# Performance Tuning using Log Files

Misi Mladoniczky, RRR AB

# Objectives and Results



- Pinpoint important AR System techniques of performance tuning
- WAN considerations
- Understand API/FLTR/SQL/ESCL log files
- Look at the most important API-calls and their triggers
- Prioritize - how to find recurring events and bottle necks
- Move focus to AR System Workflow as opposed to Hardware and Database

# Benefits/Value Add



- The application and workflow is where you want to focus. A bigger server or database is expensive and only a temporary solution.
- There are a lot of workflow objects
- Note that many things access ARS without being accessible through AR Admin
  - API-programs
  - Email-Engine
  - Various Plugins
- Focus on recurring events

**Use the Log Files to find out where the server spends time!**

# Performance Tuning – the typical approach



- Limit table-scans and improve searches
  - QBE-anywhere
  - Unindexed Set-Fields/Push-fields in ACTL/FLTR/ESCL
  - Tune your DBMS so that it actually use your indexes
- Limit #fields and field size
- …

# Performance Tuning – the typical approach



- Server before client
  - FLTR before ACTL
  - Use Set-Fields FLTR triggering on Get Entry instead of Set-Fields ACTL triggering on Display
  - Minimize Table-Refresh
  - Limit #fields included in the views
  - …

**Attend the BMC Remedy Performance Tuning Class!**

# Log Files – do you use them?



- How many of you regularly use server Log Files?
- How many use ACTL Log Files?
- When do you use them?
- Problems with the Log Files
  - They can be huge
  - No ‘grep’ command in Windows
  - Hard to find recurring things
  - Duration of calls are not shown
  - SQL-rows has no end timestamp
- Need to know the AR API to understand them???

# Log Files – a single file



- **API/ESCL/FLTR/SQL in the same file**
  - This gives you a chance to find the actual workflow that triggers an API- or SQL-call
  - This gives you end timestamps for the SQL-calls (look at the following call of the same server thread)
  - Use ACTL/API/FLTR/SQL logging from the client to investigate client side workflow

# Important API-calls



- **ARExport (EXP)**
  - Export of ARF/ARV files to the client cache
  - Called if form/field/menu/ACTL has been changed for the form
  - Called if the User-record has been changed
  - Called if ANY change has been made to the Group-form-data (except None-groups in version 7.x)
- **ARCreateEntry (CE)**
  - Creates an entry when user press Save
  - Creates an entry when a ACTL-Push-Fields has been issued
- **ARSetEntry (SE)**
  - Modifies an entry when a user press Save
  - Modifies an entry when an ACTL-Push-Fields has been issued

# Important API-calls



- **ARGetEntry (GE)**
  - Retrieves field data for a specific record
  - When a user Displays a Request
  - When an ACTL Set-Fields has found a record
- **ARGetListEntry (GLE)**
  - An ACTL Set-Fields before the ARGetEntry-call that retrieves the field data
  - An ACTL Push-Fields before the chosen record is created/changed with ARCreateEntry/ARSetEntry
  - If your ACTL Push-Fields should allways create a record. Clear out the Push-Field-If-Qualification instead of setting it to (1=0) or something similar
- **ARGetListEntryWithFields (GLEWF)**
  - A user search with QBE or Advanced Search
  - A table-field (make sure to refresh tables only when the data is needed, for example only if the the corresponding page-field is displayed)
  - A Crystal-Report with no big character fields (AR System ODBC)

# Important API-calls



- **ARGetMultipleEntries (GME)**
  - A plain text Report
  - A Crystal-Report with big character fields
- **ARGetListSqlForActiveLink (EXECAL)**
  - ACTL direct SQL
- **ARExecuteProcessForActiveLink (EXEC)**
  - ACTL calls to the server with
    - Run-Process @@:
    - $PROCESS$ @@:

# Important API-calls



- **ARGetMultipleCurrencyRatioSets (GMCRS)**
  - Make sure you have a Default Currency defined in Server Information -> Currency Types. If you do not even have hidden currency-fields, this is not necessary, but if you have any currency-fields in your forms, such as in ITSM, you need to do this.
- **ARServiceEntry (SVE)**
  - Service Calls are typically good. They do not send unnecessary data, and they can do a lot of things with a single call to the server.

