Analytical Performance Models of Robotic Storage Libraries: Status Update

Theodore Johnson

AT&T Labs - Research

johnsont@research.att.com

New Results

- Improved model of a robot arm
- Improved models of tape drives
- Experimental performance validation
- Perl / Excel interface.

Background

- *Queuing models of Tertiary Storage*
  - 1996 Mass Storage Systems and Technologies
- Developed queuing model of access to a robotic storage library
  - Batch arrivals (multiple files / media per request)
  - Multiple drives
  - Queuing at the robot arm and at the drives
  - Extensions to closed queuing networks.
- Library written in C, available at www.cise.ufl.edu

Improved robot arm model

- Previously : *M/G/1* queue with batch arrivals.
  - Reasonable accurracy for low utilization.
- Problem in validation study
  - Single-threaded media manager.
  - Service by robot equivalent to unwind, eject, return to shelf, fetch, mount.
  - Media manager became the bottleneck.
- New model
  - Limited customer population : number of queued requests bounded by number of drives.
  - Developed a finite state Markov chain model.

c : number of drives

b : number of busy drives

i : number of idle drives

q : number of queued drives

$b+i+q = c$

E_{tr} : Robot arm service time

E_{dr} : drive service time

l : arrival rate

P_q : Probability that a request is queued for a drive

$P_{i,j}$: Probability that a batch of size j arrives when there are i idle drives.

Represent a state by the number of busy and idle drives (b,i).
Compute $P_{i,j}$ to be the job batch distribution, truncated to a maximum of i arrivals.
Compute P_q to be the probability that at least 1 job is waiting for a drive.

cause	from	to	rate	condition
robot completes service	(b,i)	(b+1,i)	$1/E_{tr}$	$b<c-i$
drive competes service, no queue	(b,i)	(b-1,i+1)	b/E_{dr}	$i>0$, $b>0$
drive completes service, no queue	(b,i)	(b-1,i+1)	$(1-P_q)b/E_{dr}$	$i=0$, $b>0$
drive completes service, queued requests	(b,i)	(b-1,i)	$P_q b/E_{dr}$	$i=0$, $b>0$
job arrival	(b,i)	(b,i-j)	$P_{i,j} l$	$j<=i$

Improved models of tape drives

- Benchmarking study of a range of tape drives that use a range of technologies.
  - See the related paper in these proceedings.
- Models of tape drives
  - Serpentine
    - Each seek is across 1/3 the tape.
  - Linear
    - Each seek is across 1/k of the tape, where k files are being loaded.

Validation Study

- Storagetek 9710 with four DLT4000 drives.
- Single-threaded media manager.
- Wrote test data to 40 tapes.
- Limited batch size to a maximum of 15 tapes.
- Generated synthetic job mix
  - Exponential inter-arrival time
  - Number of media, number of files per media has geometric distribution.
  - FCFS queueing.
- One sample run per setting of parameters
  - Long-running experiment, short period of exclusive access.

Validation Study

Avg time between arrivals	Avg. media per job	Avg. files per media	Number of drives	Experimental response time	analytical response time	percent difference
1000 sec.	2	5	3	3438 sec.	3571 sec.	3.8%
750	2	3	4	1473	1695	15
1000	2	3	4	1283	1297	1.1
1000	4	3	4	2813	3273	16
1400	3	3	3	2755	2247	18
1000	2	3	3	1293	1340	3.6
1200	4	3	4	3432	2473	28

Perl / Excel interface

- Perl is a powerful public-domain scripting language.
  - www.cise.ufl.edu
- We developed wrappers for calling the solver and for displaying the results.
- Easy to specify parameters and new analyses.
- Interface to OLE calls
  - Automatically create and load an Excel spreadsheet with data and create a chart.

```
#                                          Perl code : plot response time and utilization against arrival rate.

use rsl;

$lambda_start=.0001;    $lambda_step=.0001;    $iter=19;

$Atlparam = {       lambda => .0002,

                        drives => 4,
                        robots => 1,
                        E_robot => 20,
                            V_robot => 2,
                        mount => 20,
                        seek_full => 100,
                        seek_settle => 2,
                        xfer_rate => 10,
                        E_rewind => 40,
                        V_rewind => 50,

                        files_request => 5,
                        media_request => 4,
                        E_file_size => 10,
                        V_file_size => 10,
                      };

$answer_list = rsl::scan_open("lambda",$lambda_start, $lambda_step, $iter,$Atlparam);

rsl::print_answer_list($answer_list,"lambda","robot_utilization","job_response_time");

($xl,$r,$c) = rsl::xl_print_answer_list($answer_list, "lambda","job_response_time","robot_utilization","drive_util");
rsl::xl_plot($xl,$r,$c);
```

