2012
IBM System z Technical University

Enabling the infrastructure for smarter computing

Problem Determination with Linux on System z

zLG07

Susanne Wintenberger

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

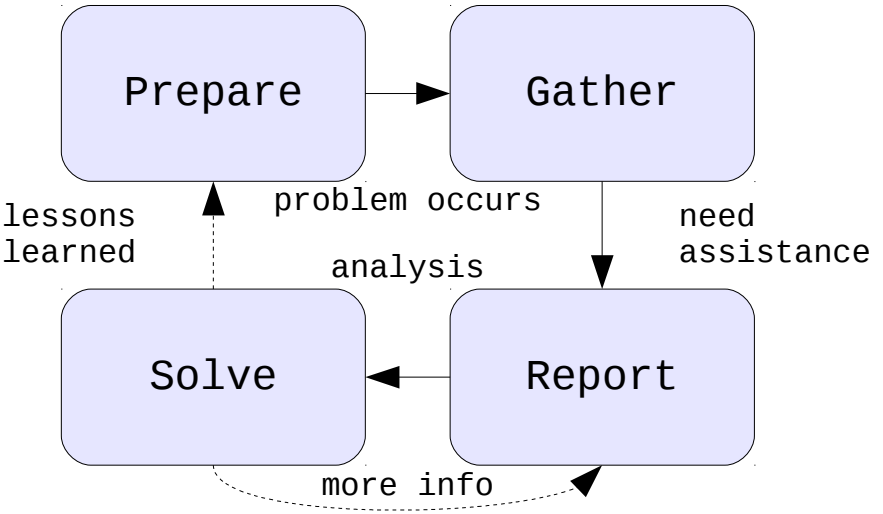
Introductory Remarks

- Looks straight forward on the charts, ...
  - But a problem does not necessarily show up on the place of origin
  - Analysis can take weeks
    - Starts to look simple once you know the solution
  - Memory overwrites as an example
    - Can cause symptoms anywhere

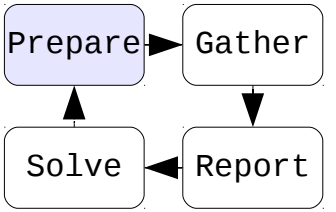
- More information → faster problem resolution
  - Gathering and submitting additional information introduces delays.
  - Having a structured process for yourself eases a service request if needed

Agenda

- Prepare
  - System and Workload descriptions
  - Healthy system data for comparison
- Gather
  - In case of emergency
- Report
  - How to report a Problem Description
- Solve
  - Tools to start an analysis

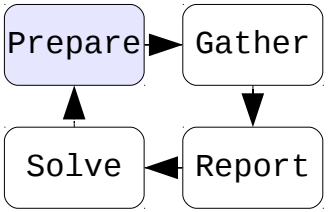


Trouble Shooting First Aid Kit – be prepared



- Install some packages required for debugging
  - s390-tools/s390-utils
    - dbginfo.sh
  - sysstat
    - sadc/sar
  - dump tools crash / lcrash
    - lcrash (lkcdutils) available with SLES10
    - crash available on SLES11
    - crash in all RHEL distributions
  - Use these pro-actively in healthy system as well

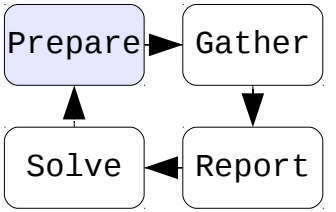
dbginfo script



- It collects various system-related files for debugging purposes.
  - It captures the current system environment and generates a tar file, which can be attached to PMRs / Bugzilla entries
- part of the s390-tools package in SUSE and s390-utils package in recent Red Hat distributions
  - dbginfo.sh gets continuously improved by service and development
  - Check out: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html
- In order to run the script properly
  - Ensure that it is run as root user.
  - Under z/VM, the appropriate privilege classes help to be authorized for some used commands (e.g. privilege class B)
- It is similar to the Red Hat tool sosreport or to the SUSE tool supportconfig

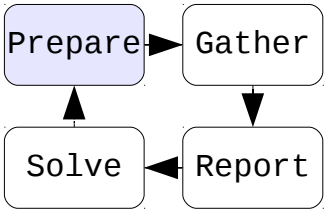
```
root@larsson:~>  dbginfo.sh
Create target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
Change to target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
[...]
```

dbginfo script (cont'd)



- dbginfo.sh captures the following information:
  - /proc/[version, cpu, meminfo, slabinfo, modules, partitions, devices ...]
  - System z specific device driver information: /sys/kernel/debug/s390dbf
  - Kernel messages /var/log/messages
  - Reads configuration files in directory /etc/ [ccwgroup.conf, fstab ...]
  - Uses several commands: ps, dmesg
  - Query setup scripts: lscss, lsdasd, lsqeth, lszfcp, lstape, ...
  - And much more
- If the Linux system runs as z/VM guest operating system, dbginfo collects information about the z/VM guest setup:
  - Release and service Level: q cplevel
  - Network setup: q [lan, nic, vswitch, v osa, ...]
  - Storage setup: q [set, v dasd, v fcp, q pav ...]
  - Configuration/memory setup: q [stor, v stor, xstore, cpus...]

