

NAME

parasrvn – simulate stochastic rendezvous networks.

SYNOPSIS

```
parasrvn [ -dMnpRvV ] [ -A time[,precision[,skip]] ] [ -B blocks[,time[,skip]] ] [ -e (e|i|w) ] [ -m file ]
[ -H alpha ] [ -o file ] [ -P pragma ] [ -S seed ] [ -t traceopts ] [ -T time ] [ -z specialopts ] [ filename ...
]
```

DESCRIPTION

Parasrvn is used to simulate stochastic rendezvous networks using the PARASOL simulation system. Refer to “*The Stochastic Rendezvous Network Model for Performance of Synchronous Multi-tasking Distributed Software*” for details of the model. Refer to “*SRVN Input File Format*” for a complete description of the input file format for the programs. See below for program restrictions and limits.

Parasrvn reads its input from *filename* specified at the command line if present, and from the standard input otherwise. Output for an input file *filename* specified on the command line will be placed in the file *filename.out*. If several files are named, then each is treated as a separate model and output will be placed in separate output files. If input is from the standard input, output will be directed to the standard output. The file name ‘-’ is used to specify standard input.

The **-o** *output* option can be used to direct all output to the file *output* regardless of the source of input. If multiple input files are specified, all output is concatenated to *output*. Output can be directed to standard output by using **-o-** (i.e., the output file name is ‘-’.)

OPTIONS

- d** Turn on debugging output. Dump task and entry information showing internal index numbers. This option is useful for determining the names of the servers and tasks since the parasol output routines do not emit this information at present. Output is directed to stdout unless redirected using **-m** *file*.
- M** Do not merge the send-no-reply waiting time with the rendezvous delay in the parseable output. The default in this release is to merge the results because some tools do not support the new fields.
- n** Read input, but do not solve. No output is generated.
- p** Generate parseable output suitable as input to other programs such as **srvndiff** and **multisrvn**. The parseable output is sent to an output file, one for each input file specified, determined from its corresponding input file by the addition of **.p**. If standard input is used as the input source, then the parseable output is output to the standard output device. In this case, only parseable output is emitted. If the **-o** *output* option is used, the parseable output is sent to a file whose name is derived from *output* by the addition of the suffix **.p**. If a suffix already exists on *output* then that suffix is replaced.
- R** Print the values of the statistical counters to the monitor file. If the **-A** or **-B** option was used, the mean value, 95th and 99th percentile are reported. At present, statistics are gathered for the task cycle time, utilization, and waiting time for messages, and for entry cycle time and utilization.
- v** Display solution execution. For **parasrvn**, the simulation block number and mean confidence level are displayed.
- V** Display version number.
- A** *time[,precision[,skip]]*
Use automatic blocking with a simulation block size of *time*. The *precision* argument specifies the desired mean 95% confidence level. By default, *precision* is 1.0%. The simulator will stop when this value is reached, or when 30 blocks have run. *Skip* specifies the time value of the initial skip period. By default, its value is 0.
- B** *blocks[,time[,skip]]*
Use manual blocking with *blocks* blocks. *blocks* must be less than or equal to 30. The run time for each block is specified with *time*. *Skip* specifies the time value of the initial skip period. Statistics gathered during the skip period are discarded. By default, its value is 0. When the run

completes, the results reported will be the average value of the data collected in all of the blocks. If the **-R** flag is used, confidence intervals will be reported to the *monitor* file.

-e (a|i|w)

Floating point error disposition. By default, floating point errors are ignored (**-ei**). Selecting **-ea** causes **parasrvn** to abort on the floating point errors *overflow*, *division by zero*, or *operand error* (see **floating_point(3)**). **-ew** prints a warning only on floating point error (not implemented at present).

-m file Direct all output generated by the various debugging and tracing options to *file* rather than to *stdout*. A filename of *-* directs output to standard output.

-H alpha

Activates recording of service time distributions. Bin size and histogram range is defined by three parameters which are:

– entry service time which defines the middle point of the histogram range

- max service time which is overloaded to mean the distance from the middle to the upper and lower boundary of the histogram.

- *alpha* which is multiplied with max service time to allow for more flexibility in sizing the histogram.

The number of bins in any given histogram is set to 20. Knowing the upper and lower range, and the number of bins, a bin size can then be calculated.

Distribution results are only displayed in the .out file (i.e. the human readable file). The usage of Activities and distribution reporting has not been tested.

