WINDOWS APPLICATION THRASHING TOOL (WATT)

Specification Document

Joseph Caton, My Nguyen, Brooks Olney, and Thomas Pannozzo

November 25, 2018

Version 1.0.3

REVISION HISTORY			
DATE	**VERSION**	**DESCRIPTION**	**AUTHOR**
09/28/2018	1.0.0	Original document	Joseph Caton, My Nguyen, Brooks Olney, Thomas Pannozzo
10/01/2018	1.0.1	Added Traceability Matrix to Description section	Joseph Caton, My Nguyen, Brooks Olney, Thomas Pannozzo
10/11/2018	1.0.2	Updated format	My Nguyen
11/25/2018	1.0.3	Updated user story	My Nguyen

TABLE OF CONTENTS

INTRODUCTION **4 – 5**

PURPOSE 4

INTENDED AUDIENCE AND PERTINENT SECTIONS 4

PROJECT SCOPE 4

DOCUMENT CONVENTIONS 4 – 5

REFERENCES 5

DESCRIPTION **5 – 7**

PRODUCT PERSPECTIVE 5

FEATURES 5 – 6

USER OVERVIEW 6

OPERATING ENVIRONMENT 6

CONSTRAINTS: IMPLEMENTATION / DESIGN 6

TRACEABILITY MATRIX 6

DOCUMENTATION 6 – 7

ASSUMPTIONS / DEPENDENCIES 7

SYSTEM FEATURES **7 – 9**

SYSTEM FEATURE 1 7

SYSTEM FEATURE 2 7

SYSTEM FEATURE 3 7

SYSTEM FEATURE 4 8

SYSTEM FEATURE 5 8

SYSTEM FEATURE 6 8

SYSTEM FEATURE 7 8 – 9

SYSTEM FEATURE 8 9

REQUIREMENTS OF EXTERNAL INTERFACE **9**

USER INTERFACES 9

HARDWARE INTERFACES 9

SOFTWARE INTERFACES 9

ADDITIONAL NONFUNCTIONAL REQUIREMENTS **10**

PERFORMANCE 10

SOFTWARE QUALITY 10

APPENDICES **10 – 12**

APPENDIX A: GLOSSARY OF TERMS 10 – 11

APPENDIX B: ANALYSIS DOCUMENTATION 11 – 12

INTRODUCTION

PURPOSE

The Windows Application Thrashing Tool (WATT) is designed to detect potential thrashing on the Windows operating system (OS) and send alerts. This tool will monitor and poll all the processes on the OS to predict potential thrashing. Once thrashing has occurred, WATT will integrate its machine learning program to predict future thrashing and send out an alert to the IT System Administrator. Thrashing for the purposes of these requirements will be tested in a virtual machine by overloading memory and CPU with processes prior to integrating into the business's (SunView Software) environment.

INTENDED AUDIENCE AND PERTINENT SECTIONS

WATT's intended audience are:

- IT System Administrators
- Developers
- Business Administrators

The whole document is crucial for both the IT System Administrators and developers because they need know what this tool does and how it will help their customers. People from the business side need to know the introduction and description portions of this document to have a better understanding of the tool. The successful application of this tool will directly benefit customers in the service industry.

PROJECT SCOPE

WATT is designed to help the IT department in detecting potential thrashing and sending alerts to the IT System Administrators to take further actions. The benefits of this tool include increased productivity and efficiency, decreased time in trying to find the problem and solving, and increased manpower in working on other projects.

DOCUMENT CONVENTIONS

C# (C Sharp) - a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg.

Central processing unit (CPU) - during thrashing, it spends more time swapping than running the processes.

Operating system (OS) - programs allocate memory from a physical address space to the processes within the programs within a virtual memory space.

Page faults - happen when memory access requested does not map to something in RAM.

PyTorch - an open source machine learning library for Python, based on Torch, used for applications such as natural language processing.

Random-access memory (RAM) - a form of computer data storage that stores data and machine code currently used.
Swapping - occurs when a page from RAM is swapped with a new page to be swapped back to RAM.
Thrashing (disk thrashing) - happens when too many processes compete for a computer's virtual memory resources.