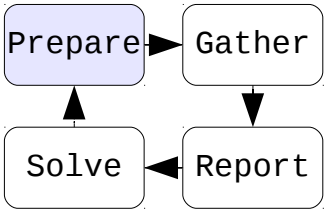
supportconfig (SUSE)



- It gathers system troubleshooting information.
  - It captures the current system environment and generates a tar-archive.
- The script file collects complementary info to dbginfo.sh.
- Running the script requires root authority.

```
root@larsson:~>  supportconfig
================================================================
                     Support Utilities - Supportconfig
                          Script Version: 2.25-96
                          Script Date: 2009 02 24
================================================================
Gathering system information
  Basic Server Health Check...                    Done
[…]
Creating Tar Ball
[ DONE ]=======================================================
  Log file tar ball: /var/log/nts_h42lp42_100719_1431.tbz
  Log file size:     572K
  Log file md5sum:   1dfc98f3a3192771ad970ecc31b6e9d9
```

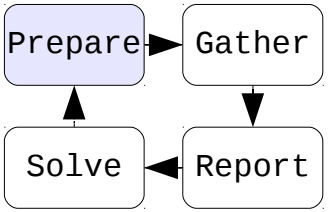
sosreport (Red Hat)



- It gathers system troubleshooting information.
  - It captures the current system environment and generates a tar-archive.
- The script file collects complementary info to dbginfo.sh.
- Running the script requires root authority.

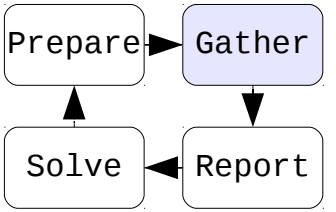
```
root@larsson:~>  sosreport
sosreport (version 1.7)
[...]
This process may take a while to complete.
No changes will be made to your system.

Press ENTER to continue, or CTRL-C to quit.

Please enter your first initial and last name [h42lp27]: ABC
Please enter the case number that you are generating this report for:
DEF

Creating compressed archive...
Your sosreport has been generated and saved in:
  /tmp/sosreport-ABC-427338-6e8879.tar.bz2
[...]
```

Describe the system



- Describe the software setup
  - What is the System/Workload intended to do ?
  - What software (versions) are used for that ?
    - System (Distribution)
    - Middle-ware components
- Describe the hardware setup
  - Machine and Storage type
  - Storage and Network attachments
- Describe the infrastructure setup
  - Clients
  - Network topology (firewalls, devices, vswitches, vlans, …)
  - Disk configuration (multipath, lvm, storage server setup, …)

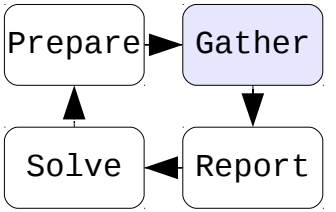
Trouble Shooting First Aid Kit - emergency



- General
  - Collect dbginfo.sh output then compare with healthy systems log
  - increase log level in /sys/kernel/debug/s390dbf for affected subsystems

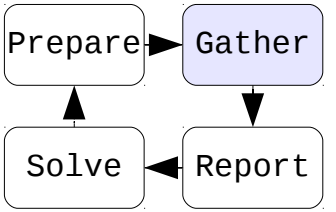
- In case of a performance problem
  - Always archive syslog (/var/log/messages)
  - Start sadc (System Activity Data Collection) and provide sar files
  - If running as guest under z/VM, collect z/VM MONWRITE data
  - Periodically, collect and archive some data during your peak periods, so that you have a historical record
    - Peak loads
    - month-end processing
    - Significant changes (e.g. moving from z10 to z196, refreshing level of application code)

Trouble Shooting First Aid Kit – emergency (cont'd)



- In case of a disk problem
  - Enable disk statistics
- In case of a network problems
  - Provide a diagram of your network setup
  - Run lsqeth (part of s390-tools package)
- In case of a system hangs
  - Take a kernel dump
    - Include System.map, Kerntypes (if available) and vmlinux file
  - See “Using the dump tools” book on http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/l26ddt02.pdf