# Accessing the Logs using RRR|Log

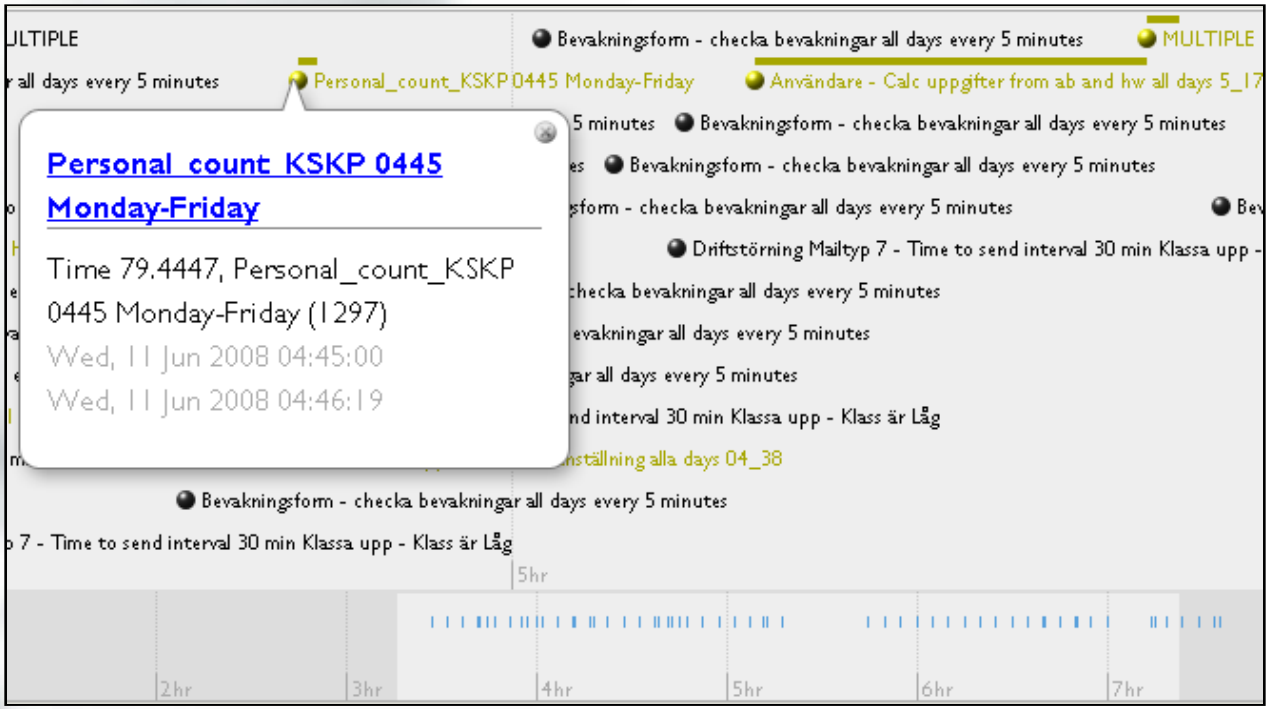


- Benefits
  - Parsing of up to **2Gb** Log Files
  - Finds **recurring** events
  - Focus on **workflow** as opposed to hardware, database, etc.
  - Become **proactive** and compare Log Files over time
  - Finds the real **bottle-necks** of your system (total time spent)
  - Helps you focus on the **"top ten"** problems
  - **Escalation Timeline**
  - **Thread** Statistics and **recommendations**
  - Network **Latency** Report

# Escalation Timeline



- Get a graphical representation of when and how long
- Click for an overview bubble with time spent and (number of records)
- Drilldown and inspect the details of individual ESCL



