

Performance Tuning Using Log Files



Misi Mladoniczky

miz@rrr.se

Misi Mladoniczky

- 12 years of AR System development
- Remedy Approved Instructor since version 2.1
- RAC certified (former)
- Founder of **RRR Scandinavia AB** www.rrr.se



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!



Performance Tuning – prioritize!

- 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!



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



API-calls that use a lot of Resources

ARExport

- Export of ARF/ARV files to the clients cache
- Called if form/field/menu/active-link 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

- Creates an entry when user press Save, or when a Push-Fields has been issued

ARSetEntry

- Modifies an entry when a user press Save, or when a Push-Fields has been issued



API-calls that use a lot of Resources

ARGetEntry

- Retrieves field data for a specific record
- When a user Displays a Request
- When an ACTL Set-Fields has found a record

ARGetListEntry

- 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

ARGetListEntry- WithFields

- A user search with QBE or Advanced Search
- A table-field
- A Crystal-Report with no big character fields (AR System ODBC)



API-calls that use a lot of Resources

ARGetMultipleEntries

- An plain text Report
- A Crystal-Report with big character fields

ARGetListSqlForActiveLink

- ACTL direct SQL

ARExecuteProcess

- ACTL calls to the server with
 - Run-Process @@:
 - \$PROCESS\$ @@:



Tools to assist you

- API/SQL Logs etc
 - arlogtmr from Remedy (community downloads)
 - Parse Log/PLOG www.mattreinfeldt.com
 - **RRR Log** from www.rrr.se (free for small log files)
- USER Logs
 - LogAnalysis www.mattreinfeldt.com
 - **RRR License** from www.rrr.se



Tuning Sample 1

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

Logfile Start	må dec 02 2002 13:04:10.7500	Show statistics on	API
Logfile End	må dec 02 2002 15:16:58.3120	Group statistics by	USER
Seconds from start to end	7967.6	Maxrows	25
Total system time of api-calls (could be more than above with MPSO)	10179.7	Change settings	
Number of calls	115608		
api-calls per second start to end	14.51		
api-calls per second system time	11.36		

<u>Rownr</u>	<u>Systemtime</u>	<u>Sysperc</u>	<u>Numcalls</u>	<u>Average</u>	<u>Maximum</u>	<u>Minimum</u>	<u>Samples</u>	<u>User</u>
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	p950brh



Tuning Sample 2

A Set-Fields ACTL has 0.4 second gap between the two API-calls even though the calls themselves execute very fast

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



Tuning Sample 3

Which call to tune here?

One takes 84 seconds to complete!

They use the same amount of processing time!

<i>Rownr</i>	<i>Systime</i>	<i>Sysperc</i>	<i>Numcalls</i>	<i>Average</i>	<i>Maximum</i>	<i>Minimum</i>	<i>Samples</i>	<i>SQL</i>
1	84.6850	17.91%	1	84.6850	84.6850	84.6850	0	SELECT TOP nnn T512.C1,C536870960,T512.C1,C8,C536870
2	84.4780	17.87%	71	1.1898	1.3750	1.0940	0 1 2 3 4 5 6 7 8 9	SELECT T604.C1,C536870915,C8,C536870917,C536870918,C
3	78.3720	16.57%	1	78.3720	78.3720	78.3720	0	BEGIN TRANSACTION UPDATE T512 SET C536871028=nnn,C536870944='xxx',C536: SELECT C536870986 FROM T512 WHERE C1 = 'xxx' UPDATE T512 SET C536870986 = 'xxx' WHERE C1 = 'xxx' SELECT C536870986 FROM T512 WHERE C1 = 'xxx' Set LOB into the above row... SELECT C536870931 FROM T512 WHERE C1 = 'xxx' UPDATE T512 SET C536870931 = 'xxx' WHERE C1 = 'xxx' SELECT C536870931 FROM T512 WHERE C1 = 'xxx' Set LOB into the above row... COMMIT TRANSACTION



Tuning Sample 4

This system apparently runs some heavy escalations!

Number of calls **216501**
api-calls per second start to end **74.50**
api-calls per second system time **43.76**

<i>Rownr</i>	<i>Systime</i>	<i>Sysperc</i>	<i>Numcalls</i>	<i>Average</i>	<i>Maximum</i>	<i>Minimum</i>	<i>Samples</i>	<i>Form</i>	<i>API</i>
1	921.3198	18.62%	59	15.6155	497.9070	0.0176	0 1 2 3 4 5 6 7 8 9	MULTIPLE	ESCL:MULTIPLE
2	391.7223	7.92%	592	0.6616	51.9493	0.0023	0 1 2 3 4 5 6 7 8 9	UB-Artikel	ARGetListEntryWithFields
3	386.1765	7.80%	144	2.6817	55.1343	0.0007	0 1 2 3 4 5 6 7 8 9	inIT	ARGetListEntryWithFields
4	253.3403	5.12%	8674	0.0292	1.0269	0.0034	0 1 2 3 4 5 6 7 8 9	KUB	ARMergeEntry
5	244.4989	4.94%	958	0.2552	10.2952	0.0665	0 1 2 3 4 5 6 7 8 9	UB-Artikel	ARSetEntry
6	146.8848	2.97%	6756	0.0217	0.4682	0.0023	0 1 2 3 4 5 6 7 8 9	PÄR	ARGetEntry



Tuning Sample 5

Thread ID 000000011
 RPC ID 0000005590
 Queue Fast
 Client-RPC 390620
 User ADMIN-MISIM

<u>Rownr</u>	<u>Type</u>	<u>Timestamp</u>	<u>Delta</u>	<u>Details</u>
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,C20103 FROM T684 WHERE C1 = '0000000000000001'
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 - 0000000000000001
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	

Inspection
 of API, FLTR
 and SQL data



The Essential tool for AR System Performance Tuning

- Try the demo version of RRR|Log
- Download this presentation at www.rrr.se/doc
- Online and Offline versions are available
- Find many other tools and utilities at RRR Online

