

# Using LoadRunner for Unsupported or Difficult to Parameterize Protocols

**Alexander Podelko and Arno Sock**

**Hyperion Solutions**

**Alexander\_Podelko@hyperion.com**

**Arno\_Sock@hyperion.com**

# Agenda

- **Performance Testing @ Hyperion**
- **Problems**
- **Alternatives**
- **The Solution**
- **If Difficult to Parameterize...**
- **Pros and Cons**

# Hyperion Solutions

- **Presentation is based on Hyperion performance team experience**
- **Hyperion Solutions is a vendor of Business Performance Management software**
  - **Revenues of \$622 million in fiscal 2004**

# Performance Testing at Hyperion

- **Centralized Performance Engineering Group was created in 1997**
- **Lab environment and customer sites**
- **Numerous products and configurations**

# **“Record and Playback”**

- **Virtual users: Record communication between two tiers and then playback an automatically created script**
- **We successfully used this approach in most project since 1997**

# Problems

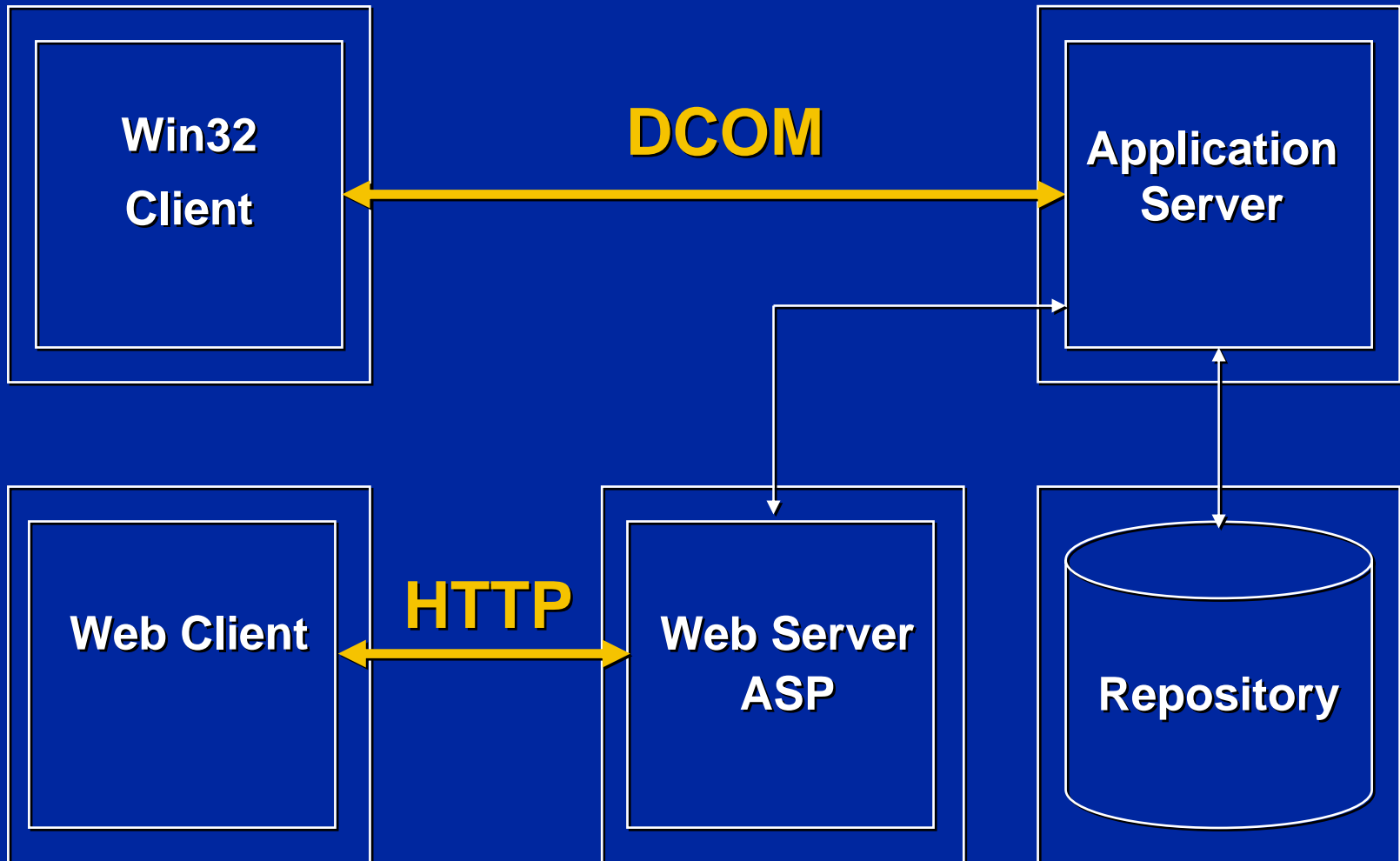
- **"Record and playback" approach often doesn't work for testing components**
- **LoadRunner supports a limited number of technologies (protocols)**
- **We had several problems back in 1999**

