

NAME

`dgsh-parallel` – Create a semi-homogeneous `dgsh` parallel processing block

SYNOPSIS

`dgsh-parallel` [**-d**] **-f** *file* | **-l** *list* | **-n** *n command ...*

DESCRIPTION

`dgsh-parallel` creates and executes a `dgsh` block that invokes multiple times the specified command and its optional arguments. If the command or its options include the `{}` string, this is replaced by the numeric or string identifier associated with each invocation.

OPTIONS

- d** Allows the debugging of the generated script, by leaving it in the temporary directory and echoing its path on the standard error.
- f** *file* Obtain string arguments from the specified file: one argument per line. One command will be generated for each line in the file. Each command will have `{}` strings replaced with the contents of the corresponding line.
- l** *list* Obtain string arguments from the specified comma-separated list. One command will be generated for each list element. Each command will have `{}` strings replaced with the corresponding element.
- n** *n* Run *n* instances of the command. Each command will have `{}` strings replaced with the command's ordinal number, starting from 1.

EXAMPLES

Count in parallel the number of times each word appears in the specified input file(s). This sequence mirrors Hadoop's WordCount example.

```
# Scatter input
dgsh-tee -s |
# Run four instances of the command
# Emulate Java's default StringTokenizer, sort, count
dgsh-parallel -n 4 "tr -s ' \t\n\r\f' '\n' | sort | uniq -c" |
# Merge the four sorted counts
dgsh-merge-sum '<|' '<|' '<|'
```

SEE ALSO

`dgsh(1)`, `dgsh-tee(1)`,

BUGS

The interface between the generated script and its invokers is currently (December 2016) being polished.

AUTHOR

Diomidis Spinellis — <<http://www.spinellis.gr>>.