http://www.sapninja.com

Splunk for SAP

version 1.0

by Jim Cooke

Dataset Guide

15 August 2011

The latest version of this document can be found at
http://code.google.com/p/sapninja/source/browse/trunk/splunk/SplunkDatasets.doc

---

PO BOX 712
RANDWICK, NSW 2031
AUSTRALIA
sapslunk@gmail.com

http://www.sapninja.com

Table of Contents

http://www.sapninja.com

Datasets

Result Syntax

The dataset information is returned in Splunk-friendly format, which consists of a timestamp and a series of keypairs. The format follows this syntax rule:

```
<timestamp in format MM/DD/YYYY HH:MM:SS> <key>=<value>[;<key>=<value>…]
```

It is best shown by way of example:

```
08/15/2011 14:00:00 saphost=devhost;sysname=DEV;sysnr=00;…
```

Standard key

These fields will invariably appear at the front of each record line returned by the ***ZPLUNK_GET*** function and will be abbreviated as ***<standard key>*** in the field descriptions to represent them.

Key	Value
MM/DD/YYYY HH:MM:SS	Record timestamp
saphost	Host name where sample was taken. If system has multiple application hosts, schedule ZPLUNK_CAPTURE in each host
sysname	SAP SID (eg. DEV)
sysnr	SAP system number (eg. 00)

http://www.sapninja.com

ST03N Workload Overview

Usefulness

This overview is great for getting an aggregate view of system load and performance. You can track average response time and set up baseline measures for database processing time, cpu time and network time.
This is also great for sizing exercises. Have you ever heard people sizing your system in terms of SAPS? Do you wonder how many SAPS your system is actually using? You can pretty much derive this yourself as [1 SAP = 1 dialog step per minute]*. You can have a dashboard that divides your hourly total steps by 60 to give you a SAPS rating for your system.

*From the SAP benchmarking documentation:
SAP Application Performance Standard (SAPS) is a hardware-independent unit of measurement that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) benchmark, where 100 SAPS is defined as 2,000 fully business processed order line items per hour.
In technical terms, this throughput is achieved by processing 6,000 dialog steps (screen changes), 2,000 postings per hour in the SD Benchmark.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Tip: To get the resulting values to look like what you see in ST03N, you can divide the metric by *STEPS* to give you average times (like average response time).

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
tasktype	The task type which depends on the associated application process and is an identifiable part of each statistics record (eg. DIALOG, RFC, UPDATE)
steps	Number of dialog steps
respti	Total response time for all steps (ms)
procti	Processing Time (ms), which is a derived field or a "catch-all" bucket defined by SAP. PROCTI = RESPTI - DBTI - QUEUETI - LOADGENTI - LOCKTI - ROLLINTI - ROLLWAITTI
cputi	CPU Time (ms) obtained by work process adding up elapsed CPU time for the step it executed
dbti	Database request time (ms). The time from when a DB request is sent to the DB server until the time data is returned
queueti	Dispatcher wait time. The time a request waits in the dispatcher queue until a work process of TASKTYPE is free and assigned to the request

loadgenti	Load and generation time (ms). Time taken for required ABAP programs/screens to be loaded into buffers or generated (compiled). The load and generation time is the time required to load objects such as source code, GUI interfaces, and screen information from the database and generate these objects, if required.
lockti	Enqueue time (ms). The time during which a work process sets an enqueue (lock) request
cpicti	RFC Interface time (ms)
guinettime	Frontend network time (ms). This metric is less useful, showing network chatter time that takes place invisibly during a transaction. SAP says "A dialog step may include several communication steps between the application server and the local front end. The front end network time is made up of the following components: (1) The time for the first data transfer from the front end to the application server (2) The time for the last data transfer from the application server to the front end (3) The time used at the front end after the last data transfer for data and screen formatting"
guitime	GUI time (ms). This is probably the metric you want, rather than guinettime. It is really shows the incremental difference of running a transaction of a remote PC as opposed to doing it locally on the SAP server itself. SAP says "A dialog step may contain several communication steps between the application server and the local front end. The GUI time is the time used in the network and the local front end for these communications steps (not the time in the application server, however). The GUI time does not contain the front end network time."
guicnt	Number of GUI round trips
bytes	Data Requested from Database (KB)
rollinstep	Number of roll-in steps
rollinti	Roll-in time(ms). The roll-in time is the time used during a dialog step for rolling in the user context to the process's local working memory. During paging in, user-specific data is loaded from the roll buffer of the shared memory into the work area of the work process. During paging out, this data is written back to the roll buffer or the roll file again.
rolloutti	Roll-out time (ms). During processing of some dialog steps, the user context may be rolled out; for example, during RFCs when the client is waiting for a response from the server. This wait time until the dialog step can continue is called the roll wait time
rolloutcnt	Number of roll-out steps
rollwaitti	Roll-wait time (ms). During processing of some dialog steps, the user context may be rolled out; for example, during RFCs when the client is waiting for a response from the server. This wait time until the dialog step can continue is called the roll wait time.
rollstep	Roll-wait steps
vmc_call_count	Number of VMC requests
vmc_cpu_time	CPU time required by the VMC (ms)
vmc_elap_time	The time the work process was required by the VMC (ms)
phyreadcnt	Number of logical database changes. Number of logical change accesses to a table. These are database accesses of the types UPDATE, DELETE, or INSERT to one or more table rows.

