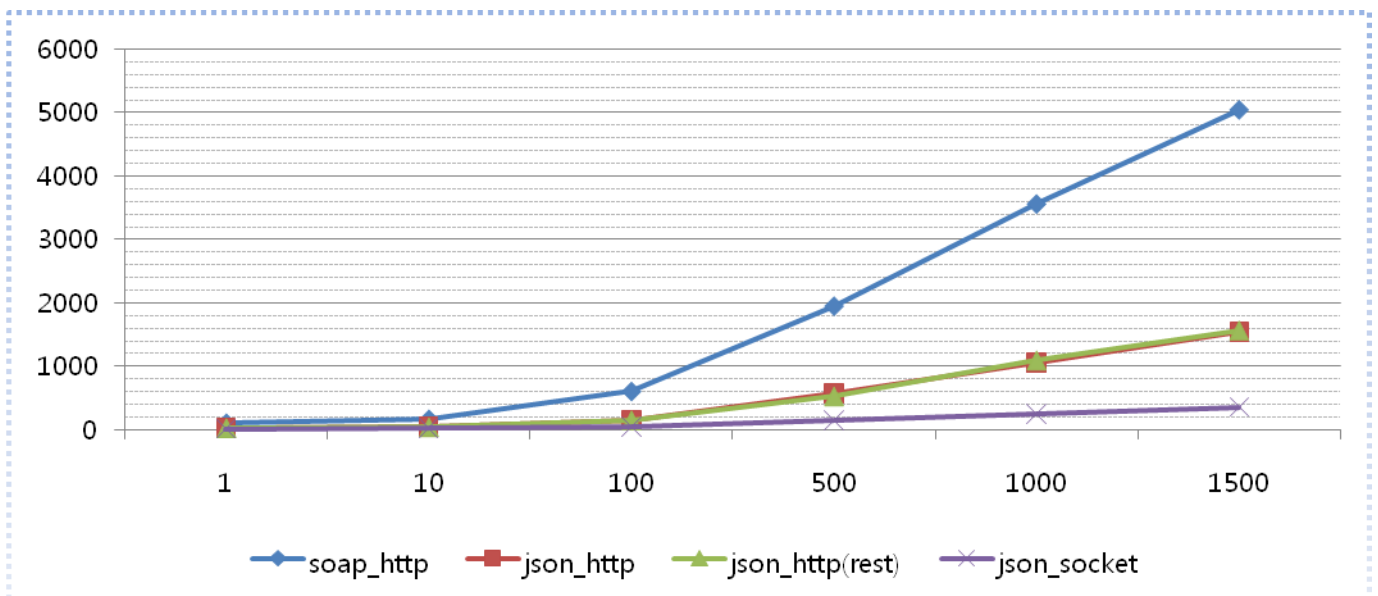
른 처리결과에 대한 response time을 기록하였습니다.

2.2.1 1개의 쓰레드로 처리한 결과

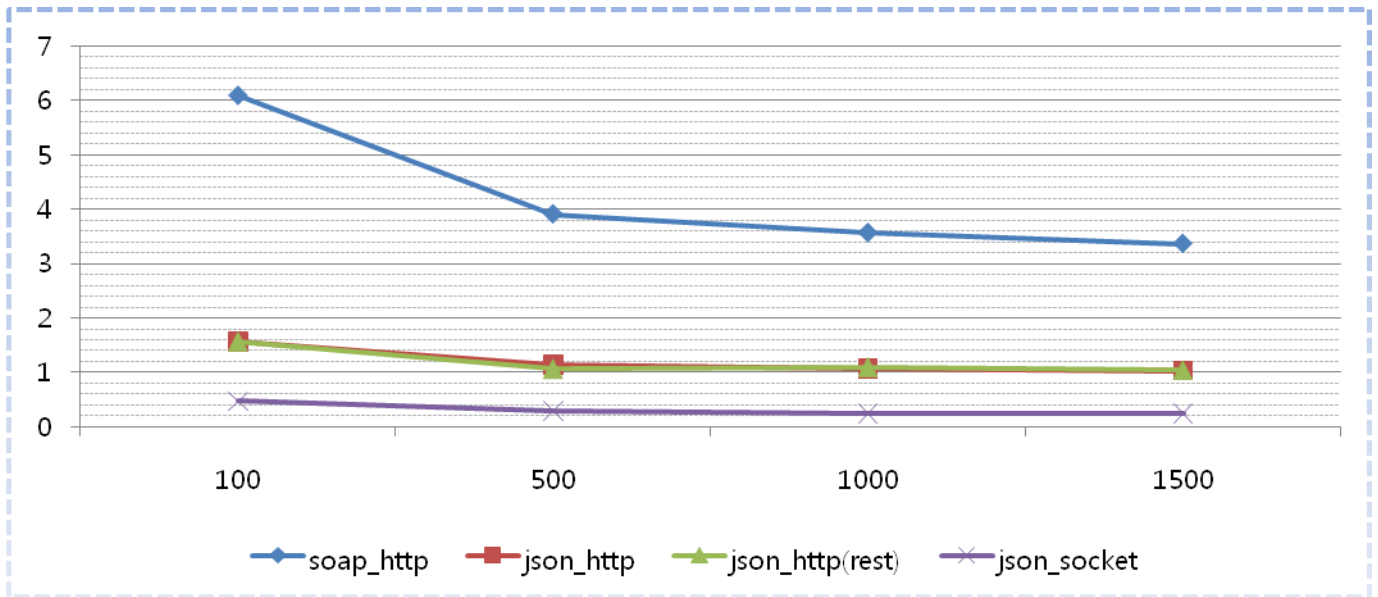
1개의 쓰레드로 아래의 여러가지의 방식을 통해서 저장된 Queue의 작업을 처리한 결과에 대한 내용입니다.