# Network Latency



- For WAN users, the network latency might be a bigger problem than slow execution on the server

| User | Client | IP | Median Latency | Medium Latency | Total Measured Latency | Measurment Count | Total Estimated Latency | Call Count |
|---|---|---|---|---|---|---|---|---|
| IEKE | Remedy User | 151.156.34.224 | 0.0136 | 0.0500 | 0.6001 | 12 | 0.8568 | 63 |
| PARU | Remedy User | 151.156.35.35 | 0.0134 | 0.0562 | 0.1125 | 2 | 0.4422 | 33 |
| GLAR | Remedy User | 151.156.170.234 | 0.0107 | 0.0107 | 0.1072 | 10 | 0.8025 | 75 |
| MOEM | Remedy User | 144.27.22.17 | 0.0103 | 0.0106 | 0.2331 | 22 | 1.8540 | 180 |
| KEPE | Remedy User | 151.156.75.67 | 0.0101 | 0.0268 | 0.3212 | 12 | 0.6565 | 65 |
| MAEG | Remedy User | 151.156.50.54 | 0.0096 | 0.0112 | 0.1450 | 13 | 0.6624 | 69 |
| INHO | Remedy User | 151.156.144.83 | 0.0091 | 0.0092 | 0.1107 | 12 | 0.5642 | 62 |
| LIFO | Remedy User | 151.156.239.165 | 0.0010 | 0.0010 | 0.0329 | 32 | 0.2640 | 264 |
| HBON | Remedy User | 151.156.237.144 | 0.0010 | 0.0059 | 0.1062 | 18 | 0.0450 | 45 |
| Remedy Application Service | E-mail Engine | 151.156.180.136 | 0.0010 | 0.0010 | 0.0125 | 12 | 0.6180 | 618 |

# Server Threads



- Find out how busy your server threads are, and that there is ”enough” gap between each call
- No gap means that there are queued calls
- Try to have less than 10% queued calls

| Thread id | Queue | Number of API-calls | Total active time | Total idle time | Idle percentage | Idle time = 0.0000 sec | Idle time <= 0.0001 sec | Idle time <= 0.0010 sec |
|---|---|---|---|---|---|---|---|---|
| 4444 | Admin | 30023 | 1886.7 sec | 26918.0 sec | 93.4% | 89.6% | 89.6% | 89.6% |
| 4380 | Escalation | 534 | 1224.6 sec | 37137.4 sec | 96.8% | | | |
| 4540 | Fast | 26784 | 1000.1 sec | 37457.6 sec | 97.4% | 18.2% | 18.2% | 18.2% |
| 820 | Fast | 26758 | 962.6 sec | 37495.1 sec | 97.5% | 18.1% | 18.1% | 18.1% |
| 3568 | List | 13709 | 1518.6 sec | 36937.1 sec | 96.1% | 10.2% | 10.2% | 10.2% |
| 4324 | List | 11724 | 2469.0 sec | 35985.6 sec | 93.6% | 7.1% | 7.1% | 7.1% |

# Slow or fast but recurring



- An API-integration **user** used an unindexed field.
- The integration **user** used 53% of the system Resources!

| | |
|---|---|
| Logfile Start | må dec 02 2002 **13:04:10**.7500 |
| Logfile End | må dec 02 2002 **15:16:58**.3120 |
| Seconds from start to end | **7967.6** |
| Total system time of api-calls (could be more than above with MPSO) | **10179.7** |
| Number of calls | **115608** |
| api-calls per second start to end | **14.51** |
| api-calls per second system time | **11.36** |

