

CS/ECE 541 Project # 1, Due in class October 29, 2009

1 Introduction

In this assignment, your job is to model the Pegasus project. The purpose of Pegasus is to provide astronomers web access to data from a world class telescope. The telescope is located on a remote mountain top. It collects astronomy observations and sends them to a centralized web server (the Pegasus web server) to be processed and stored in a database. Processing includes calibrating the observations and collecting a variety of statistics from the observations. Astronomers also have access to the Pegasus web server. Astronomers make requests that require the web server to search for and extract certain data, perform various transformations on that data, and return the result to the astronomer. The Pegasus architecture is depicted in Figure 1.

2 System Description

2.1 Astronomy Clients

Astronomy clients submit requests to the Pegasus web server at a rate of λ . 80% of the time those requests are for one data unit, and 20% of the time they are for two data units.

2.2 The Telescope Site

The telescope site is made up of one telescope and one hard disk. The telescope performs two tasks: observation and position rotation. An observation gathers data from the sky. Observations are measured in data units, and each observation is 10 data units in size. While the telescope is making an observation, it stores a data unit worth of observation to the local hard disk as soon as that data unit has been observed. (In other words, the telescope does not have to wait for the entire observation to finish before it begins writing data to the local hard drive.) The collection of each data unit in an observation lasts for an average of τ_{obs} .

As mentioned earlier, telescopes also perform position rotations. Rotations occur after each observation and reposition the telescope for its next observation. The telescope rotates clockwise between 9 positions. On average, it takes τ_{rot} time to perform a rotation.

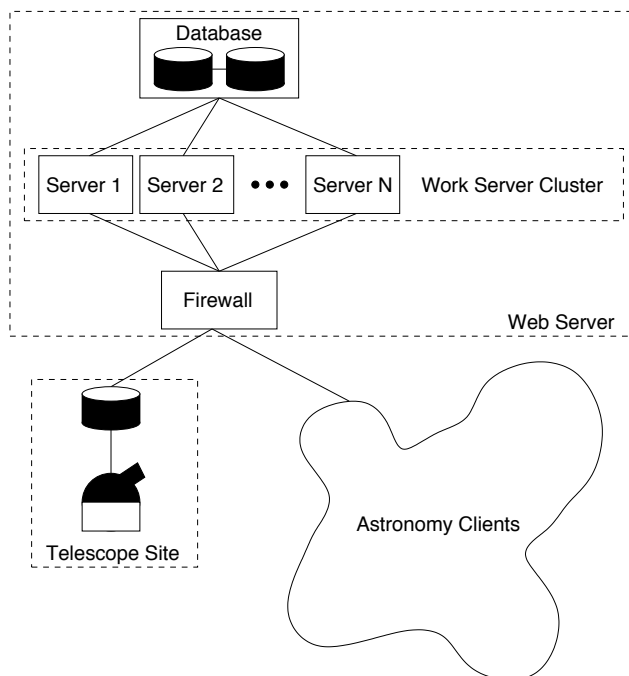


Figure 1: Pegasus Architecture

In addition to the telescope, the telescope site contains a hard disk. The purpose of the hard disk is to buffer data from the telescope, before it can be sent to the Pegasus web server. Data transfer to the web server occurs concurrently with telescope operation. Data transfer rates are considered negligible for this assignment. At any one time, the hard disk can buffer up to 50 data units. If the hard disk ever fills up and the telescope is in observation mode, the observation must be halted until space frees up on the hard disk.

Unfortunately, the telescope site experiences failures. Those failures occur somewhat frequently due to the nature of the telescope's remote location. Network failures can occur that prevent the telescope site from sending data to the Pegasus web server. Network failures occur at a rate of α_n . Network failures are remedied at a rate of γ_n . Also, the telescope-hard drive combo suffers power failures at a rate of α_p . Power failures prevent the telescope from performing any tasks and prevent the local hard disk from transferring data to the Pegasus web server. Power failures are fixed at a rate of γ_p . Finally, the telescope contains a triply redundant critical component. If all three replicas of the critical component fail, the telescope is inoperable until at least one of them is repaired. The critical components each fail at a rate of α_{cc} . The telescope site is monitored by a single repairman who begins repairs as soon as a component fails. The repairman repairs a critical component at a rate of γ_{cc} . Fortunately, the hard disk at the telescope site is highly reliable, and we do not consider direct failures of the hard disk.

