

PBUGG.de

PowerBuilder User Group Germany e.V.

Appeon Performance Tuning Best Practices

MEONI Marco

November 14th, 2017

APPEON[®]

©2017 Appeon Limited and its subsidiaries. All rights reserved.

Apppeon to Apppeon (Web) Feature list



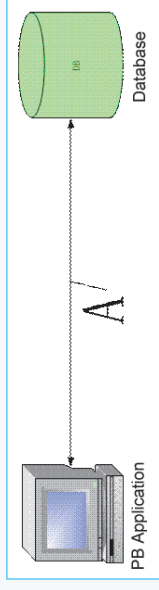
- Windows, UI Controls, VO/NVO
- System Objects, PowerScript
- Windows Registry, File System
- Local devices (printer, scanner, barcode...)
- Win32 DLL, OLE/OCX, PBDOM, RTF
- DW/DS, SPs, Trigger, dynamic SQL
- Web Service
- .NET Assembly, File Service

} JS WebLibrary

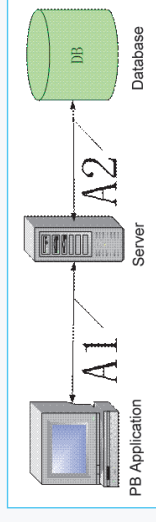
} Apppeon Plugin

} Apppeon Server

PB vs Appeon Performance



- Connection is LAN or high speed enterprise network
- Connection hardly impacts performance
- Performance mostly affected by Application + DB



- Server holds BL and data service
- Client interacts with DB through server using Internet protocol
- A1 is usually WAN/Internet
- A2 is high speed network
- Performance also affected by Server + A1

Traditional PB app architecture

Appeon Web/Mobile app architecture

Insights from a (very) large migration project

- 76 legacy PB applications
- 2.3 GB of source code (copy&paste not included 😊)
 - 13k windows, 18k DWs, 2k UOs, 223k methods, 162k events
- 60000 end-users (30000 concurrent)
- DTAP sw cycle
 - Development: 2 Win2008 R2 servers + 10 PB workstations
 - Test: IBM Pure Appl. System, 30 Appeon Web VMs, 1/8-1 vcores, 2 GB RAM
 - Acceptance & Production
 - IBM Pure Appl. Systems, 30 Appeon Web VMs (1/8 - 4 vcores, 8 GB RAM)

“The Appeon application is slow”

What are the reasons of Slow Code?

Server Requests

JavaScript

Apppeon Data Management

- Apppeon data handling is largely optimized and does not have any noticeable impact on the performance of Web/Mobile apps
 - Web files (html, JS, xml) are automatically cached "as needed"
 - Built-in 10X data compression for DW result sets to reduce time spent downloading them
 - DW caching (for static data): buffer data files in the server memory or in the browser cache

Appeon Data Management (cont)

- Connection Pooling
 - Cache of DB connections that can be reused when future requests to the DB are required
 - Enhance performance of executing commands on a DB
 - After a connection is created, it is placed in the pool and used again
 - If all connections are being used, a new one is made and added to the pool
 - Connection pooling cuts down waiting time to establish a DB connection
- Appeon performance on data retrieval is frequently faster than PB

Server Requests

- Server Call
 - Any code that results in a HTTP call when executed multiple times sequentially has potential to create network chatter
 - Data Access (Embedded/Dynamic SQL, SPs, DW/DS functions & events)
 - RMI (PB NVO, Java EJB, .NET Component) and WS

Runtime Performance (cont)

- Appeon provides labels to group server calls

```
dw_1.Retrieve()  
dw_2.Retrieve(ll_parm)  
dw_3.Update()
```

3x faster

```
appeon_nvo_db_update inv_appeondb  
inv_appeondb.of_startqueue( )  
  
dw_1.Retrieve()  
dw_2.Retrieve()  
dw_3.Update()  
  
inv_appeondb.of_commitqueue( )
```

```
if dw_1.update() then  
  if dw_2.update() then  
    if dw_3.update() then  
      commit;  
    else  
      rollback;  
  else  
    rollback;  
else  
  rollback;  
end if
```

3x faster

```
appeon_nvo_db_update inv_appeondb  
inv_appeondb.of_update(dw_1,dw_2,dw_3)
```

Javascript



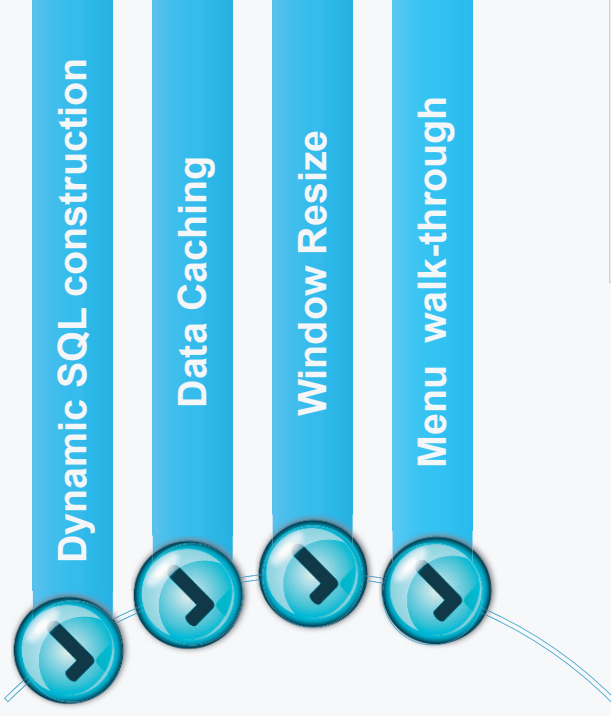
APPEON

Each line of PB code is *compiled* into binary code

PBscript is translated into JavaScript, which is *interpreted* by the Web Browser