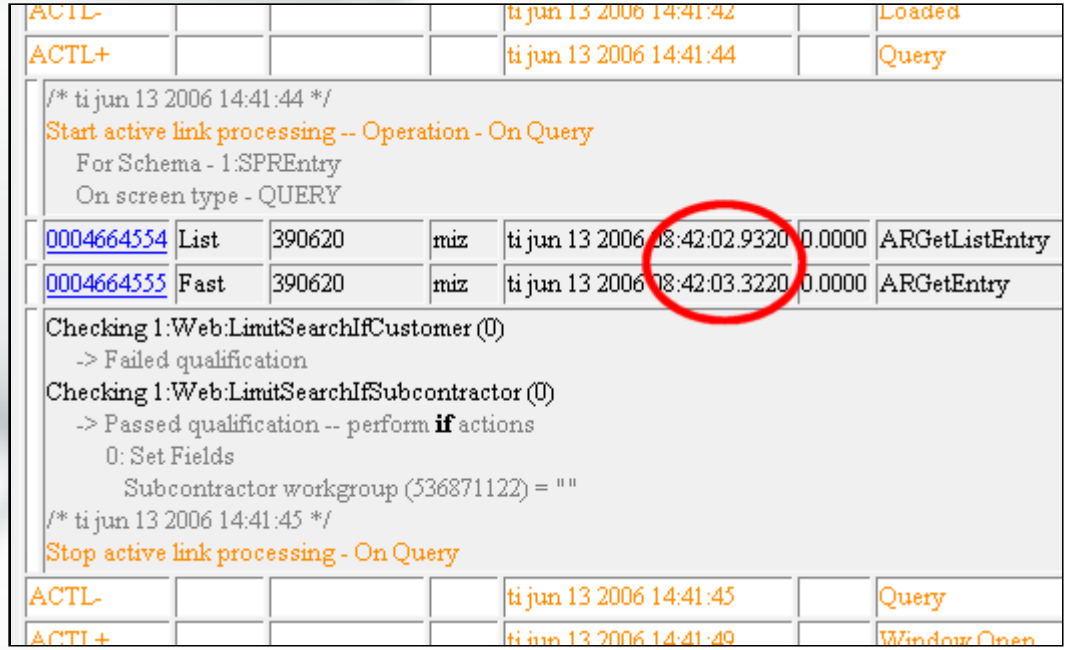
Show statistics on: API
Group statistics by: USER
Maxrows: 25
Change settings

| *Rownr* | ***Systime*** | *Sysperc* | *Numcalls* | *Average* | *Maximum* | *Minimum* | *Samples* | *User* |
|---|---|---|---|---|---|---|---|---|
| 1 | 5426.3820 | 53.31% | 7124 | 0.7617 | 35.4690 | 0.1250 | 0 1 2 3 4 5 6 7 8 9 | icluser2 |
| 2 | 538.9880 | 5.29% | 20259 | 0.0266 | 14.0000 | 0.0310 | 0 1 2 3 4 5 6 7 8 9 | p950mrg |
| 3 | 292.6140 | 2.87% | 20724 | 0.0141 | 3.2960 | 0.0150 | 0 1 2 3 4 5 6 7 8 9 | p803kro |
| 4 | 150.7260 | 1.48% | 6168 | 0.0244 | 26.7810 | 0.0150 | 0 1 2 3 4 5 6 7 8 9 | p950krh |

# Finding the Cause (Workflow)



- Search all calls from one user in chronological order
- Search for calls at a specific point in time
- Turn on client side logging and study the ACTL



| | | | | | | |
|---|---|---|---|---|---|---|
| ACTL- | | | | ti jun 13 2006 14:41:42 | | Loaded |
| ACTL+ | | | | ti jun 13 2006 14:41:44 | | Query |

/* ti jun 13 2006 14:41:44 */
Start active link processing -- Operation - On Query
For Schema - 1:SPREntry
On screen type - QUERY

| | | | | | | |
|---|---|---|---|---|---|---|
| 0004664554 | List | 390620 | miz | ti jun 13 2006 08:42:02.9320 | 0.0000 | ARGetListEntry |
| 0004664555 | Fast | 390620 | miz | ti jun 13 2006 08:42:03.3220 | 0.0000 | ARGetEntry |