REFERENCES

[1] What is thrashing in an operating system? (2018). *Quora*. [online] Available at: https://www.quora.com/What-is-thrashing-in-an-operating-system [Accessed 12 Sep. 2018].
[2] Thrashing (computer science). (2018). *Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/Thrashing_(computer_science) [Accessed 12 Sep. 2018].
[3] Thrashing. (2018) *Techopedia*. [online] Available at: https://www.techopedia.com/definition/4766/thrashing [Accessed 12 Sep. 2018].
[4] Thrashing. (2017) *Computer Hope*. [online] Available at: https://www.computerhope.com/jargon/t/thrash.htm [Accessed 12 Sep. 2018].
[5] Pytorch.org. (2018). *PyTorch documentation — PyTorch master documentation.* [online] Available at: https://pytorch.org/docs/stable/index.html [Accessed 28 Sep. 2018].
[6] Otte, S. (2018). *Deep Neural Networks with PyTorch - Stefan Otte.* [online] YouTube. Available at: https://www.youtube.com/watch?v=_H3aw6wkCv0 [Accessed 28 Sep. 2018].
[7] "Introduction To Machine Learning | Machine Learning Crash Course | Google Developers". *Google Developers*, 2018, https://developers.google.com/machine-learning/crash-course/ml-intro. Accessed 28 Sept 2018.
[8] Géron, Aurélien. *Hands-On Machine Learning With Scikit-Learn And Tensorflow*. O'reilly, 2018.

DESCRIPTION

PRODUCT PERSPECTIVE

WATT is expected to detect thrashing based on the monitoring and polling results of page faults, low CPU usage, and high memory usage.

FEATURES

WATT has the following features listed below:

- capture the high page file activity (when pages are swapped in and out per second) to determine thrashing;
- capture the system's high memory usage to detect thrashing;
- capture the low CPU usage (due to every process waiting for the I/O to finish) to detect thrashing;
- measure the rate of change for page faults as it increases to detect precursors of thrashing;

- package these statistics in an organized manner to be formatted into CSV and integrated it with an API;
- automatically detect thrashing based on the page file activity, high memory usage, and CPU usage;
- implement machine learning so it will automatically determine the conditions of thrashing;
- determine when one of the systems is thrashing and then send an alert;
- and limit the possibility of false positives.

USER OVERVIEW

WATT will be most useful to IT System Administrators in helping them detect potential thrashing on their machines and their customer's machines. In turn, WATT will help IT System Administrators in helping their customers to be productive while being productive themselves.

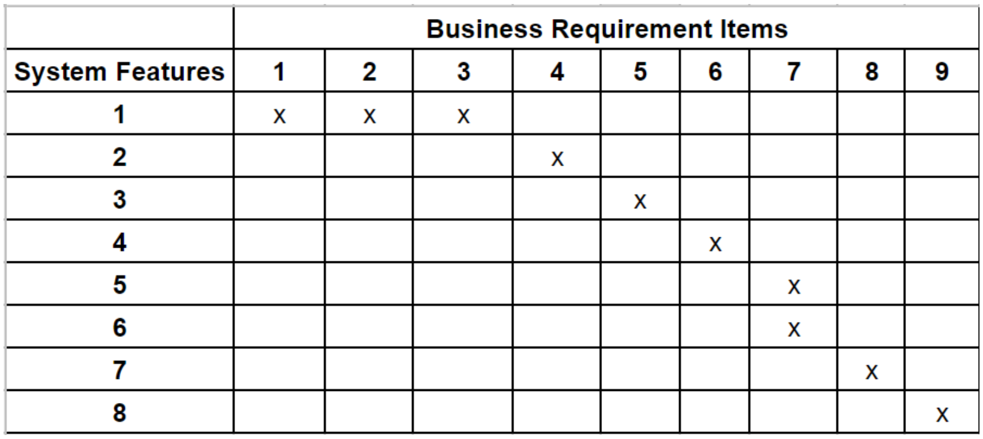
OPERATING ENVIRONMENT

WATT's algorithm will be written in C# and Python with the implementation of machine learning for thrashing. WATT will have the potential to be integrated into the business's (SunView Software) environment.