http://www.sapninja.com

chngcnt	Number of logical database changes. Number of logical change accesses to a table. These are database accesses of the types UPDATE, DELETE, or INSERT to one or more table rows.
readdirbuf	Records read from SAP Application single record buffer. Logical database calls of the type "direct read" are triggered by ABAP commands such as SELECT SINGLE. If possible, the data is read from the local SAP buffer. Otherwise, the system must access the database itself.
phycngrec	Number of modified DB records. Number of database data records modified by UPDATE, DELETE, or INSERT requests.
readseqcnt	Number of DB sequential reads. Database calls of the type "sequential reads" are initiated by ABAP commands such as SELECT * FROM. If possible, the data is read from the local SAP buffer, otherwise from the database itself.

DERIVABLE METRICS

Formula	Description	Healthy Value	Problems Indicated
RESPTI / STEPS	Average response time (ms)	<1000 for type DIALOG	Many and varied
DBTI / STEPS	Average database time per dialog step (ms)	200-600ms or < 0.4*((RESPTI/STEPS)-(QUEUETI/STEPS))	Database problem, network problem, CPU bottleneck on DB server, DB parameters
QUEUETI / STEPS	Average dispatcher wait time (ms)	<50ms and < 0.1*RESPTI / STEPS	General problem, many causes. Too few work processes
LOADGENTI / STEPS	Average load & generation time (ms)	<50ms	Program buffer too small. CPU bottleneck. SGEN not run after upgrades
ROLLINTI / ROLLINSTEP	Average roll-in times (ms)	<20ms	SAP roll buffer too small. Extended memory too small. CPU bottleneck
ROLLOUTTI / ROLLOUTCNT	Average roll-out times (ms)	<20ms	SAP roll buffer too small. Extended memory too small. CPU bottleneck
ROLLWAITTI / ROLLSTEP	Average roll-wait times (ms)	<200ms	Problem with front-end connection speed (if shown with high GUI time) or communication to external component
GUITIME / STEPS	Average GUI time (ms)	<200ms	Problem with front-end communication, if shown with high roll-wait times
LOCKTI / STEPS	Average Enqueue time (ms)	<5ms	Problem with enqueue process. Network problems.
PROCTI / CPUTI	Proportion of CPU time to processing time	<2	CPU bottleneck or communication problem

http://www.sapninja.com

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module “splunk-sap.pl” and your connection file is called “sap.yml”, then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_ST03N_WORKLOAD
```

http://www.sapninja.com

ST03N Transactions by User

Usefulness

This dataset is great at breaking down system load by user or by transaction. For example, your company might want to develop a "user pays" charging system that charges customers differently depending on their use of resources.

This dataset is fantastic at baselining the response time of individual transactions. It makes little sense to baseline your total average dialog response time when it is perfectly normal for some transactions to take 400ms as opposed to others that might normally take 10000ms. Once you have this baseline, and you see performance degradation, you can quickly troubleshoot:

- Which part of response time grew (eg. Database processing time)
- Did this affect all transactions?
- Etc.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
tasktype	The task type which depends on the associated application process and is an identifiable part of each statistics record (eg. DIALOG, RFC, UPDATE)
bname	User name
tcode	Transaction code
repname	Report name
program	The underlying function returns either a "tcode" or a "repname", but never both … one is always blank. This is a derived field that is either a "tcode" or "repname", whichever one is not blank.
steps	Number of dialog steps
respti	Total response time for all steps (ms)
procti	Processing Time (ms), which is a derived field or a "catch-all" bucket defined by SAP. PROCTI = RESPTI - DBTI - QUEUETI - LOADGENTI - LOCKTI - ROLLINTI - ROLLWAITTI
cputi	CPU Time (ms) obtained by work process adding up elapsed CPU time for the step it executed
dbti	Database request time (ms). The time from when a DB request is sent to the DB server until the time data is returned
readdirti	Total time for direct buffer reads (ms)
readseqti	Total time for sequential reads (ms)
chngti	Total time for logical database changes (ms)
queueti	Dispatcher wait time. The time a request waits in the dispatcher queue until a work process of TASKTYPE is free and assigned to the request.

rollwaitti	Roll-wait time (ms). During processing of some dialog steps, the user context may be rolled out; for example, during RFCs when the client is waiting for a response from the server. This wait time until the dialog step can continue is called the roll wait time.
guitime	GUI time (ms). This is probably the metric you want, rather than guinettime. It is really shows the incremental difference of running a transaction of a remote PC as opposed to doing it locally on the SAP server itself. SAP says "A dialog step may contain several communication steps between the application server and the local front end. The GUI time is the time used in the network and the local front end for these communications steps (not the time in the application server, however). The GUI time does not contain the front end network time."
guicnt	Number of GUI round trips
guinettime	Frontend network time (ms). This metric is less useful, showing network chatter time that takes place invisibly during a transaction. SAP says "A dialog step may include several communication steps between the application server and the local front end. The front end network time is made up of the following components: (1) The time for the first data transfer from the front end to the application server (2) The time for the last data transfer from the application server to the front end (3) The time used at the front end after the last data transfer for data and screen formatting"

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_ST03N_USERTCODE
```