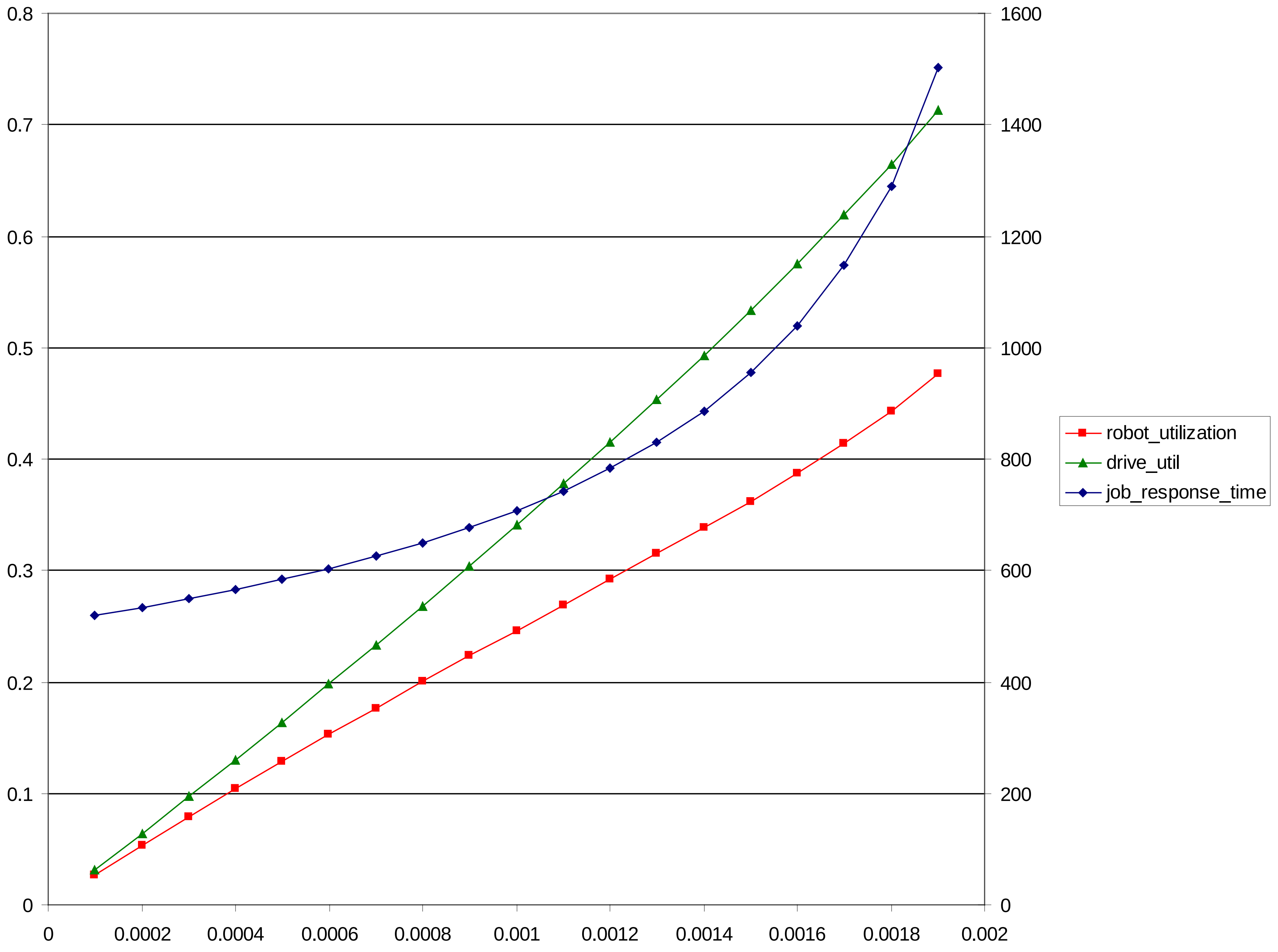
```
sub xl_print_answer_list{
  my $answer_list = shift;
  my @print_list = @_;

#   Creates OLE object to Excel
  $ExcelApp = CreateObject OLE "excel.application" || die "Unable to create Excel Object: $!\n";
  $ExcelApp->{'Visible'} = 1;
  $ExcelApp->Workbooks->Add();

#                     Print column headings
   for($col=1;$col < scalar(@print_list); $col++){
      $ExcelApp->Cells(1,int($col+1))->{'Value'} = $print_list[$col];
   }

   $row = 2;
   for($i=0;$i<scalar(@{$answer_list});$i+=2){
       $col = 1;
       for($j=0;$j<scalar(@print_list);$j++){
         $key = $print_list[$j];
         if(defined($answer_list->[$i]->{$key})){
           $ExcelApp->Cells(int($row),int($col))->{Value} = $answer_list->[$i]->{$key};
         }
         if(defined($answer_list->[$i]->{$key})){
           $ExcelApp->Cells(int($row),int($col))->{Value} = $answer_list->[$i]->{$key};
         }
         $col++;
       }
       $row++;
   }

   return ($ExcelApp,$row-1,$col-1);
}
```


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1600
1400
1200
1000
800
600
400
200
0
0
0.0002
0.0004
0.0006
0.0008
0.001
0.0012
0.0014
0.0016
0.0018
0.002
robot_utilization
drive_util
job_response_time