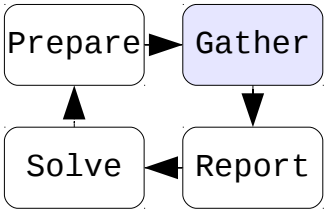
System z debug feature (s390dbf traces)



- System z specific driver tracing environment
  - Uses ring buffers
  - Available in live system and in system dumps
- Must be mounted for live view:
  - '`mount -t debugfs /sys/debug /sys/kernel/debug`'
- Each component has these control interfaces
  - level – controlling the trace detail between 0 <-> 6 (lowest-highest) default: 2
    - Increase pages when logging with high levels: '`echo 6 > level`'
  - pages – shows and defines the preallocated space: '`echo 20 > pages`'
  - flush – cleans the ring buffer: '`echo 1 > flush`'
- And one of these output files
  - hex_ascii – output is not that human readable, but very useful for debugging
  - sprintf – human readable output, usually an event log

```
cat /sys/kernel/debug/s390dbf/qeth_msg/sprintf
00 01289399222:389736 5 - 01 000003c01956f346  IPA: delipm(xB5) for eth1 succeeded
00 01289399222:390166 5 - 01 000003c01956f346  IPA: destroy_addr(xC4) for eth1 succeeded
00 01289399224:977051 5 - 01 000003c01956f346  IPA: qipassist(xB2) for eth1 succeeded
```

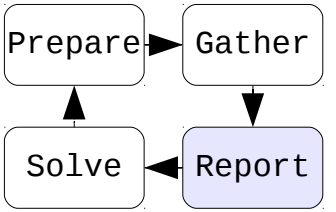
Describe the problem



- What is the symptom ?
  - When did it happen ?
    - Date and time, important to dig into logs
    - How frequently does it occur ?
    - Is there any pattern ?
  - Is this a first time occurrence ?
    - Was anything changed recently ?
    - Diffs of dbginfo can save your day
  - Where did it happen ?
    - One or more systems, production or test environment ?
  - Is the problem reproducible ?

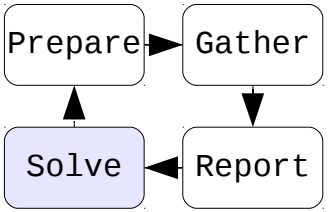
- Write down as much as possible information about the problem !

Trouble Shooting First Aid Kit - report



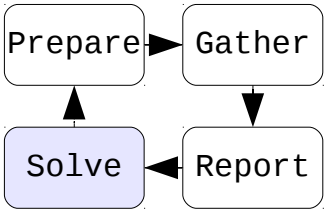
- Problem report
  - Provide your problem and environment description
  - Attach the output file of dbginfo.sh, any (performance) reports or logs
  - Upload dump data
  - Use meaningful names for the output files (e.g. tool_test_case_date_and_time)
  - z/VM MONWRITE data
    - Binary format, make sure, record size settings are correct.
    - For details see http://www.vm.ibm.com/perf/tips/collect.html
- When opening a PMR
  - Upload comprehensive documentation to directory associated to your PMR at
    - ftp://ecurep.ibm.com/, or ftp://testcase.boulder.ibm.com/
  - See Instructions: http://www.ibm.com/de/support/ecurep/other.html
- If opening multiple partner tickets, let them know about each other
- When opening a Bugzilla (bug tracker web application) at distribution partner attach documentation to Bugzilla

sadc/sar



- Capture Linux performance data with sadc/sar
  - CPU utilization
  - Disk I/O overview and on device level
  - Network I/O and errors on device level
  - Memory usage/swapping
  - Reports statistics data over time and creates average values for each item
- sadc example (for more see man sadc)
  - System Activity Data Collector (sadc) --> data gatherer
  - **`/usr/lib64/sa/sadc [options] [interval [count]] [binary outfile]`**
  - `/usr/lib64/sa/sadc 10 20 sadc_outfile`
  - `/usr/lib64/sa/sadc -d 10 sadc_outfile`
  - -d option: collects disk statistics
  - Choosing the right interval can be important
    - Too small → too much data & overhead, can mask the issue
    - Too large → values are too “averaged”, peaks no more visible

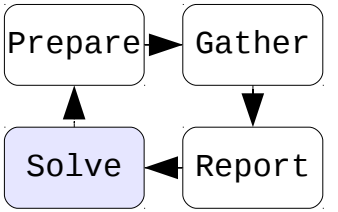
sadc/sar (cont'd)



- sar example (for more see man sar)
  - System Activity Report (sar) command --> reporting tool
  - **`sar [options] sadc_outfile > [sar outfile]`**
  - `sar -A -f sadc_outfile > sar_outfile`
  - -A option: reports all the collected statistics
  - -f option: specifies the binary sadc output file
  - enables the creation of item specific reports e.g. network
  - enables the specification of a start and end time → averages are created for the time of interest
- Should be started as a service during system start e.g.