http://www.sapninja.com

ST03N Frontend Activity

Usefulness

This dataset is great for network sizing exercises. It can be used to get baseline metrics of network usage per user (and even by transaction mix, if combine with other dataset) and measure this against actual network traffic reported by your network administrators. Some customers have developed great SAP network sizing footprints relevant to their particular situation and have come up with simple sizing statements like "each SAP customer service user needs 15Kbps bandwidth" (a made-up example).

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
phost	Client hostname
steps	Number of dialog steps
bytesin	Bytes from frontend to application server
bytesout	Bytes from application server to frontend
guitime	GUI time (ms). This is probably the metric you want, rather than guinettime. It is really shows the incremental difference of running a transaction of a remote PC as opposed to doing it locally on the SAP server itself. SAP says "A dialog step may contain several communication steps between the application server and the local front end. The GUI time is the time used in the network and the local front end for these communications steps (not the time in the application server, however). The GUI time does not contain the front end network time."
guicnt	Number of GUI round trips
guinettime	Frontend network time (ms). This metric is less useful, showing network chatter time that takes place invisibly during a transaction. SAP says "A dialog step may include several communication steps between the application server and the local front end. The front end network time is made up of the following components: (1) The time for the first data transfer from the front end to the application server (2) The time for the last data transfer from the application server to the front end (3) The time used at the front end after the last data transfer for data and screen formatting"

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_ST03N_FRONTEND
```

http://www.sapninja.com

SM04 User View

Usefulness

This is good performing a count of active and inactive users (depending on time of last interaction). It can also can be used to do counts of users by location by using the subnet value from the ip address. It has also been used to go back in time and see which users were executing a particular transaction during a particular time period.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
mandt	SAP Client that user is logged on to.
bname	User name
tcode	Transaction code
lastaction	Time of last dialog interaction
term	User's fronten
ustyp	User type (Dialog, System, Communications Data, Reference, Service)
ipaddr	User's IP address

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_SM04USR
```

http://www.sapninja.com

SM04 Memory View

Usefulness

This is really good for tracking programs that are memory-intensive. It has also been used to identify people who need to be better trained in the appropriate use of report selection fields. For example, you may have a very well-behaved report with minimal memory cost, but for a particular user, they use many megabytes more memory than their peers.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
mandt	SAP Client that user is logged on to.
bname	User name
tcode	Transaction code
pagemem	Amount of page memory in bytes currently being consumed by the user.
rollmem	Amount of roll memory in bytes currently being consumed by this user.
extendedmem	Amount of extended memory in bytes currently being consumed by the user. Extended memory is used for storing the user context.
privmem	Currently full process-local (heap) memory of a user. If the extended memory is used up, or if you are dealing with a nondialog work process, process-local memory is allocated for this user context. Dialog work processes that do not receive any more extended memory change to PRIV mode (no further context switch possible) and receive heap memory.