2.3 The Pegasus Web Server

The Pegasus web server consists of a firewall, a reliable database, and a cluster made up of N work servers. The firewall controls all access from the outside world as well as the scheduling of

Name	Description	Value
λ	astronomy client arrival rate	1 per minute
τ_{obs}	average time to observe 1 data unit	1 minute
τ_{rot}	average time to rotate 1 position	5 minutes
α_n	failure rate of the telescope site link to the web server	1 per 2 weeks
γ_n	repair rate of the telescope site link to the web server	1 per 6 hours
α_p	telescope site power failure rate	1 per month
γ_p	telescope site power repair rate	1 per day
α_{cc}	telescope critical component failure rate	1 per month
γ_{cc}	telescope critical component repair rate	1 per 2 days
μ	work server processing rate	1 data unit per 5 minutes

Table 1: System Parameters

the work servers. The work servers process incoming data from the telescope and store it in the reliable database. The work servers also search for and extract data from the database, and process it to fulfill client requests. Whether for a client request or for telescope data, each work server can process one data unit at a time, at a rate of μ .

Scheduling of the work servers operates as follows. Telescope data is processed ahead of astronomy client requests. Scheduling is performed at the granularity of a data unit. If there are no available work servers, then the firewall can not schedule the processing of anymore data units until a work server becomes free. In that case, the firewall can buffer one and only one client request, regardless of the size of the request¹. When the firewall buffer has been emptied, it can be used for the next client request, if needed.

2.4 Model Parameters

All times in this project are exponential unless stated otherwise. Any timing aspects of the system that have not been detailed are assumed to be instantaneous. All failures are independent unless specified otherwise. A list of model parameters can be found in Table 1. In that table, a month is considered to be 28 days.

3 System Rewards

The goal of this assignment is to evaluate several rewards in steady-state, while varying N , the number of work servers in the web server's cluster. The value of N should be varied from 1 to 10. The following rewards should be computed.

- average utilization of a work server
- availability of the Pegasus web service to astronomy clients

¹Telescope data does not need to be buffered at the firewall because it is already buffered at the telescope site.

- the fraction of time that the telescope is performing useful work, i.e., either rotating or observing
- utilization of the telescope site hard disk
- availability of the network link between the telescope site and the Pegasus web server
- the effective client arrival rate
- the effective combined processing rate of all work servers
- the effective telescope data generation rate

4 Möbius Details

To complete this homework assignment, you will use the Möbius modeling tool. All of your atomic models are to be created using Stochastic Activity Networks. Below, several Möbius features are described that are needed for this assignment. Not every aspect of those features is described here. So, if you need more information regarding those features, please consult the Möbius user's manual. Möbius features that were covered in previous homework assignments will not be covered here.

4.1 Case Probabilities

In this assignment you will need to use case probabilities. Case probabilities model uncertainty regarding the completion of an activity. Case probabilities channel a fired event into one of several categories, according to a specified distribution. Each of those categories can be accessed through circles on the right hand side of an activity.

Consider a system with a failure rate of β , in which 30% of the failures are hardware failures and 70% are software failures. A case probability can be used to model that scenario as follows. An activity is created with rate β , and it is defined to have two cases. (The number of cases is accessed by editing the activity). Also in the definition of the activity, the first case is given a .7 probability, and the second case is given a .3 probability. (The probabilities of each case are accessed by editing the activity). When the activity appears in the model, it will have two circles on its right side. Objects² connected to the top circle are consulted when a hardware failure occurs, and objects connected to the bottom circle are consulted when a software failure occurs.

4.2 Extended Places

In addition to normal places, Möbius supports the used of extended places. Extended places use types other than the standard integers that normal places use. Extended places can be of types such as chars, shorts, floats, doubles, structs, and arrays. In order for an extended place to have a struct or array type, that type must first be defined. A dialog allowing the definition of a struct or array type can be accessed from the edit menu of the SAN editor. Once the type is defined, the type of

²Objects refer to places and gates.

an extended place can be accessed by editing the place. Given an extended place X , with a struct type and a field Y , Y can be accessed using $X \rightarrow Y \rightarrow \text{Mark}()$. Given an extended place Z , with an array type, the i^{th} element can be accessed using $Z \rightarrow \text{Index}(i) \rightarrow \text{Mark}()$. You are required to use an extended place with a struct type to track the operational state of the telescope in this assignment.

4.3 Steady-State Rewards

Previously, you solved Möbius models for transient solutions. In this homework, you are solving for results in the steady-state. In order to get steady-state results, you must specify your rewards to have a steady-state type. The type can be set from the time tab of each reward variable. When the type is set, additional parameters will appear in the tab. However, those parameters can be ignored because they are only required for simulation.