CONSTRAINTS: IMPLEMENTATION / DESIGN

The biggest constraint is time for research and development due to the 16-weeks long timeframe for this project. Another potential constraint would be the team's busy schedule and the ability to meet up more than once weekly to develop and test.

TRACEABILITY MATRIX



	Business Requirement Items								
System Features	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	x	x	x						
2				x					
3					x				
4						x			
5							x		
6							x		
7								x	
8									x

DOCUMENTATION

The following documents are used to track the content of this project and the deliverables:

- Business Requirements Document Version 2.05
- WATT Design Review

- Functionalities of WATT i.e. tasks/roles, user stories, sprints

ASSUMPTIONS / DEPENDENCIES

The machine learning portion of WATT not producing the correct output. WATT falsely identifies thrashing and sends an alert too early.

SYSTEM FEATURES

SYSTEM FEATURE 1

DESCRIPTION AND PRIORITY	Track system resource usage i.e. page faults, physical memory usage, and CPU usage.
STIMULUS / RESPONSE SEQUENCES	Monitors are polled in intervals, returning values for each of the system resources.
FUNCTIONAL REQUIREMENTS	Timeline is formed in conjunction with the polling function, and organized to pass into the machine learning platform.

SYSTEM FEATURE 2

DESCRIPTION AND PRIORITY	Measure the rate of change for the page faults as it increases.
STIMULUS / RESPONSE SEQUENCES	Page fault rate changes, logs taken as necessary.
FUNCTIONAL REQUIREMENTS	Poll interval is appropriate for page changes.

SYSTEM FEATURE 3

DESCRIPTION AND PRIORITY	Package statistics in a CSV format and integrated it with an API.
STIMULUS / RESPONSE SEQUENCES	Take data from monitor polling.
FUNCTIONAL REQUIREMENTS	Integrates with the business's (SunView Software) Web platform.

SYSTEM FEATURE 4

DESCRIPTION AND PRIORITY	Force thrashing on virtual machine.
STIMULUS / RESPONSE SEQUENCES	Create different scenarios of thrashing system resources.
FUNCTIONAL REQUIREMENTS	WATT must detect this occurrence.

SYSTEM FEATURE 5

DESCRIPTION AND PRIORITY	Apply gathered data to machine learning model.
STIMULUS / RESPONSE SEQUENCES	Based on the results of the polling and monitoring, the model will send an alert of processes that are potentially thrashing.
FUNCTIONAL REQUIREMENTS	Machine learning algorithm should produce accurate model to indicate potential thrashing.

SYSTEM FEATURE 6

DESCRIPTION AND PRIORITY	Interaction between Python machine learning model and C# polling and monitoring function.
STIMULUS / RESPONSE SEQUENCES	Based on the results of the polling and monitoring from C#, there must be an interaction from the results to the machine learning model.
FUNCTIONAL REQUIREMENTS	Machine learning algorithm should produce accurate model to indicate potential thrashing.

SYSTEM FEATURE 7

DESCRIPTION AND PRIORITY	Send a warning or alert when signs of thrashing are occurring on the system.

STIMULUS / RESPONSE SEQUENCES	System behavior mimics thrashing conditions, utility acts accordingly.
FUNCTIONAL REQUIREMENTS	Machine learning algorithm detects thrashing and sends alert.

SYSTEM FEATURE 8

DESCRIPTION AND PRIORITY	Limit the possibility of false positives.
STIMULUS / RESPONSE SEQUENCES	System behavior mimics thrashing conditions for small period of time, goes back to normal, and no action taken by the utility.
FUNCTIONAL REQUIREMENTS	The utility does not alert the IT System Administrator when thrashing is not actually occurring.