# Hyperion Enterprise

Proven Financial Consolidation, Reporting,  
and Analysis



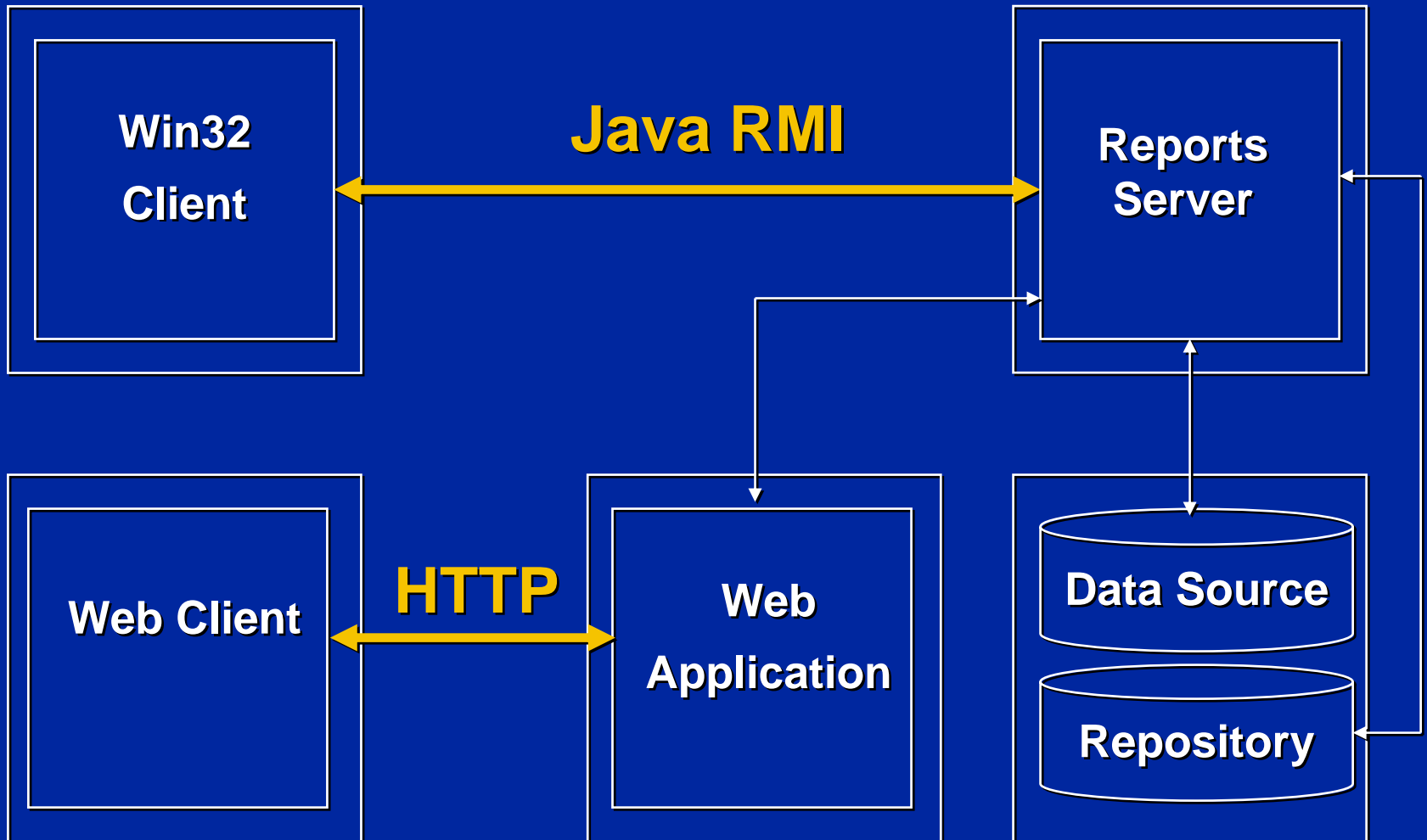
# Hyperion Financial Management



\*All brands and trademarks are the property of their owners



# Hyperion Reports



\*All brands and trademarks are the property of their owners

# GUI Users

- **WinRunner**
- **Record and playback communication between user and client GUI**
- **Don't care about communication protocols/ internals**
- **Requires a real machine for each user**

# Custom Test Harness

- **Special program to generate workload**
- **Requires access to the API or source code**
- **Requires programming**
- **Could be cost effective solution in some simple cases**

# Advantages

- **Doesn't require any special tool**
