Prophasis - An IT Infrastructure Monitoring Solution - Interim Report

Cameron Gray

Fourth Year Project Report

School of Informatics

University of Edinburgh

2016

Abstract

Prophasis is an IT infrastructure monitoring system that is designed to suit small to medium size businesses where a system needs to be intuitive to manage. Management of the entire system can therefore be handled from a single, responsive web interface. It is also suitable as a one-stop tool with support for both time series monitoring in addition to real time alerting. Traditionally two different tools would be needed to gain this level of monitoring.

Table of Contents

Chapter 1

Interim Report

1.1 Current Progress

Stuff wot i dun

1.2 Future Work

Stuff wot i need 2 do

Chapter 2

Introduction

2.1 Background

In recent years, almost all businesses have been expanding their IT infrastructure to handle the modern demand for IT systems. As these systems grow and become increasingly important for business operation it is crucial that they are sufficiently monitored to prevent faults and periods of downtime going unnoticed. There is already a large market of tools for monitoring IT systems however they are designed for use on massive scale networks managed by teams of specialised systems administrators. They are therefore complicated to set up and manage and multiple tools are often required to gain a suitable level of monitoring.

For example, tools generally either fall into the category of real time alerting (i.e. telling someone when something breaks) and time series monitoring (i.e. capturing data about the performance of systems and presenting graphs and statistics based on it), there is a large gap in the market for tools that provide both of these in one package. This reduces the time required to manage the system as it eliminates the need to set up and configure two completely separate tools.

These tools are also generally managed and configured through various configuration files split across different machines on the network. This means that in order to efficiently use these tools a configuration management system such as Puppet must be used. In a small business with limited IT resources, a completely self contained system is often preferable.

2.2 Improvements

Prophasis is designed for use in a small to medium business with limited IT resources. They may have a small IT team with limited resources or may not even have a dedicated IT team at all, instead relying on one or two employees in other roles who manage the business's IT systems on the side of their regular jobs. Therefore the system needs to be quick to deploy and manage with a shallow learning curve. In order to use the system efficiently there should be no requirement for additional tooling to be deployed across the company.

2.2.1 Configuration Management

It should be possible to manage the configuration of the system from a single location. Prophasis therefore provides a responsive web interface where every aspect of the system's operation can be configured, Prophasis then handles distributing this configuration to all other machines in the system in the background. Custom code for plugins is handled in the same way; it is uploaded to the single management machine and is then automatically distributed to the appropriate remote machines when it is required.

2.2.2 Time Series Monitoring & Real Time Alerting

Prophasis provides both the ability to alert administrators in real time when a fault is discovered with the system alongside functionality to collect performance metrics over time and use this data to generate statistics about how the system has been performing. This time series data can be used to both investigate the cause of a failure in post-mortem investigations in addition to being able to be used to predict future failures by looking at trends in the collected data.

2.2.3 Expandability

It is important that a monitoring tool can be expanded to support the monitoring of custom hardware and software. An example of this would be hardware RAID cards. Getting the drive health from these types of devices can range from probing for SMART data all the way to communicating with the card over a serial port. It is therefore crucial that Prophasis can be easily expanded to support custom functionality such as this. Therefore Prophasis supports a system of custom ”plugins” which can be written and uploaded to the monitoring server where they can then be configured

to monitor machines. These plugins are designed to be self contained and to follow a well defined and documented structure. This provides scope for a plugin ”marketplace” much like there already exists for plugins for software such as Wordpress and Drupal allowing plugins to be easily shared and installed for monitoring various pieces of hardware and software, therefore eliminating the need for every user to implement custom monitoring code for the systems they are using.

Chapter 3

Design

3.1 Technology Choice

3.1.1 Why Python?

3.1.2 Why HTTPS?

3.2 System Structure

- Explain different components
- Diagram
- Why separate?

3.3 Monitoring Methodology

3.3.1 Host Management

- Stuff about hosts and host groups

3.3.2 Plugins

- What is a plugin?

3.3.3 Checks

- What is a check?

3.3.4 Schedules

- What is a schedule?

3.3.5 Services

- What is a service?
- Dependencies & Redundancy Groups
- Why use dependencies?
  - Alert only if service functionality is severely impacted
  - Prevents unnecessary alerts due to failures in redundant infrastructure
  - Provides clearer view of impact of a given failure

3.3.6 Alerts

- What is an alert?
- Separate alerts for different checks/hosts/plugins/services
- State transition restrictions - Reduce unnecessary alerts

3.4 Plugin Interface

- Explain structure of a plugin
- UML Diagram?
- Why Lua for classification logic?