	soap_http	json_http	json_http(rest)	json_socket
1	110	31	31	16
10	172	47	47	31
100	609	156	156	47
500	1954	571	531	148
1000	3563	1063	1094	250
1500	5046	1547	1562	359

[표-1, 1개의 쓰레드로 왼쪽의 Queue에 저장된 Task를 처리한 결과]



[그래프-1, 표-1 데이터 그래프]



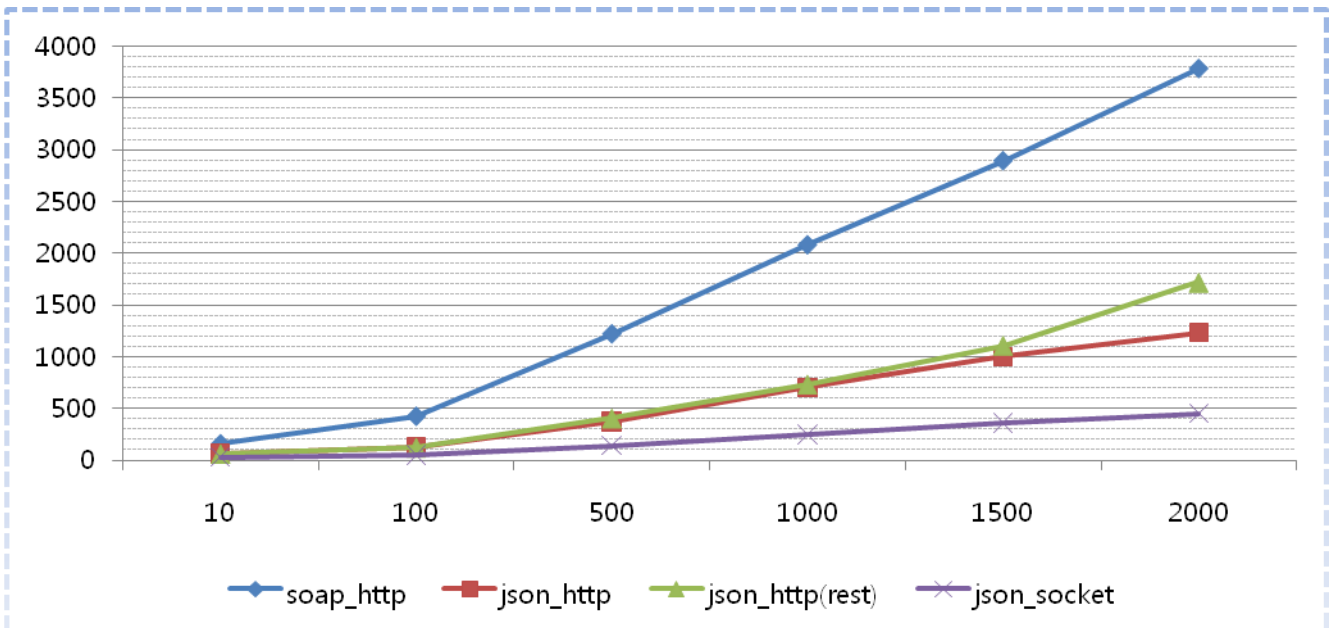
[그래프-2, 표-1 데이터의 평균 응답시간 그래프]

