Distribution	Type	Parameters	Example	Description
Exponential	exp	rate (rate of the exponential distribution)	{type = "exp", params = {rate = 0.1}}	It generates an exponential random variable with average 10
Erlang	erl	k (number of stages) lambda (rate of each stage)	{type = "erl", params = {k = 4, lambda = 0.4}}	It generates a four stages Erlang random variable with average 10, and c.v. = 0.5
Deterministic	det	val (value returned)	{type = "det", params = {val = 10.0}}	It always generates 10
Uniform	unif	Min (min. value) Max (max. value)	{type = "det", params = {min = 5.0, max = 15.0}}	It generates a uniform random number between 5 and 15 (average 10)
Hyper Sum of Erlangs	HSE	{<BLOCK/>, ...} a sequence of 1 or more blocks describing a probabilistic choice	{type,= "HSE"„params = {{0.5,{k = 2, lambda = 0.5}„{k = 4, lambda = 1}}„{0.5,{lambda = 1/15.0}},},}	It generates a random variable that is: • the sum of the Erlangs: Erlang(2, 1/2)+Erlang(4, 1) 50% of the times; • exponential, with rate 1/15 50% of the times
PH	Phase Type distribution	alpha, Q	alpha = 1.0, 0.0, 0.0, Q = -0.2, 0.2, 0.0, 0.0, -0.4, 0.2, 0.0, 0.0, -0.2	Generates a random number that follows a PH distribution with 3 states. Always starst from state 1, goes to state 2 after 1/0.2, then It either finish or goes to state 3 after 1/0.4, with both alternatives equally prob.

Table 1: Probability distributions supported by dagSim

1 Appendix - dagSim

The following sections provide the guidelines i) to install dagSim, ii) to interpret the output produced and iii) understanding the LUA configuration file format given in input to the tool. dagSim can be freely downloaded; supported platforms are currently *Linux* and *MacOs*.

dagSim is located under https://github.com/eubr-bigsea/dagSim repository. It requires GCC compiler (version 6 or newer) and LUA interpreter (version 5.3 or newer).

Installation procedure and usage

The installation procedure consists of the following steps:

- Extract the compressed package in a folder;
- Compile dagSim by using the "make" utility.

The usage is the following:
./dagSim < $configurationfile.lua$ >

The configuration file

The input to dagSim is a LUA file representing a DAG model. As an example, Figure 1 refers to a Tez job corresponding to Q1 query.

The configuration file modeling the DAG starts by specifying M1, M4, R2, R3 and R5 stages. For each stage, it is necessary to specify a few attributes, such as: i) the number of tasks, ii) the distribution type (int this case, exponential) and iii) the set of input and output stages to and from the current one. A distribution (**distr)** is characterized by different parameters. The probability

distributions currently supported are shown in Table 1. including the possibility of replaying from an external file.

Other properties denote the following: i) the number of computation nodes in the system, ii) the number of users accessing the system, iii) the distribution of the think time for the users, iv) the total number of jobs to simulate (this value can be incresead to reduce the simulation error at a cost of increasing execution time), and finally v) the coefficient for the Confidence Intervals.

Output

The output is currently generated in text format including different columns organized as follows:

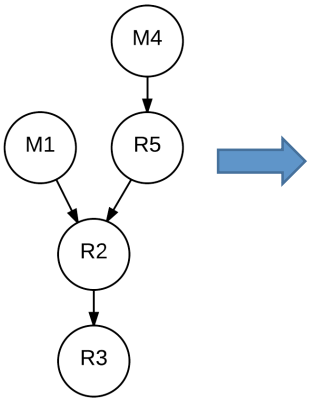
- Class type (where `0` value refers to the whole system, and `1` indicates one of DAG stages).
- Stage id (when class type is different than `0`);
- Time in ms;
- Standard Deviation;
- Confidence interval bounds;
- Error.

The DAG stage measurements rows refer to:

- The average start time of the stage corresponding to the beginning of a job;
- The average end time of the stage corresponding to the end of a job;
- The average stage execution time.