  `'service sysstat start'`
- Please always include both the sadc and the 'sar -A' files when submitting SAR information to IBM support
  - This often allows to verify/falsify conclusions seen in other parts of the report

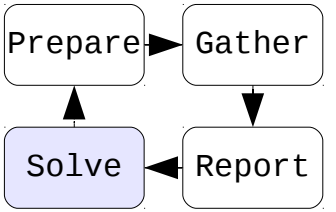
Processes created




```
root@h42lp42
File  Edit  View  Terminal  Help
Linux 2.6.16.60-0.59.1-default (h42lp42)        23/02/10

14:14:55       proc/s
14:15:05         2.69
14:15:15         0.40
14:15:25         0.10
14:15:35         0.30
14:15:45         0.00
Average:         0.70
```

Processes created per second usually small (e.g. < 10) except during startup.
If constantly at a high rate (e.g. > 100) your application likely has an issue.
Be aware – the numbers scale with your system size and setup.

Context switches



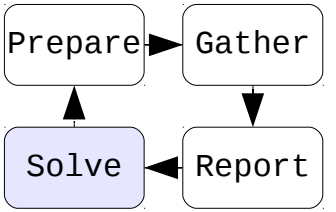
root@h42lp42:~

File Edit View Terminal Help

```
14:14:55      cswch/s
14:15:05      1110.10
14:15:15      1045.70
14:15:25      1049.10
14:15:35      1120.78
14:15:45      1229.09
Average:      1110.95
```

Context switches per second usually < 1000 per cpu
except during startup or while running a benchmark
if > 10000 (per cpu) your application likely has an issue or critical resources are blocked

CPU utilization



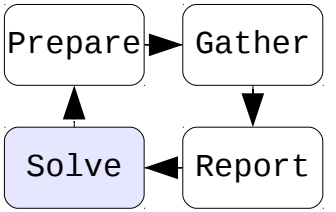

```
Per CPU values:
watch out for
        system time (kernel time)
        iowait time (runnable, but waiting for I/O)
        steal time (runnable, but time taken by
other guests)
```

root@h42lp42

File Edit View Terminal Help

14:14:55	CPU	%user	%nice	%system	%iowait	%steal	%idle
14:15:05	all	26.64	0.00	12.03	25.92	6.24	29.16
14:15:05	0	43.81	0.00	5.49	23.25	4.99	22.46
14:15:05	1	4.30	0.00	10.19	28.67	9.89	46.95
14:15:05	2	11.81	0.00	28.03	45.15	5.01	10.01
14:15:05	3	46.61	0.00	4.49	6.79	4.99	37.13
14:15:15	all	27.19	0.00	11.93	25.11	7.75	28.01
14:15:15	0	90.60	0.00	3.70	0.00	5.70	0.00
14:15:15	1	9.24	0.00	22.49	41.57	9.24	17.47
14:15:15	2	5.98	0.00	14.64	46.71	9.06	23.61
14:15:15	3	2.90	0.00	6.99	12.09	7.09	70.93

Networking data (1)



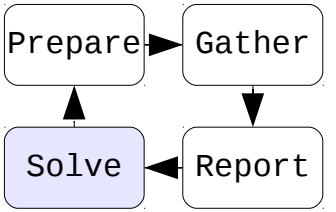
root@h42lp42

File Edit View Terminal Help

14:14:55	IFACE	rxpck/s	txpck/s	rxkB/s	txkB/s	rxcmp/s	txcmp/s	rxmcst/s
14:15:05	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:05	sit0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:05	eth0	4587.92	5278.34	307.53	482.56	0.00	0.00	0.00
14:15:15	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:15	sit0	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:15	eth0	4206.40	4827.10	281.43	441.17	0.00	0.00	0.00

Per interface statistic of packets/bytes
You can easily derive average packet sizes from that.
Sometimes people expect - and planned for – different sizes.