- **Starting version could be quickly created by a programmer familiar with API**
- **Should work if API works**
- **You don't care what protocol is used for communication**

# Disadvantages

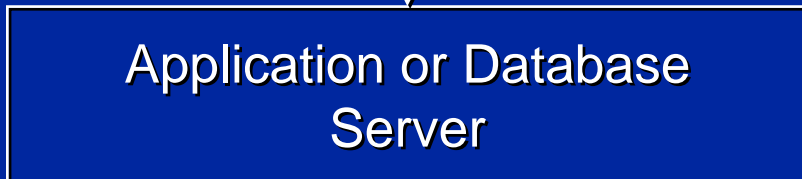
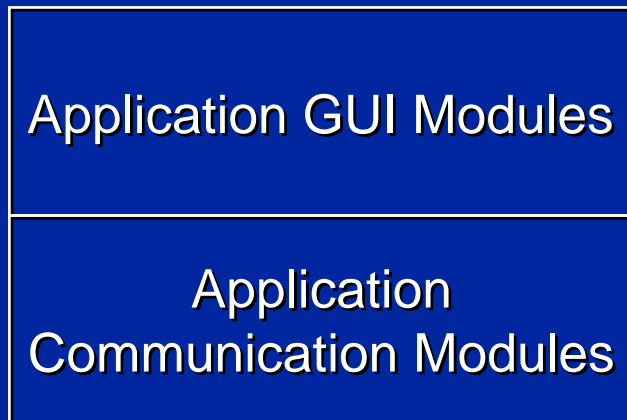
- **Efforts to update and maintain harness can increase drastically**
- **When you have numerous products you really need to create something like a commercial load testing tool**

# Custom Load Generation

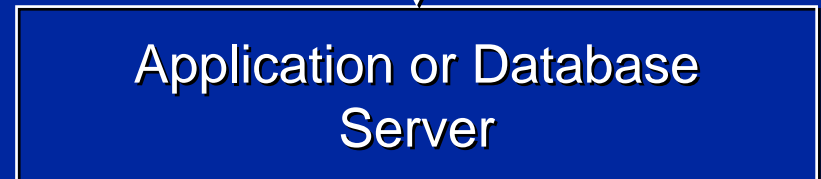
- **Mixed approach**
  - **Lightweight custom client stubs to work with an application**
  - **LoadRunner to manage these stubs and analyze results**
- **There are several ways to implement it in LoadRunner:**
  - **External DLL**
  - **C, Java, VB, VB Script, JavaScript custom Vuser scripts**