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_SM04MEM
```

http://www.sapninja.com

SM21 System Log

Usefulness

SM21 information is surprisingly difficult to get from SAP. SAP does provide a program called *RSLGVIEW* which unpacks the central log file, but it strips out some useful information like the work process number and a couple of other things. SM21 generates a lot of "noise", but it also contains very important messages which should administrators would react to immediately if they were alerted to their presence, rather than looking for them reactively later on.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
wptype	Work process type (DIA=Dialog; BGD=Batch; UPD=Update; UP2=Secondary Update; SPO=Spool; ENQ=Lock)
wpno	Work process number as shown in SM50
mandt	Client for which message was reaised
bname	User name for which message was raised
tcode	Transaction code for which message was raised
repname	Report name for which message was raised
term	Front-end terminal associated with this message
devclass	Problem class
area	System log message group
subid	Sub-name
errmsg	Error message

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_SM21
```

http://www.sapninja.com

RZ20 CCMS Alerts

Usefulness

Yes, I know, RZ20 has it's own alert console and you can centralise your alerts into one system, and you can see it in solution manager. Okay, now the problem for you. These alerts do not resolve themselves. They stay red until a person or a program completes them. Furthermore, the interfaces and dashboards to work with them can be pretty ugly in SAP. This can generate a lot of needless work with no real value if alerts are unable to heal themselves. If you import the alerts into Splunk, you have another way to monitor them and react to them with a much less "clunky" interface. There is even a "Splunk for Nagios" plug-in (http://splunk-base.splunk.com/apps/22374/splunk-for-nagios), if you use Nagios alerting.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
mteclass	The monitoring tree element (MTE)
severity	Y for yellow, or R for Red
msg	Alert text

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_CCMS_A
```

http://www.sapninja.com

RZ20 CCMS MTE Current Values

Usefulness

A hidden little gem and might be the only source of some types of information. For example, you may have configured your system to alert you if you run out of number range entries for spool jobs. However, you can plot how many of the numbers are used by sampling the current measurements of MTE ***R3SpoolUsedNumbers*** over time. This way, you can see how things are trending, rather than reacting to alerts.
It is very useful for tuning memory buffers. Rather than looking at ST02 and seeing a current value and a high-water mark, you can actually start to track how much is used and can tune your system for the normal range of operation, rather than to cater for rogue events which give the high-water mark a spurious value.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
mteclass	The monitoring tree element (MTE)
severity	G for Green, Y for yellow, or R for Red
msg	Alert text
obs	Observation value (eg. 512)
obs_uom	Observation unit of measure (eg. Bytes)

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_CCMS_C
```

http://www.sapninja.com

SM50 Work Process Overview

Usefulness

This dataset is very useful. One of the problems of SM50 and dpmon is that the information is transitory and not recorded historically. There have been many occasions when problems have been reported out of hours and the opportunity to view SM50 information has passed. This dataset has enabled me to get to the root cause of many problems. Examples of such root causes are:

- No available free work processes for DIALOG, BACKGROUND or UPDATE
- UPDATES stuck
- Long-running rogue processes
- Disproportionately long INSERT statements (turned out to be an ASSM problem in Oracle)
- Locks on the number range buffer (status HOLD WAIT on table NRIV)
- Long-running sequential reads

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
wp_no	Work process number as shown in SM50
wp_type	Work process type (DIA=Dialog; BGD=Batch; UPD=Update; UP2=Secondary Update; SPO=Spool; ENQ=Lock)
wp_pid	Work process operating system PID
wp_status	Status of work process "Waiting"=doing nothing,waiting for work; "Running"=executing, etc. etc.
wp_dumps	Number of error dumps or work process restarts
mandt	Client that this work process is servicing request for
bname	User that this work process is servicing request for
repname	Program being executed
eltime	Elapsed time (execution time in seconds)
wp_action	What the work process is doing. Useful to see for sequential reads and other types of database reads
Table	Which database table, if applicable, the action is being performed on

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module “splunk-sap.pl” and your connection file is called “sap.yml”, then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_SM50
```

http://www.sapninja.com

SP01 Completed Spool Jobs

Usefulness

This metric is useful for analysing print volumes from your SAP server, especially if you need to get an indication of high-volume users or printers. The real value for this dataset is to establish baselines for trending.

SAP doesn't always get the information like pages and bytes correct. If you need high accuracy for network traffic, I suggest using *tcpdump* or *wireshark*. Accurate measures of page counts and bytes can be obtained by the logs of the print servers themselves.

Technical Details

DATASET FIELDS RETURNED BY ZPLUNK_GET

Key	Value
<Standard Key>	Timestamp, host, system name and system number as described above
spoolid	Spood Id number (as shown in SP01)
mandt	Client from which print job originated
bname	User who submitted print job
prnshortname	Printer short name (legacy 4-character short name code)
printer	Full printer name
prnpages	Pages output (as shown in SP01)
prnbytes	Bytes sent to print spooler (as shown in SP01)
prntype	Output type (as shown in SP01)

RFC CALLING SYNTAX

If you have been a follower of the Installation Guide to the letter and have named the Perl module "splunk-sap.pl" and your connection file is called "sap.yml", then the exact command to suck out the data is as follows.

```
perl splunk-sap.pl sap.yml ZPLUNK_SPOOL
```