-o file Direct analysis results to *output*. A filename of *-* directs output to standard output.

-P pragma[=value]

Set *pragma* to value. See the section on pragmas below.

-S seed

Set the initial seed value for the random number generator. By default, the system time from **time(3)** is used. The same seed value is used to initialize the random number generator for each file when multiple input files are specified.

-t traceopts

Tracing options.

driver Print out the underlying tracing information from the Parasol simulation engine.

processor

Trace activity for processors whose name match *pattern*. If *pattern* is not specified, activity on all processors is reported. *pattern* is regular expression of the type accepted by **egrep(1)**.

task Trace activity for tasks whose name match *pattern*. If *pattern* is not specified, activity on all tasks is reported. *pattern* is regular expression of the type accepted by **egrep(1)**.

event=*pattern*

Display only events matching *pattern*. The events are: msg-async, msg-send, msg-receive, msg-reply, msg-done, msg-abort, msg-forward, worker-dispatch, worker-idle, task-created, task-ready, task-running, task-computing, task-waiting, thread-start, thread-enqueue, thread-dequeue, thread-idle, thread-create, thread-reap, thread-stop, activity-start, activity_execute, activity_fork, and activity_join.

msgbuf Show *msgbuf* allocation and deallocation.

timeline

Generate events for the timeline tool.

-T *time*

Set the run time for the simulation. The default is 10,000 units. Specifying **-T** after either **-A** or **-B** changes the simulation block size, but does not turn off blocked statistics collection.

-z *specialopts*

This flag is used to select special options. Arguments of the form *nn* are integers while arguments of the form *nn.n* are real numbers.

print-interval=*nn*

Set the printing interval to *nn*. Results are printed after *nn* blocks have run. The default value is 10.

global-delay=*nn.n*

Set the interprocessor delay to *nn.n* for all tasks. Delays specified in the input file will override the global value.

PRAGMAS

scheduling

default

Use the scheduler built into parasol for processor scheduling. (faster)

custom

Use the custom scheduler for scheduling which permits more statistics to be gathered about processor utilization and waiting times. However, this option invokes more internal tasks, so simulations are slower than when using the default scheduler.

default-natural

Use the parasol scheduler; don't reschedule after the end of each phase or activity.

custom-natural

Use the custom scheduler; don't reschedule after the end of each phase or activity.

messages = nn

Set the number of message buffers to *nn*. The default is 1000.

halt-on-loss = {on,off}

In models with open queueing (open arrivals or asynchronous messages), arrival rates may exceed service rates. The simulator can either discard the arrival, or it can halt.

STOPPING CRITERIA

It is important that the length of the simulation be chosen properly. Results may be inaccurate if the simulation run is too short. Simulations that run too long waste time and resources.

Parasrvn uses *batch means* (or the *method of samples*) to generate confidence intervals. With automatic blocking, the confidence intervals are computed after running for three blocks plus the initial *skip* period. If the root or the mean of the squares of the confidence intervals for the entry service times is within the specified *precision*, the simulation stops. Otherwise, it runs for another block and repeats the test. With manual blocking, **parasrvn** runs the number of blocks specified then stops. In either case, the simulator will stop after 30 blocks.

Confidence intervals can be tightened by either running additional blocks or by increasing the block size. A rule of thumb is the block size should be 10,000 times larger than the largest service time demand in the input model.

MODEL LIMITS

The following table lists the acceptable parameter types for **parasrvn**. An error will be reported if an unsupported parameter is supplied unless the value is the same as the default.

Phases

3

Scheduling	FIFO, HOL, PPR, RAND
Open arrivals	yes
Phase type	stochastic, deterministic
Coefficient of variation	yes
Interprocessor-delay	yes
Asynchronous connections	yes
Forwarding	yes
Multi-servers	yes
Infinite-servers	yes
Max Entries	unlimited
Max Tasks	1000
Max Processors	1000
Max Entries per Task	unlimited

SEE ALSO

“The Stochastic Rendezvous Network Model for Performance of Synchronous Multi-tasking Distributed Software” by C.M. Woodside et.al.

“SRVN Input File Format” by Dorina Petriu et al.

srvndiff(1), egrep(1), floating_point(3)

BUGS

The format of the debugging output generated by the **-d** option requires a knowledge of the internals of **srvn** in order to be understood.

To promote harmony among users, and to simplify error processing, **parasrvn** creates a child process to perform the actual simulation. The child is automatically niced.