# Custom Load Generation

Client PC



Load Generation PC



# Implementation with External DLL

- To use the external dll in Vuser script LoadRunner could
  - Use the `lr_load_dll` function to load it
  - Define it globally in the `vugen.dat` file
- Then functions defined in the dll could be used without declaration in the script
- DLL (and environment) should be set on each agent machine



# Hyperion Enterprise DLL

```
extern "C" {  
    _declspec(dllexport) int appLogin (char * appName, char  
    * usrName, char * usrPassword);  
    _declspec(dllexport) int loadApp (char* lpszCategory,  
    char* lpszLoadFile);  
    ...  
    int appLogin (char *appName, char * usrName, char *  
        usrPassword) {  
        short FAR * pwRet = new short;  
        hApp = EntOpenApplication (appName, usrName,  
        usrPassword, pwRet );  
        int rc = (int) *pwRet;  
        delete pwRet;  
        return rc; }  
    ...  
}
```

# Hyperion Enterprise Script

```
Action1() {  
char *pszAppName;  
char *pszUserId;  
char *pszPassword;  
  
...  
IRetVal =  
    lr_load_dll("d:\\Ent\\retdll\\Release\\retdll.dll");  
pszUserId = lr_eval_string("user{MyUID}");  
lr_start_transaction("OpenApp_dv");  
IRetVal = appLogin (pszAppName, pszUserId,  
    pszPassword);  
lr_end_transaction("OpenApp_dv", LR_AUTO);  
if (IRetVal != 0 ) lr_output_message("RetVal is: %d\n",  
    IRetVal);  
  
...  
}
```

# Implementation with Java Vuser Script

- It could be difficult to use external DLLs from other languages
- LoadRunner script can be in Java
- Requires license for a Java-base protocol

# Hyperion Reports Script

```
import Irapi.Ir;  
import com.hyperion.reporting.job.*;  
  
...  
public class Actions { ...  
    public int action() { ...  
        try {  
            ...  
            DesignerManager dm = new DesignerManager();  
            String sReportServer = getProperty(sPropFile,  
"ReportServer");  
            Ir.start_transaction("takeConnection");  
            takeConnection(sReportServer,"reportserver");  
            Ir.end_transaction("takeConnection", Ir.AUTO);  
            ...  
        }  
    }  
}
```

# If Difficult to Parameterize...

- Recording and parameterization of a script can be time-consuming
- “Custom load generation” approach sometimes can be a better choice

# Example 1: Essbase Query

- **Multi-dimensional database**
- **Application manager and Excel add-in—  
Winsock scripts**
- **Quite difficult to parameterize and verify**
- **External DLL was made for major functions**

# Winsock Script

```
lrs_create_socket("socket0", "TCP", "LocalHost=0",  
"RemoteHost=ess001.hyperion.com:1423",  
LrsLastArg);  
lrs_send("socket0", "buf0", LrsLastArg);  
lrs_receive("socket0", "buf1", LrsLastArg);  
lrs_send("socket0", "buf2", LrsLastArg);  
lrs_receive("socket0", "buf3", LrsLastArg);  
lrs_save_searched_string("socket0",  
    LRS_LAST_RECEIVED, "Handle1",  
    "LB/BIN=\\x00\\x00\\v\\x00\\x04\\x00",  
    "RB/BIN=\\x04\\x00\\x06\\x00\\x06", 1, 0, -1);  
lrs_send("socket0", "buf4", LrsLastArg);  
lrs_receive("socket0", "buf5", LrsLastArg);  
lrs_close_socket("socket0");
```

# Winsock Script

```
send buf22 26165
```

```
"\xff\x00\xff0\a"
```

```
"\x00\x00\x00\x00\x01\x00\x00\x00\x01\x00\x03\x00"
```

```
"d\x00\b\x00"
```

```
"y'<Handle1>\x00"
```

```
"\b\r\x00\x06\x00\f\x00\x1be\x00\x00\r\x00\xd6\aRN"
```

```
"\x1a\x00\x06\x00\x00\x00\x00\x00\x00\x00\x00\b"
```

```
"\x00\x00\x00\xe7\x00\x00\x01\x00\x03\x00\x04\x00"
```

```
"\x10\x00\xcc\x04\x05\x00\x04\x00\x80\xd0\x05\x00\t"
```

```
"\x00\x02\x00\x02\x00\b\x00<\x00\x04"
```

```
"FY04\aWorking\tYearTotal\tELEMENT-F\tProduct-P"
```

```
"\x10<entity>\t\x00\x02\x00"
```

...



