

# PQS Autonomic Computing

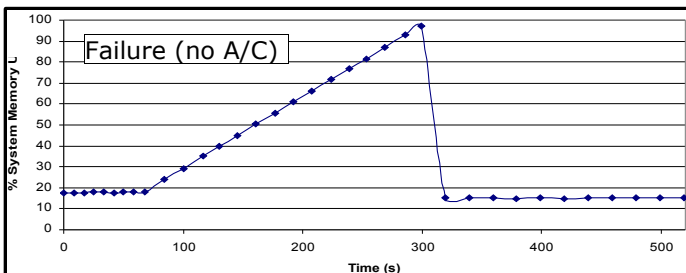
## Project description

### INTRODUCTION

At least half the cost of an IT infrastructure is in the manpower required for setup and maintenance. Configuring a network with workstations and servers, and keeping it running, usually requires lots of man-hours and expertise; something that can be quite expensive. As networks are getting larger, and systems more complex, this cost factor is expected to increase drastically, especially considering that fewer and fewer numbers of new and skilled administrators are joining the workforce each year.

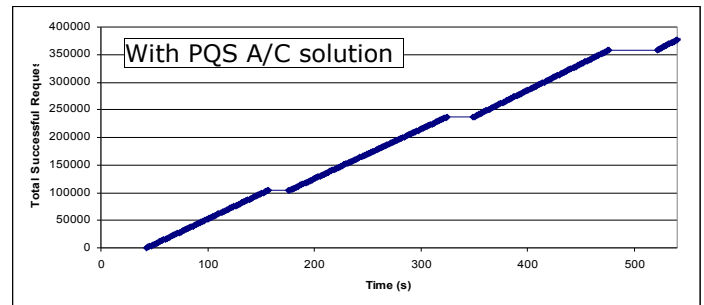
### THE APPLICATION

Traditional autonomic computing has been notoriously involved, requiring large amounts of processing power generally unavailable on the monitored systems. Our approach differs by using many indicators (both local to the monitored system, as well as remote throughout the network) and comparing with past performance metrics, as well as comparing with current performance metrics on similar systems in the network. This way misconfigurations can be caught, as well as security breaches, such as worms, viruses, or rootkits.



Simple system metrics such as resource usage (think CPU time, disk, I/O bandwidth), and network service response times are combined with security notices from virus scanners, intrusion detection systems, and firewall logs to determine the cause of deviant host or network behavior. In most cases our application will be able to quickly respond and mitigate a degrading situation. For instance, a server application that leaks memory can often safely be restarted in order to ensure more requests can be serviced per day, while avoiding all-out failures.

In other cases the only best action to take is notify the administrator and request human involvement. These cases are often related to security problems and are more structural to the network. Using this system allows administrators to focus on the problems that require their skill and attention, while autonomously solving the problems that don't require their time.



### PROCESS QUERY SYSTEMS

A PQS is a generic correlation engine that puts the focus on the dynamics of an environment, instead of using traditional static methods. By describing how things change over time, a PQS is able to achieve previously unseen levels of detection and correlation in environments too complex for conventional techniques.

### FUNCTIONAL SPECS

This PQS application was implemented using the following system requirements:

- PQS platform (either TRAFEN or C-TRACK)
- Pentium 4 or better
- SPARC III or better
- 512 MB RAM
- Java 1.4 or better

This application will monitor:

- MS Windows XP/2000, Linux 2.6, Solaris > 8

### WOULD YOU LIKE TO KNOW MORE?

If you would like to learn more about this PQS application, or if you would like to use this functionality in your environment, please contact:

Vincent Berk  
vberk@Dartmouth.EDU  
8000 Cummings Hall  
Hanover, NH 03755