The system measurements rows refer to:

- The task total execution time;
- The task total waiting time before the job starts (note that dagSim supports general closed models with multiple users);
- The job execution time.



- **Stages** = {
- { **name** = "M1", **tasks** = 10, **distr** = {**type** = "exp", **params** = {**rate** = 1/1000.0}}, **pre** = {}, **post** = {"R2"} },
- { **name** = "M4", **tasks** = 10, **distr** = {**type** = "exp", **params** = {**rate** = 1/1500.0}}, **pre** = {}, **post** = {"R5"} },
- { **name** = "R5", **tasks** = 10, **distr** = {**type** = "exp", **params** = {**rate** = 1/500.0}}, **pre** = {"M4"}, **post** = {"R2"} },
- { **name** = "R2", **tasks** = 10, **distr** = {**type** = "exp", **params** = {**rate** = 1/500.0}}, **pre** = {"M1","R5"}, **post** = {"R3"} },
- { **name** = "R3", **tasks** = 10, **distr** = {**type** = "exp", **params** = {**rate** = 1/500.0}}, **pre** = {"R2"}, **post** = {} }};
- **Containers** = 20 **Users** = 5;
- **UThinkTimeDistr** = {**type** = "exp", **params** = {**rate** = 1/10000.0}};

Figure 1: A DAG corresponding to a Tez job.

SYSTEM

0.0	0.0	26826.779593111	2335.3573269241	26762.046790618	26891.512395603	0.0048259838470664
0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	2.0	26826.779593111	2335.3573269241	26762.046790618	26891.512395603	0.0048259838470664
0.0	3.0	10157.68622371	9946.4179091541	9882.013151198	10433.359296223	0.054278713959278

STAGES

1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	1.0	1.0	13857.596874517	1527.4407734546	13815.258376349	13899.935372685	0.0061105108701297
1.0	1.0	2.0	13857.596874517	1527.4407734546	13815.258376349	13899.935372685	0.0061105108701297
1.0	1.0	3.0	1003.650733786	1003.1982107999	1000.8700102157	1006.4314573563	0.0055412176301918
1.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	2.0	1.0	15898.562382447	2048.7193750987	15841.774778622	15955.349986271	0.0071437407306231
1.0	2.0	2.0	15898.562382447	2048.7193750987	15841.774778622	15955.349986271	0.0071437407306231
1.0	2.0	3.0	1498.8907845702	1501.7728726614	1494.7280825445	1503.0534865959	0.0055543767011315
1.0	3.0	0.0	15898.562382447	2048.7193750987	15841.774778622	15955.349986271	0.0071437407306231
1.0	3.0	1.0	19221.692391137	2140.9217273581	19162.349068498	19281.035713776	0.0061746199482848
1.0	3.0	2.0	3323.1300086906	673.94457245411	3304.4492182195	3341.8107991618	0.011242888735819
1.0	3.0	3.0	500.58282091734	501.25267140606	499.02942279496	502.13621903973	0.0062063580989163
1.0	4.0	0.0	19233.5805769	2132.5614229704	19174.468990075	19292.692163724	0.0061467064427284
1.0	4.0	1.0	23023.578395554	2234.3645255399	22961.644970694	23085.511820414	0.0053799999110402
1.0	4.0	2.0	3789.9978186546	661.80854682466	3771.6534216513	3808.3422156578	0.0096804261537775
1.0	4.0	3.0	499.38845555516	499.69989295022	498.00335811834	500.77355299199	0.0055471744347167
1.0	5.0	0.0	23023.578395554	2234.3645255399	22961.644970694	23085.511820414	0.0053799999110402
1.0	5.0	1.0	26826.779593111	2335.3573269241	26762.046790618	26891.512395603	0.0048259838470664
1.0	5.0	2.0	3803.2011975562	661.46060565895	3784.86644499	3821.5359501223	0.0096417473669025
1.0	5.0	3.0	500.20251468379	500.3166489238	498.81570768663	501.58932168095	0.0055449821080541
			Measure	**Standard dev.**	**Lower Bound**	**Upper Bound**	**Error**

Figure 2: dagSim Output