4.4 Studies

In previous assignments, you created default range studies. In this assignment, you will use your range study to vary the value of N . In order to achieve this, you need to define a global variable for N in each of your models and rewards that use N . To define a global variable, a dialog can be accessed from the edit menu of a model or reward. Once a global variable is defined, its name can be used whenever its value is required. Each unique global variable that you define will show up when you create your study. In the study editor, you can give each global variable a value. For N , give it an incremental value that goes from 1 to 10.

4.5 Solvers

For this assignment, you will use the normal state space generator and the iterative steady-state solver.

5 Suggestions for Completing this Assignment

The following items are suggestions for completing this assignment.

- START EARLY!
- Determine which aspects of the problem statement are important and which are not.
- Before building your model, decide what aspects of the system should be tracked by the state of your model and what aspects of the system represent events/activities that will occur in your model.
- Develop and debug your model with N set to 1. Setting N to 1 will reduce the time it takes to solve your model

- The verbosity of the solver determines how often solution updates are given in the results file. If the verbosity is set to $x > 0$, then every x iterations the solver will update the results file with the current accuracy level of the solution. That accuracy level is labeled maximum difference. When the maximum difference reaches the predetermined accuracy level specified to the solver³, the solver terminates. Thus, the verbosity can be used to track the progress of the solver.
- Leave time to solve your model for all values of N . On a 1.67 GHz Athlon processor, the ISS solver took approximately 1 minute to solve the model when N was 1, and 35 minutes when N was 10.
- If you have problems with the size of your state space, you can change the state space generation build type to normal and increase the trace level. That will create trace files that will you to see what states are being generated.
- If you are getting compiling errors with your models, look at the Mobius generated cpp files if you have to.
- Note that integer division may result in rates of 0, which may lead to absorbing states.
- Note that impulse rewards are not required for this homework.
- Note that the firewall buffer is only for client data, not telescope data.
- Note that incoming client data can not be buffered at the firewall until ALL buffered data has been processed.

6 Submission Instructions and Special Requirements

6.1 Special Requirements

As mentioned in Section 4.2, you are required to use an extended place with a struct type to track the operational state of the telescope.

6.2 Submission Instructions

In this section, we describe submission instructions. We start by describing several general guidelines.

- The items to submit and the order in which they should be submitted are described later. Please do not turn in extraneous items unless they are essential to your solution. For example, please do not submit print outs of the Möbius results files. Instead, present your results in the specified formats.
- Please join your entire assignment together with a staple or paper clip.

³The default accuracy level is $1 * 10^{-9}$.

- For this assignment, please type your solution.

Now, we list the items to submit in class (hard copy) and the order in which they should be submitted.

1. Submit a description of your solution. The purpose of the description is so the grader understands how your model works and why you designed it the way you did. The description should include, but is not limited to:
 - an explanation of any atomic models, composed models, and rewards
 - an explanation of how rewards were derived, that were not directly computed with Möbius
2. Submit Möbius model documentation. That documentation should cover every component of your model, with the exception of the state space generator and the numerical solver. If you create a model A and copy it to a model B , and both are used in your solution, then submit documentation for both, even if the two models are identical. In your documentation, you are responsible for making sure everything makes it to your print outs. If something is missing, write it in by hand.
3. Submit in table form, the results of each reward variable versus the value of N . Be sure to identify what data each table is detailing. When applicable, be sure to specify units for the data in your tables.
4. Submit in graph form, the results of each reward variable versus the value of N . Be sure to identify what data each graph is depicting. When applicable, be sure to specify units for the data in your graphs. If there are any REAL asymptotes, display them on your graphs as well. All of the effective rates have real asymptotes.
5. Submit a graph that plots state space size versus the number of iterations that the iterative steady state solver took to solve each of your experiments. (You should have 10 data points here, one for each value of N .)
6. For each graph that you submitted, describe your results. Try to give explanations of why your results came out the way they did. Descriptions such as the reward variable increases as N increases will not be scored very highly. Your descriptions do not have to be extensive, but they should demonstrate some thought and insight on your part.
7. Submit any other materials that you feel are needed for your solution.
8. ELECTRONIC SUBMISSION: Please archive your Möbius project (select the minimal archive option), and submit the resulting .tar.gz file using Compass. The project submitted electronically should match the model documentation and results submitted on paper.