2.2.2 10개의 쓰레드로 성능비교

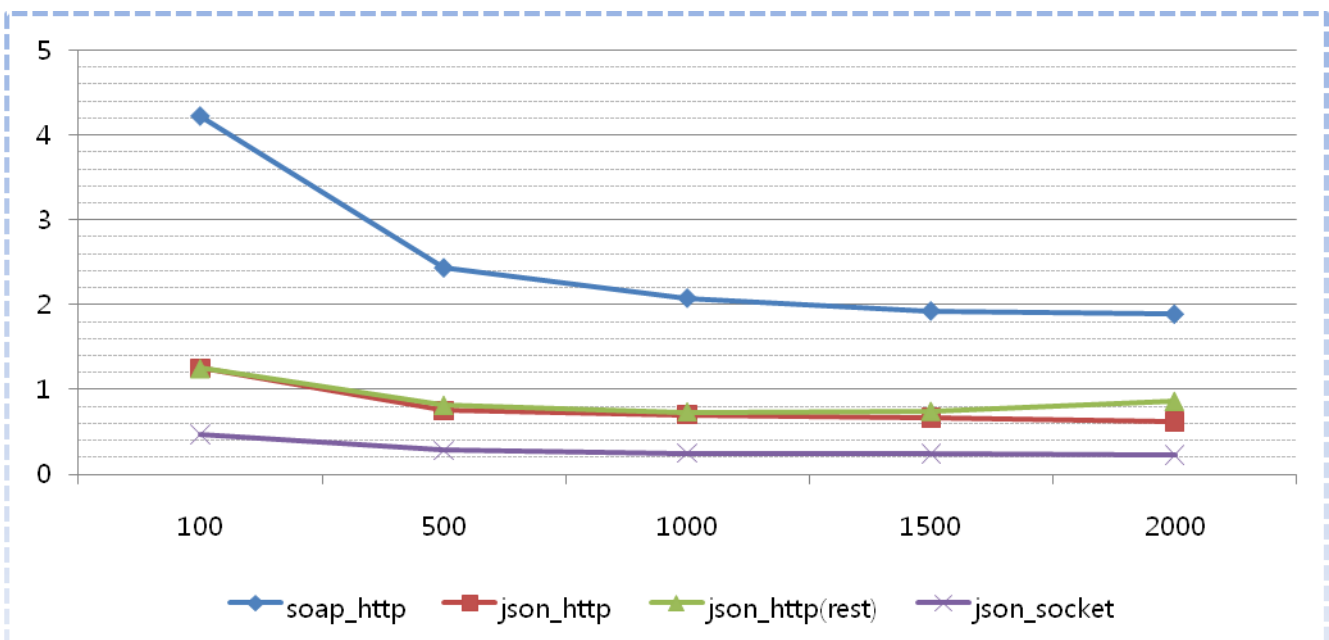
10개의 쓰레드로 아래의 여러가지의 방식을 통해서 저장된 Queue의 작업을 처리한 결과에 대한 내용입니다.

	soap_http	json_http	json_http(rest)	json_socket
10	156	63	63	31
100	422	125	125	47
500	1218	375	406	141
1000	2078	703	734	250
1500	2891	1000	1110	360
2000	3781	1234	1719	453

[표-2, 10개의 쓰레드로 왼쪽의 Queue에 저장된 Task를 처리한 결과]



[그래프-3, 표-2 데이터 그래프]



[그래프-4, 표-2 데이터의 평균 응답시간 그래프]

2.3 분석

분석은 위에서 테스트한 결과에 대한 업무로드 및 성능 그리고 방향에 대한 내용입니다.

2.3.1 업무로드

위의 예제코드로 추정을 해 본 업무로드에 해당하는 내용은 아래와 같습니다.

- soap_http <= json_http < json_socket

2.3.2 성능분석

위의 성능분석 결과는 아래와 같습니다.
 - json_socket > json_http > soap_http

2.3.3 추후 방향

위의 업무로드와 성능의 절충안으로 json_http 또는 json_http(rest) 방식이 좋을 듯 하며, 추후 성능을 고려하면, json_socket이 좋은 방향이 될 것으로 생각합니다. 하지만, 기존 서비스 인터페이스(soap_http) 방식외의 방식이 추가됨으로 인해서 2가지 이상의 인터페이스 유지를 위한 비용발생도 중요한 이슈라고 생각합니다.

3. 데이터 포맷 별 사이즈 비교

네트워크를 통해 데이터를 전송하기 위한 방법이 있고, 아래에서는 언어에서 제공하는 방법(object serialize/deserialize), soap, json 포맷 데이터의 사이즈 비교를 하였습니다.

예) PurchaseModel object Lists

```
for(int i=0; i<10; i++)
    model.addModel(new PurchaseModel(13, 120, "상품_" + i, "상품_"+i+"는 30일 정액제 입니다."));
```

서버 : c# dataset, c# list
 클라이언트 : java arraylist
 리턴결과 : json, soap

java arraylist	c# list	c# dataset	soap	json
1089	1830	3864	4689	1042