REQUIREMENTS OF EXTERNAL INTERFACE

USER INTERFACES

Users will utilize the command line to run the program. If the program detects the OS thrashing it will alert the user through a dialog window. Upon successful functionality of the program, integration will be considered with the business's (SunView Software) Web platform. In this scenario, the program will alert the user through the API.

HARDWARE INTERFACES

WATT is a software tool that will integrate into the business's (SunView Software) machines either on the client or server.

SOFTWARE INTERFACES

WATT's algorithm is written in C# using a .NET library with PyTorch. WATT will integrate into the Windows operating system to utilize the resources i.e. page faults, low CPU usage, high memory usage to monitor potential thrashing. Based on monitoring and polling, WATT will send an alert if potential thrashing will occur.

ADDITIONAL NONFUNCTIONAL REQUIREMENTS

PERFORMANCE

In addition to detecting thrashing and sending alerts to the IT System Administrators, an enhanced feature that WATT would do is to detect which process is causing the thrashing.

SOFTWARE QUALITY

The application will be considered successful based on the following:

- the utilization of a system call returns the number of the page faults for each process identifier (PID);
- the utilization of a system call returns the highest memory usage for each process identifier (PID);
- the utilization of a system call returns the lowest CPU usage for each process identifier (PID);
- a flag is raised for page faults;
- the product successfully integrates with the business's (SunView Software) UI platform;
- the detector model gives a warning that a potential thrashing might occur;
- the product gives the conditions of thrashing;
- the product identifies thrashing based on polling;
- and the alert has not been given until the minimum period has elapsed.

APPENDICES

APPENDIX A: GLOSSARY OF TERMS

C# (C Sharp) - a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg.

Central processing unit (CPU) - during thrashing, it spends more time swapping than running the processes.

Machine learning - a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data without being explicitly programmed.

Natural language processing - an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages.

Operating system (OS) - programs allocate memory from a physical address space to the processes within the programs within a virtual memory space.

Page faults - happen when memory access requested does not map to something in RAM.

PyTorch - an open source machine learning library for Python, based on Torch, used for applications such as natural language processing.

Random-access memory (RAM) - a form of computer data storage that stores data and machine code currently used.

Swapping - occurs when a page from RAM is swapped with a new page to be swapped back to RAM.

Thrashing (disk thrashing) - happens when too many processes compete for a computer's virtual memory resources.

APPENDIX B: ANALYSIS DOCUMENTATION

The following files are pertinent to this project:

- Business Requirements Document Version 2.05
- WATT Design Review
- Functionalities of WATT i.e. tasks/roles, user stories, sprints
- Figures below

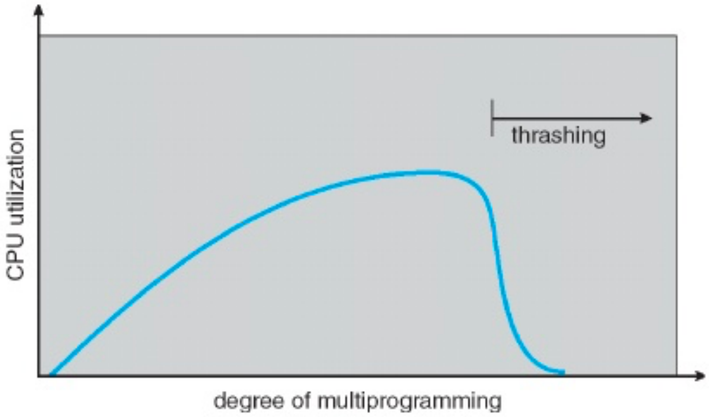
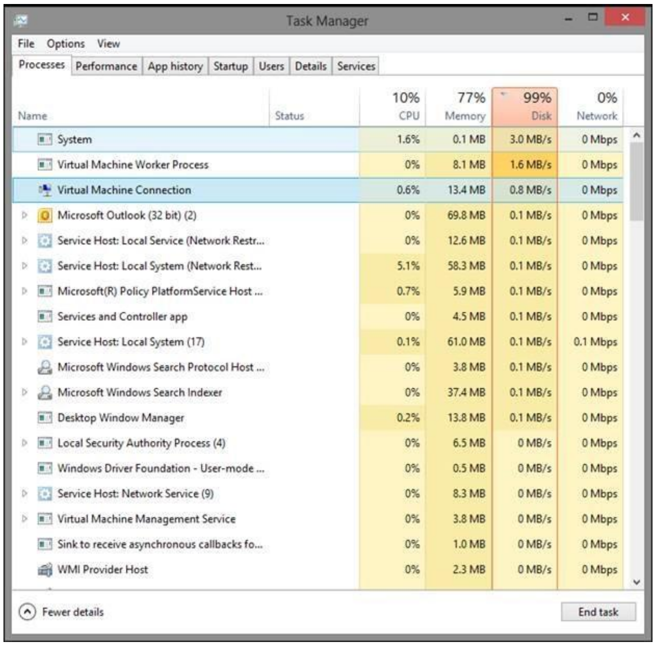