Networking data (2)



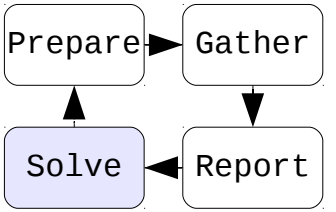
root@h42lp42

File Edit View Terminal Help

14:14:55	IFACE	rxerr/s	txerr/s	coll/s	rxdrop/s	txdrop/s	txcarr/s	rxfram/s	rxfifo/s	txfifo/s
14:15:05	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:05	sit0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:05	eth0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:15	lo	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:15	sit0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14:15:15	eth0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

```
Rates of unsuccessful transmits/receives
        per interface
        rx/tx errors
        dropped packages
                inbound error
```

Disk I/O I - overall



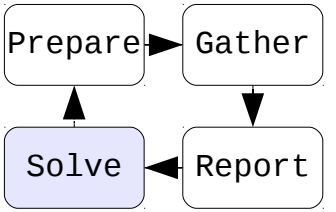
root@h42lp42

File Edit View Terminal Help

14:14:55	tps	rtps	wtps	bread/s	bwrtn/s
14:15:05	445.71	61.38	384.33	7715.77	55529.74
14:15:15	192.20	32.90	159.30	7308.80	68233.60
14:15:25	171.70	1.20	170.50	9.60	70798.40
14:15:35	327.25	174.95	152.30	1399.60	68261.88
14:15:45	444.74	310.51	134.23	2484.88	59704.50
Average:	316.35	116.15	200.20	3784.61	64504.50

Overview of
- operations per second
- transferred amount

Disk I/O II – per device



root@h42lp42

File Edit View Terminal Help

14:18:14	DEV	tps	rd_sec/s	wr_sec/s	avgrq-sz	avgqu-sz	await	svctm	%util
14:18:24	dev94-0	7.41	260.26	37.64	40.22	0.01	1.35	0.95	0.70
14:18:24	dev94-4	403.20	46784.38	13756.96	150.15	5.06	12.56	2.03	81.88
14:18:24	dev94-8	547.15	22830.83	21249.25	80.56	3.42	6.25	1.39	76.18
14:18:34	dev94-0	8.30	557.31	10.28	68.38	0.01	1.31	0.71	0.59
14:18:34	dev94-4	284.39	35453.75	35618.18	249.91	7.82	23.45	2.97	84.58
14:18:34	dev94-8	549.51	16032.41	41554.94	104.80	25.23	40.35	1.42	78.06

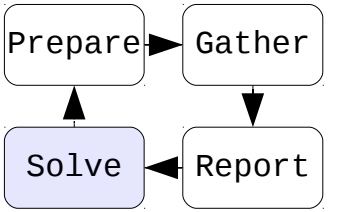
Is your I/O balanced across devices?
Imbalances can indicate issues with a LV setup.

Avgqu-sz shows how many I/O requests are not dispatched

Await shows the time the application has to wait.
(includes the time spent by the requests in queue and the time spent servicing them).

Svctm shows the time spent outside linux

Memory statistics



root@h42lp42

File Edit View Terminal Help

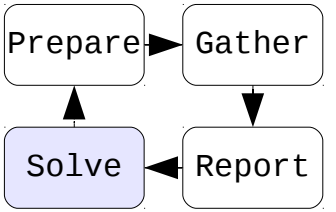
14:18:14	kbmemfree	kbmemused	%memused	kbbuffers	kbcached	kbswpfree	kbswpused	%swpused	kbswpcad
14:18:24	9616	2045284	99.53	2772	90328	1621184	782792	32.56	616916
14:18:34	8624	2046276	99.58	2936	154636	1443732	960244	39.94	729948
14:18:44	7024	2047876	99.66	5400	240140	1132356	1271620	52.90	953644
14:18:54	7308	2047592	99.64	4556	348796	1201988	1201988	50.00	778752
14:19:04	7876	2047024	99.62	7800	333844	1201988	1201988	50.00	780656
Average:	8090	2046810	99.61	4693	233549	1320250	1083726	45.08	771983

Watch

Be aware that high %memused and low kbmemfree
is no indication of a memory shortage (common mistake).

Same for swap – to use swap is actually good,
but to access it (swapin/-out) all the time is bad.

Swap rate



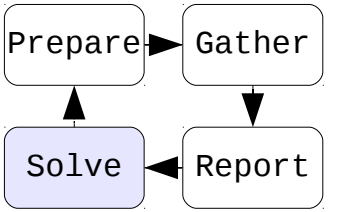
root@h42lp42

File Edit View Terminal Help

```
14:18:14     pswpin/s pswpout/s
14:18:24      2853.95   2658.26
14:18:34      2003.26   5399.80
14:18:44        88.59   9921.92
14:18:54      3199.30     53.15
14:19:04      4057.46      0.00
Average:      2443.91   3598.50
```

