Measurement Tools and Techniques II

Jan Hetflejš, Karel Motlík

18. 3. 2013

Measurement Tools and Techniques II

```
Linux 2.4.22-1.2149.nptl (bhcompile@daffy) (gcc 3.2.3 20030422 ) #1 1CPU [pax]

Memory:      Total        Used        Free      Shared     Buffers      Cached
Mem:        514672      501928       12744           0      189932       83804
Swap:       594396       47704      546692

Bootup: Mon Jan 26 10:47:13 2004    Load average: 0.10 0.05 0.01 1/93 21142

user  :    1d  1:40:40.16   6.4%  page in :  2185526  disk 1:       45r       1w
nice  :        0:01:56.80   0.0%  page out: 17943250  disk 2:   287031r 2014488w
system:      16:19:44.16   4.0%  swap in :    14041  disk 3:    17512r  131673w
idle  :  15d  1:55:06.23  89.6%  swap out:    32127  disk 4:      593r     523w
uptime:  16d 19:57:27.35         context :431330839

irq  0: 145424735 timer               irq  8:          1 rtc
irq  1:    156715 keyboard            irq  9:   19757050 eth0, eth1
irq  2:         0 cascade [4]         irq 10:      12236 usb-uhci
irq  3:  14023393 serial              irq 11:       5658 es1371
irq  4:  21183769 serial              irq 12: 143313718 PS/2 Mouse
irq  6:       161                     irq 14:    2437954 ide0
irq  7:   4818744 serial              irq 15:    6405449 ide1
```

Fig. 2.4. Output of the Linux `procinfo` command

Measurement Tools and Techniques II

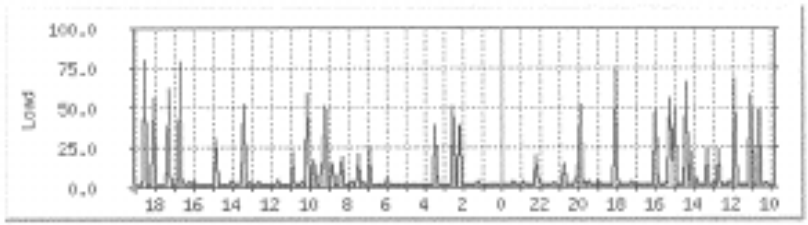

Fig. 2.5. Screenshot of an MRTG strip chart showing load average as a time series. The intervals on the time axis are the reverse of normal convention (cf. Figs. 3.1 and 2.6)

Measurement Tools and Techniques II

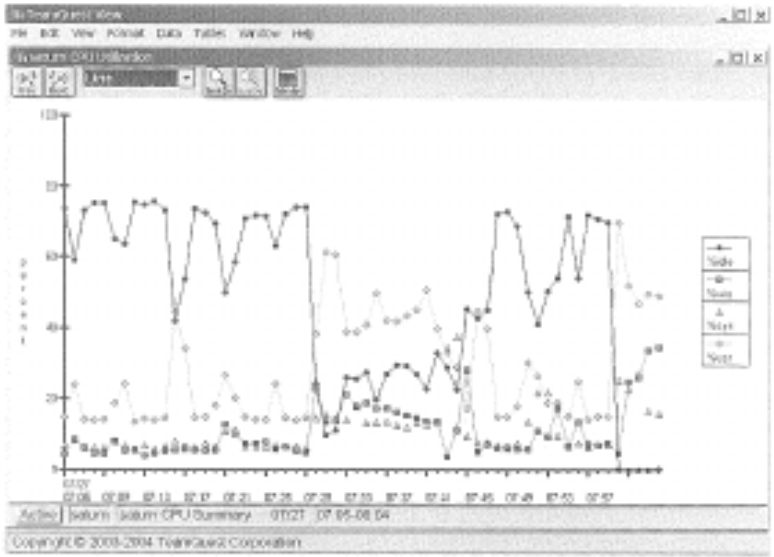

Fig. 2.6. Screenshot of CPU utilization components showing how the intervals on the time axis increase correctly from left to right in TeamQuest View (used with permission)

Measurement Tools and Techniques II

Table 2.2. K-S parameters for exponential fit of ranked data

n	Data	Ranked	Empl.	Theory	F_n	F_{n-1}	D_+	D_-
1	11.72	1.43	0.1220	0.1797	0.1667	0.0000	-0.0130	0.1797
2	10.43	4.12	0.3515	0.4348	0.3333	0.1667	-0.1015	0.2682
3	8.02	7.58	0.6468	0.6500	0.5000	0.3333	-0.1500	0.3167
4	7.58	8.02	0.6843	0.6707	0.6667	0.5000	-0.0040	0.1707
5	1.43	10.43	0.8899	0.7642	0.8333	0.6667	0.0692	0.0975
6	4.12	11.72	1.0000	0.8027	1.0000	0.8333	0.1973	-0.0306

Measurement Tools and Techniques II

Table 2.3. Critical K-S statistics for exponential fit

Statistic	Value
N	6
D_{max}	0.3167
K_{crit}	0.7757
p-value	0.7632
α	0.2368

Measurement Tools and Techniques II

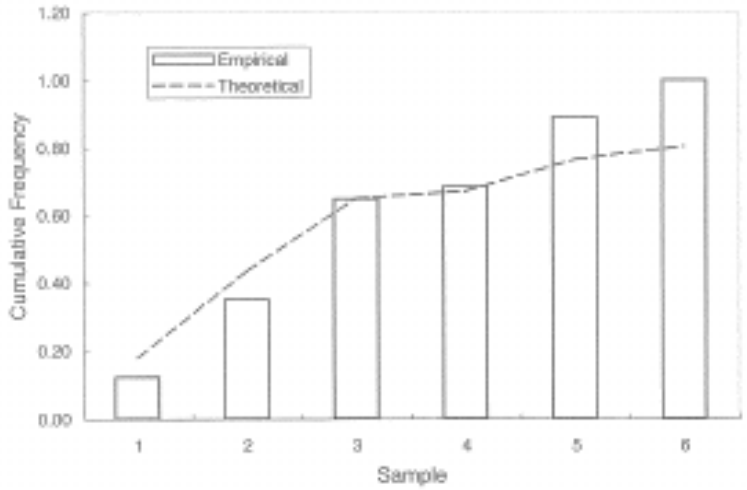

Fig. 2.7. Exponential fit to cumulative frequency data

Measurement Tools and Techniques II

Listing 2.2. Exponential variate generator

```
#! /usr/bin/perl
# genexp.pl

$x = 1; # Seed the RNG

# Generate 20 EXP variates with mean = 5
for ($i = 1; $i <= 20; $i++) {
    printf("%2d\t%6.4f\n", $i, exp_variate(5.0));
}

sub exp_variate {
# Return an exponential variate.
# log == Ln in Perl.
    my ($mean) = @_;
    return(-log(rand_num() / $mean));
}

sub rand_num {
# Portable RNG
# Return a (pseudo) random number between 0.0 and 1.0
    use integer;
    use constant ac =>      16807;  # Multiplier
    use constant mc => 2147483647;  # Modulus
    use constant qc =>     127773;  # m div a
    use constant rc =>       2836;  # m mod a
    my $x_div_q;                    # x divided by q
    my $x_mod_q;                    # x modulo q
    my $x_new;                      # New x value
    $x_div_q = $x / qc;
    $x_mod_q = $x % qc;
    $x_new = (ac * $x_mod_q) - (rc * $x_div_q);
    if ($x_new > 0) { $x = $x_new; }
    else { $x = $x_new + mc; }
    no integer;
    return($x / mc);
}
```

Measurement Tools and Techniques I