Figure 1: Causes of thrashing



		10%	77%	99%	0%
Name	Status	CPU	Memory	Disk	Network
System		1.6%	0.1 MB	3.0 MB/s	0 Mbps
Virtual Machine Worker Process		0%	8.1 MB	1.6 MB/s	0 Mbps
Virtual Machine Connection		0.6%	13.4 MB	0.8 MB/s	0 Mbps
Microsoft Outlook (32 bit) (2)		0%	69.8 MB	0.1 MB/s	0 Mbps
Service Host: Local Service (Network Restr...		0%	12.6 MB	0.1 MB/s	0 Mbps
Service Host: Local System (Network Rest...		5.1%	58.3 MB	0.1 MB/s	0 Mbps
Microsoft(R) Policy PlatformService Host ...		0.7%	5.9 MB	0.1 MB/s	0 Mbps
Services and Controller app		0%	4.5 MB	0.1 MB/s	0 Mbps
Service Host: Local System (17)		0.1%	61.0 MB	0.1 MB/s	0.1 Mbps
Microsoft Windows Search Protocol Host ...		0%	3.8 MB	0.1 MB/s	0 Mbps
Microsoft Windows Search Indexer		0%	37.4 MB	0.1 MB/s	0 Mbps
Desktop Window Manager		0.2%	13.8 MB	0.1 MB/s	0 Mbps
Local Security Authority Process (4)		0%	6.5 MB	0 MB/s	0 Mbps
Windows Driver Foundation - User-mode ...		0%	0.5 MB	0 MB/s	0 Mbps
Service Host: Network Service (9)		0%	8.3 MB	0 MB/s	0 Mbps
Virtual Machine Management Service		0%	3.8 MB	0 MB/s	0 Mbps
Sink to receive asynchronous callbacks fo...		0%	1.0 MB	0 MB/s	0 Mbps
WMI Provider Host		0%	2.3 MB	0 MB/s	0 Mbps

Figure 2: Characteristics of thrashing e.g. high page faults, high memory usage, and low CPU

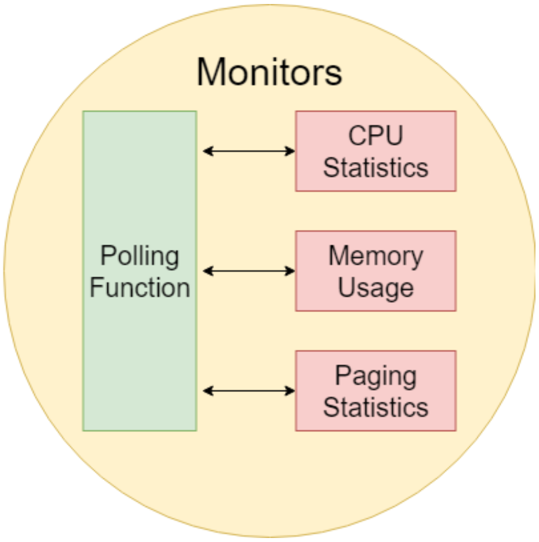


Figure 3: Monitors for data gathering and polling