Swap rate to disk swap space
(application heap & stack)
Ideally the swap rates is near zero after a rampup time.
if high rates (>1000 pg/sec) for longer time you are likely short on memory
or your application has a memory leak

Disk I/O Paging statistics



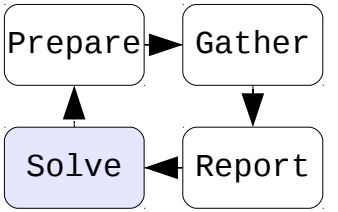
root@h42lp42

File Edit View Terminal Help

14:18:14	pgpgin/s	pgpgout/s	fault/s	majflt/s	pgfree/s	pgscank/s	pgscand/s	pgsteal/s	%vmeff
14:18:24	34953.75	17528.73	4613.41	383.98	16879.78	24873.87	12569.07	10445.25	27.90
14:18:34	26002.77	39554.15	3009.39	282.11	17059.49	29168.48	12723.91	10922.33	26.07
14:18:44	14628.69	41913.94	162.32	13.74	8904.65	17556.67	8983.33	4180.91	15.75
14:18:54	49157.64	234.17	8755.84	507.49	19203.10	19190.11	659.34	12217.98	61.55
14:19:04	40633.03	17185.19	5696.40	668.87	22180.28	17035.14	62.76	15202.60	88.92
Average:	33096.42	23282.78	4453.17	371.71	16861.25	21590.88	7008.46	10606.86	37.09

Faults populate memory
Major faults need I/O
Scank/s is background reclaim by kswap/flush (modern)
Scand/s is reclaim with a “waiting” allocation
Steal is the amount reclaimed by those scans

System load



root@h42lp42

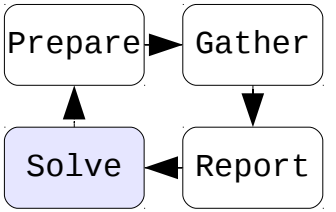
File Edit View Terminal Help

```
14:14:55      runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15
14:15:05            3        87      3.76      3.69      3.70
14:15:15            4        87      4.10      3.76      3.72
14:15:25            3        88      4.54      3.87      3.76
14:15:35            2        89      4.45      3.87      3.76
14:15:45            2        87      4.70      3.94      3.78
Average:            3        88      4.31      3.83      3.74
```

Watch runqueue size snapshots runq-sz (runnable programs)
It's not bad to have many, but if they exceed the amount of CPUs you could do more work in parallel.

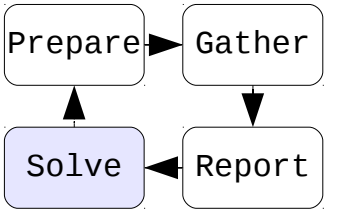
Plist-sz is the overall number of processes, if that is always growing you have likely a process starvation or connection issue.
Load average is runqueue length average in 1/5/15 minutes

DASD statistics



- DASD statistics
  - records (mostly processing time) of I/O operations of a specific period as statistic data
  - Monitors activities of the DASD device driver and the storage sub system
  - Shows I/O statistics for the whole system
- Capture DASD statistics data
  - Activate via `'echo set on > /proc/dasd/statistics'`
  - Summarized histogram information available in /proc/dasd/statistics
  - Deactivate via `'echo set off > /proc/dasd/statistics'`
  - To view the statistics:
    - Summary over all statistics: `'cat /proc/dasd/statistics'`
    - For individual DASDs: `'tunedasd -P /dev/dasda'`

DASD statistics (cont'd)