Checking 1:Web:LimitSearchIfCustomer (0)
-> Failed qualification
Checking 1:Web:LimitSearchIfSubcontractor (0)
-> Passed qualification -- perform **if** actions
0: Set Fields
Subcontractor workgroup (536871122) = ""
/* ti jun 13 2006 14:41:45 */
Stop active link processing - On Query

| | | | | | | |
|---|---|---|---|---|---|---|
| ACTL- | | | | ti jun 13 2006 14:41:45 | | Query |
| ACTL+ | | | | ti jun 13 2006 14:41:49 | | Window Open |

# Searching the Log



- Enter any combination of user/text/time/form/API
- Drilldown to view the details of each call

Enter any combination of user/text/time/form/API you wish to filter/search for, at least one field need to be specified:

User APIadmin Form

API Free text filter *(slow)*

Min call length (sec.nnnn) Only Failed APIs From 2008-06-02 00:38:03.8050 (yyyy-mm-dd hh:mm:ss.nnnn)

To 2008-06-02 11:19:01.5390 (yyyy-mm-dd hh:mm:ss.nnnn) Max rows 1000 Search

| RPC ID | Thread | Queue | Client-RPC | User | Start time | Sec | API | Form |
|---|---|---|---|---|---|---|---|---|
| 2036893 | 820 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8050 | 0.0160 | n/a | |
| 2036892 | 4540 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8050 | 0.0320 | n/a | |
| 2036895 | 4540 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8520 | 0.0160 | ARGetEntry | BIANC |
| 2036894 | 820 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8370 | 0.0620 | ARMergeEntry | BIANC |
| 2036897 | 820 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8990 | 0.0160 | ARGetEntry | BIANC |
| 2036898 | 820 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.9150 | 0.0000 | ARDeleteEntry | BIANC |
| 2036896 | 4540 | Fast | 390620 | APIadmin | Mon Jun 02 2008 00:38:03.8680 | 0.0620 | ARDeleteEntry | BIANC |

# Individual Call Instances



- Drilldown to show individual calls with FLTR and SQL rows

Thread ID 0000000011
RPC ID 0000005590
Queue Fast
Client-RPC 390620
User ADMIN-MISIM

| *Rownr* | *Type* | *Timestamp* | *Delta* | *Details* |
|---|---|---|---|---|
| 46 | API | mån feb 06 2006 17:36:37.9045 | | +GE ARGetEntry -- schema AR System User Preference e |
| 47 | SQL | mån feb 06 2006 17:36:37.9056 | 0.0011 | SELECT C1,C2,C3,C4,C5,C6,C7,C8,0,C20100,C20101,C20102,C2010 FROM T684 WHERE C1 = '000000000000001' |
| 48 | SQL | mån feb 06 2006 17:36:37.9355 | 0.0299 | SELECT entryId,T0,U0,T1,U1,T2,U2,T3,U3,T4,U4 FROM |
| 49 | SQL | mån feb 06 2006 17:36:37.9377 | 0.0022 | COMMIT WORK |
| 50 | FLTR | mån feb 06 2006 17:36:37.9384 | 0.0007 | Start filter processing -- Operation - GET |
| 51 | FLTR | | | AR System User Preference - 000000000000001 |
| 52 | FLTR | | | Checking ARSystemUserPrefCustomCheck (500) |
| 53 | FLTR | | | --> Passed -- perform actions |
| 54 | FLTR | | | 0: Set Fields |
| 55 | FLTR | | | Custom Date Format (24003) = |
| 56 | FLTR | | | Custom Time Format (24015) = |
| 57 | FLTR | mån feb 06 2006 17:36:37.9391 | 0.0007 | End of filter processing (phase 1) |
| 58 | FLTR | mån feb 06 2006 17:36:37.9392 | 0.0001 | Restart of filter processing (phase 3) |
| 59 | FLTR | mån feb 06 2006 17:36:37.9393 | 0.0001 | Stop filter processing |
| 60 | API | mån feb 06 2006 17:36:37.9399 | 0.0006 | -GE OK |
| Total time of call | | | 0.0354 | |

# Visit RRR at the Vendor Fair

20

## Thank You - Questions?

Misi Mladoniczky

miz@rrr.se

+46 705 854300