# Script Using External DLL

```
lr_load_dll("c:\\temp\\lr_msas2k.dll");  
pCTX = Init_Context();  
hr = Connect(pCTX, "ess01", "user001", "password");  
...  
lr_start_transaction("Mdx_q1");  
sprintf(report, "SELECT %s.children on columns,  
%s.children on rows FROM Shipment WHERE  
([Measures].[Qty Shipped], %s, %s)",  
lr_eval_string("{day}"), lr_eval_string("{product}"),  
lr_eval_string("{customer}"),  
lr_eval_string("{shipper}"));  
hr = RunQuery(pCTX, report);  
lr_end_transaction("Mdx_q1", LR_AUTO);
```

# Example 2: EDS

- **Essbase Deployment Services**
- **Middleware, no GUI interface**
- **Test scripts in Java from the QA group**
- **Solution: Creation of LoadRunner scripts from the test script**

# EDS Java Script

```
import Irapi.Ir;  
import com.essbase.api.base.*;  
import com.essbase.api.session.*;  
...  
public class Actions{  
    public int init() {  
        return 0;  
    }  
    }//end of init  
    public int action() {  
        String s_userName = "system";  
        String s_password = "password";
```

# EDS Java Script

```
lr.enable_redirection(true);
```

```
try {
```

```
    lr.start_transaction("01_Create_API_instance");
```

```
        ess =  
        IEssbase.Home.create(IEssbase.JAPI_VERSION);
```

```
    lr.end_transaction("01_Create_API_instance",  
    lr.AUTO);
```

```
    lr.start_transaction("02_SignOn");
```

```
        IEssDomain dom = ess.signOn(s_userName,  
s_password, s_domainName, s_prefEesSvrName,  
s_orbType, s_port);
```

```
    lr.end_transaction("02_SignOn", lr.AUTO);
```

# Advantages

- **Eliminates dependence on supporting specific protocols**
- **Leverages all the features of LoadRunner and uses it as a test harness**
- **Sometimes simplifies work with difficult-to-parameterize protocols**

# Considerations

- **Requires access to API or source code**
- **Requires programming**
- **Minimal transaction that could be measured is an external function**
- **Requires understanding of internals**

# Recording vs. API

- RMI recording

```
_integer =  
    _ireportserver.executeJob(_designjobobject);  
_ireportserver.getStatus(new Integer(3));  
_ireportserver.getStatus(new Integer(3));  
_ireportserver.getStatus(new Integer(3));  
_iinstance = _ireportserver.getInstance(new Integer(3));
```

- Real code

```
joid = poReportServer.executeJob(djo);  
bStatus = true;  
while (bStatus) {  
    bStatus = poReportServer.getStatus(joid);  
    Thread.sleep(300); }  
poReportServer.getInstance(joid);
```

# More Considerations

- **Requires a LoadRunner license for the necessary number of virtual users**
- **Environment should be set on all agents**
- **Usually requires more resources on agent machines**
- **Results should be cautiously interpreted**



# Summary

- **LoadRunner is “Optimized for Growth” platform for the Hyperion performance group**
- **Excellent record/playback support for many protocols**
- **“Custom load generation” approach complements it for unsupported and difficult-to-parameterize protocols**

# Questions?

**Alexander Podelko**

**Arno Sock**

**Hyperion Solutions**

**Alexander\_Podelko@hyperion.com**

**Arno\_Sock@hyperion.com**