4 kb <= request size < 8 kb

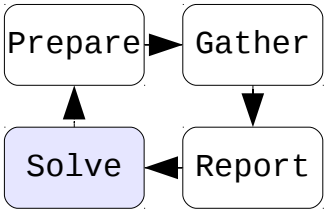
1 ms <= response time < 2 ms

root@h42lp27:~

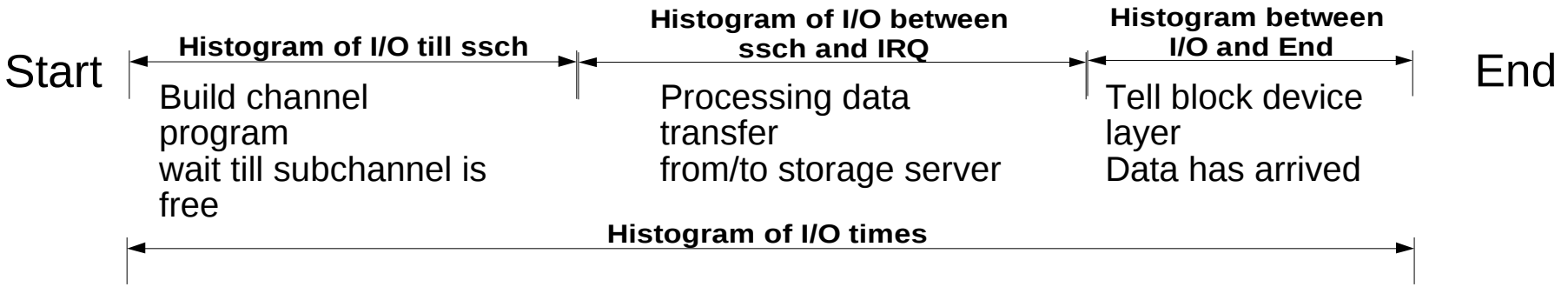
File Edit View Terminal Tabs Help

```
[root@h42lp27 ~]# cat /proc/dasd/statistics
38975 dasd I/O requests
with 11427880 sectors(512B each)
   __<4     ___8     __16     __32     __64     _128     _256     _512     __1k     __2k     __4k     __8k     _16k     _32k     _64k     128k
   _256     _512      _1M      __2M     __4M     __8M     _16M     _32M     _64M     128M     256M     512M      __1G     __2G     __4G     >4G
Histogram of sizes (512B secs)
      0        0    12331      334     1906     2734     4422     7218     9702      328        0        0        0        0        0        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O times (microseconds)
      0        0        0        0        0        0        0     2966     1879    11897     2812     4530     8965     5905       19        2
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O times per sector
      0     2263     4981    16461     3564      516     8743     2022      195      196       29        5        0        0        0        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O time till ssch
   5325       11      132      107        3        7       14      730     1550    10480     2438     5902     9783     2481       12        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O time between ssch and irq
      0        0        0        0        0        0        0    14473     4675     7186     9333     3299        3        5        1        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O time between ssch and irq per sector
      0    22357     4001      277    12322       13        3        0        0        1        1        0        0        0        0        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
Histogram of I/O time between irq and end
  38902       72        0        0        0        1        0        0        0        0        0        0        0        0        0        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
# of req in chanq at enqueuing (1..32)
      0     5571     2292      376      339    30396        0        0        0        0        0        0        0        0        0        0
      0        0        0        0        0        0        0        0        0        0        0        0        0        0        0        0
```

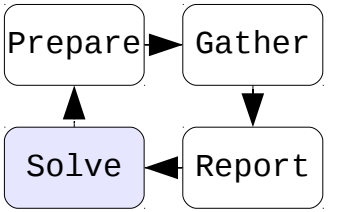
DASD statistics (cont'd)



- DASD statistics decomposition
  - Each line represents a histogram of times for a certain operation
  - Operations split up into the following :



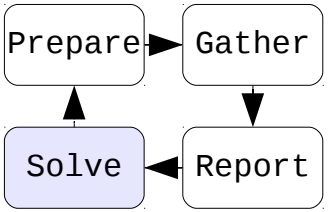
top



- The top command shows resource usage on process thread level
- top example (for more see man top)
  - **`top [options] -d [delay] -n [iterations] -p [pid, [pid]]`**
  - *`top -d 1`*
  - `top -b -d 1 -n 180  >top.log 2>&1 &` => batch mode, 3 minutes
  - Customize interactively, “`W`” writes to ~/.toprc (default config)

```
root@h42lp42
File  Edit  View  Terminal  Help
top - 17:16:36 up  4:32,  3 users,  load average: 2.93, 2.76, 2.72
Tasks:  70 total,   1 running,  69 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.3%us, 14.8%sy,  0.0%ni, 78.2%id,  5.2%wa,  0.1%hi,  0.2%si,  0.2%st
Mem:   2054900k total,   226584k used,  1828316k free,    37320k buffers
Swap:  2403976k total,    18368k used,  2385608k free,   110672k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2193 root      16   0 28148 1836  972 S   56  0.1 135:26.27 blast.LzS
    1 root      16   0   848   64   32 S    0  0.0   0:00.68 init
    5 root      34  19     0    0    0 S    0  0.0   0:03.36 ksoftirqd/1
  239 root      15   0     0    0    0 S    0  0.0   0:00.35 kjournald
```