- Usually PBscript runs fine in the browser even if it becomes JS
- PB applications contain lots of legacy code
 - There are some extreme cases not commonly found in most PB apps
 - Complex string or data manipulation (arrays, structures...), lots of loops, large objects, lots of tabpages in tab control, etc..

Heavy PB script - Examples



	PB	Appeon
Dynamic SQL construction	1 to 3 secs	1 to 40 secs
Data Caching	1 to 4 secs	1 to 60 secs
Window Resize	0.1 secs	1 to 20 secs
Menu walk-through	0.1 secs	1 to 2 secs

These are extreme cases of intensive
PBscript found in Customer's applications

Dynamic SQL

- Slow CRUD operations
 - DW operations via dynamic SQL rather than built-in functions
 - Each operation needs to pre-build a (potentially long) SQL string

Scan DW rows/columns looking for update flags

SQL construction (string concatenation)

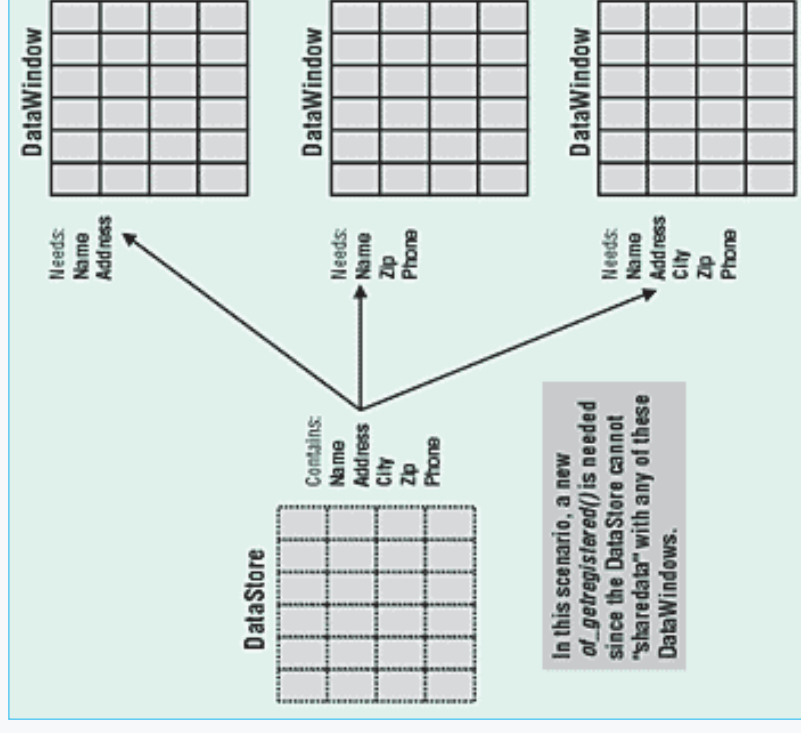
Quote replacement (string replacement)

SQL execution and resultset handling

DW Client Cache

- This is not caching DB data to avoid same server calls
- This is Master/Detail windows exchanging (part of the) same recordset
 - Changes into Detail window must be reported back to Master window
 - DW ShareData() only applicable to a subset of columns

PFC app_mgr.of_SetdwCache (TRUE)



UI events

- Limit window's resize-related events
 - Scan nested objects and recursively walk through `parent.control[]`
 - Trigger dependent objects resize

UI events (cont)

- “Retrieve as needed” menu (sub)items
 - Menu-level authorizations when a window opens
 - Retrieve items from DB and recursively walk through menu.item[]

Heavy PB script – Appeon speedup



PB	Appeon	Optimized Appeon
1 to 3 secs	1 to 40 secs	1 to 3 secs
1 to 4 secs	1 to 60 secs	1 secs
0.1 secs	1 to 20 secs	0.5 secs
0.1 secs	1 to 2 secs	0.1 secs

Summary

- Conversion to Appeon is the easy part
- Performance tuning, especially for complex legacy app, can become headache and have large impact on migration projects
- Server calls: minimize usage and/or group them
- Powerscript: review loops on large object arrays, string manipulation, useless nested resize, etc...
- Debug is the best tool to pinpoint the slow block of code

Connect with the Appeon Community



community.appeon.com

Discussions, tech articles and videos, free online training, and more.



facebook.com/AppeonPB

Encourage us with a “like”, see cool pics, and get notified of upcoming events.



twitter.com/AppeonPB

Follow Appeon and community members to get the latest tech news.



linkedin.com

Build up your career profile, and stay in contact with other professionals.



youtube.com/c/AppeonHQ

Share important Appeon videos with others; no account registration required.



google.appeon.com

Follow Appeon and community members to get the latest tech news.

「 Thank You 」

marco.meoni@gmail.com