ps



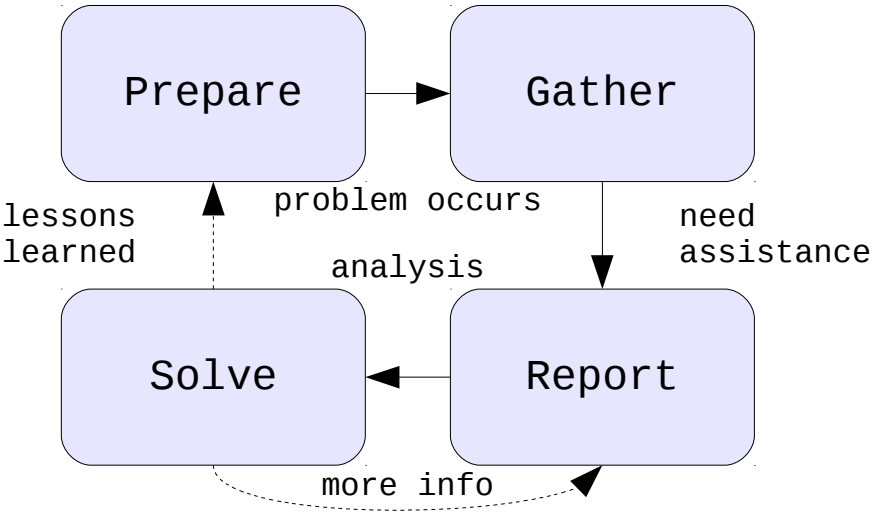
- The ps command reports a snapshot of the current processes
- ps example (for more see man ps)
  - to see every process with a user-defined format
  - *ps -eLo pid,user,%cpu, %mem,wchan:15,nwchan,stat,time,flags,etime,command:50*

```
wchan/stat to search stalls/serialization
Time is accumulated
```

```
root@h42lp42:~
File  Edit  View  Terminal  Help
  PID USER     %CPU %MEM WCHAN           WCHAN STAT     TIME F     ELAPSED COMMAND
 1627 root      0.5  0.0 SyS_select      256024 Ss   00:01:24 0    04:32:35 zmd /usr/lib/zmd/zmd.exe --sleep 84568
 1643 root      0.0  0.0 SyS_select      256024 Ss   00:00:00 5 13-04:23:07 /usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
 1704 root      0.0  0.1 SyS_epoll_wait  2962b0 Ss   00:00:03 4 13-04:23:07 /usr/lib/postfix/master
 1713 postfix   0.0  0.1 SyS_epoll_wait  2962b0 S    00:00:00 4 13-04:23:07 qmgr -l -t fifo -u
 1728 root      0.0  0.0 SyS_nanosleep   18d8b6 Ss   00:00:01 1 13-04:23:07 /usr/sbin/cron
 1736 root      0.0  0.0 read_chan       35b900 Ss+  00:00:00 4 13-04:23:06 /sbin/mingetty --noclear /dev/ttyS0 dumb
 2015 root      0.0  0.0 zfcp_erp_thread af213a S    00:00:00 1 13-04:21:27 [zfcperp0.0.1900]
 2016 root      0.0  0.0 scsi_error_hand 98fcee S<   00:00:00 1 13-04:21:27 [scsi_eh_0]
 2017 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [scsi_wq_0]
 2018 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [fc_wq_0]
 2019 root      0.0  0.0 worker_thread   17453a S<   00:00:00 1 13-04:21:27 [fc_dl_0]
 7936 root      0.0  0.0 kjournald       829c22 S    00:00:00 1 11-16:37:13 [kjournald]
20212 root      0.0  0.0 pdflush         1ce904 S    00:00:06 1 10-04:40:02 [pdflush]
26186 root     93.9  0.1 -                    - Rl   00:00:39 1       00:43 ./blast.LzS blast.cfg run.list
```

Summary

- Preparation can be the key to a quick solution
  - Use our documentation resources
- It all starts with good descriptions
  - System
  - Workload
  - Environment
  - Problem
- Tools gathering important data
  - dbginfo script
  - System z debug feature
  - sadc/sar, ps, top, ... if applicable
  - Dump tools for hangs



http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/com.ibm.trouble.doc/troubl

References

- Trouble Shooting and Support for Linux on System z:
  http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/com.ibm.trouble.doc/t
- Linux on System z project at IBM DeveloperWorks:
  http://www.ibm.com/developerworks/linux/linux390/
- Linux on System z: Tuning Hints & Tips
  http://www.ibm.com/developerworks/linux/linux390/perf
- Optimize disk configuration for performance:
  http://www.ibm.com/developerworks/linux/linux390/perf/tuning_rec_dasd_optimize
- Linux-VM Performance Website:
  http://www.vm.ibm.com/perf/tips/linuxper.html
- IBM Redbooks:
  http://www.redbooks.ibm.com/
- IBM Techdocs:
  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions?

Susanne Wintenberger

Certified IT Specialist

Linux on System z

Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-3514
swinten@de.ibm.com