© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 06 | July – September 2016
Paper is available at www.jrps.in | Email : info@jrps.in

REVIEW PAPER ON PERFORMANCE OF TEST SUITES IN MATLAB

[1]Jyoti , Research Scholar, Department of CE, IIET Kinana, Jind, jyotilather07@gmail.com
[2]Amit Garg, Assistant Professor, Department of CE, IIET Kinana, Jind, amit.indus86@gmail.com

ABSTRACT: Software testing involves execution of a software component/system component at evaluate one or many properties of interest. **Software testing** is an inquiry of conduct to provide stakeholders within information about quality of product or service under test.[1] Software testing could also provide an objective, independent view of software to allow business to appreciate & understand risks of software implementation. Testing could determine correctness of software under assumption of specific hypotheses testing could not identify all defects within software. Instead, it furnishes a critique or difference that compares state & behavior of product against oracles principles or mechanisms by which some problem.

[1] Introduction

Software testing is an inquiry of conduct to provide stakeholders within information about quality of product or service under test.[1] Software testing could also provide an objective, independent view of software to allow business to appreciate & understand risks of software implementation. Test techniques include process of executing a program or application within intent of find software bugs.

Software testing involves execution of a software component/system component at evaluate one or many properties of interest. In general, these properties indicate extent to which component or system under test:

1. meets requirements that guided its design & development,
2. responds correctly to all kinds of inputs,
3. performs its functions within an acceptable time,
4. is sufficiently usable,
5. can be installed & run in its intended environments, and
6. Achieves general result its stakeholders desire.

Defects & failures

One common source of expensive defects are requirement gaps, unrecognized requirements which result in errors of omission by program designer.[6] Requirement gaps could often be non functional requirements such as testability, maintainability, usability, performance Software faults occur through following processes. If this defect is executed, in certain situations system would produce wrong results, causing a failure.[7] Not all defects would important result in failures. For example, defects in dead code would never result in failures. A defect could turn into a failure when environment is changed.

White-box testing

White-box testing tests internal structures or workings of a program, as opposed to functionality exposed to end-user. In white-box testing an internal perspective of system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through code & determine appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

[2] Literature Review

Zhao Changwei1, Huang Tao1 and Dong Yongsheng in 2, April 2013 "Research On Distributed Software Testing"

In order to solve low efficiency problem of large-scale distributed software testing, CBDSTP Cloud-Based Distributed Software Testing Platform is put forward. This platform could provide continuous integration & automation of testing for big software systems, which could make full use of resources on cloud clients, achieving testing result s in real environment & reasonable allocating testing jobs, to resolve Web application software configuration test, compatibility test & distributed test problems, to reduce costs. Through making MySQL testing on this prototype system, verification is made for platform architecture & job allocation effectiveness. Based on idea of making full use of cloud resources which supports continuous

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 06 | July – September 2016
Paper is available at www.jrps.in | Email : info@jrps.in

online testing for large-scale software systems & it could achieve test task division & scheduling, collection & analysis & other functions which provides a well solution for distributed software test & many experiments verification feasibility & reliability of prototype design & architecture.

Alessandro Orso Gregg Rothermel written by Software Testing: A Research Travelogue (2000–2014)

Despite decades of work by researchers & practitioners on numerous software quality assurance techniques, testing remains one of most widely practiced & studied approaches for assessing & improving software quality. Our goal, in this paper, is to provide an accounting of some of most successful research performed in software testing since year 2000, & to present what appear to be some of most significant challenges & opportunities in this area. To be more inclusive in this effort, & to go beyond our own personal opinions & biases, we began by contacting over 50 of our colleagues who are active in testing research area, & asked them what they believed were (1) most significant contributions to software testing since 2000 & (2) greatest open challenges & opportunities for future research in this area.

QI Wenlu Shi Xiaolong Chen Canglong in November 2014 written by Research on "Automatic Test of Rulestream Function Based on IBM RFT"

How to improve efficiency of software testing to ensure software quality & thus shorten period of software development, had been become a very urgent task needing to handle. On basis of study of automated testing technology, & combined with existed software system, this paper employs Rational Functional Tester (RFT) as automatic software test platform. Application results show automatic software test technique proposed in this paper greatly reduce test cost, increase test efficiency, & shorten software development period.

Arpad Beszedes Laszlo Vidacs "Academic & Industrial Software Testing Conferences: Survey & Synergies"

Just as with any other profession, an efficient way to exchange ideas & networking in software testing are conferences, workshops & similar events. This is true for both professional testers & researchers working in testing area. However, these two groups usually look for different events: a tester likes to attend "industrial conferences, whereas a researcher is more likely interested in "academic" (scientific or research) conferences. Although there is notable exceptions, in that case separation is substantial, which hinders a successful academy industry collaboration, & communication about demand & supply of research in software testing. This paper reviews 101 conferences: two thirds are academic ones, rest being industrial. Besides providing this reasonably comprehensive list, we analyze any visible synergies such as events that have a mixed Program Committee & offer a program with elements from both sides. We found only a handful of such events, but these could serve both as opportunities for attendees who wish to extend their perspectives & as models for organizers of future conferences

[3] Tools & Technology

MATLAB

MATLAB is programming language by Math Works. It started out as a matrix programming language where linear algebra programming was simple. It could be run both under interactive sessions & as a batch job.

This tutorial gives you aggressively of MATLAB programming language. Problem based MATLAB had been give in simple & easy path to make your learning fast & effective.

MATLAB (matrix laboratory) is a fourth generation programming language & interactive environment for visualization & programming.

It plotting of functions & data; creation of user interfaces; interfacing with programs written in other languages, and analyze data; develop algorithms; & create models & applications.

It have numerous built-in commands & math functions that help you in mathematical calculations, generating plots, & performing numerical methods.

Uses of MATLAB

MATLAB has large are used as a computational tool in science & engineering encompassing fields of physics,

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 06 | July – September 2016
Paper is available at www.jrps.in | Email : info@jrps.in

chemistry, math & all engineering streams.Signal Processing & Communications

- Image & Video Processing
- Systems Control
- Measurement & Test
- Computational Finance
- Computational Biology

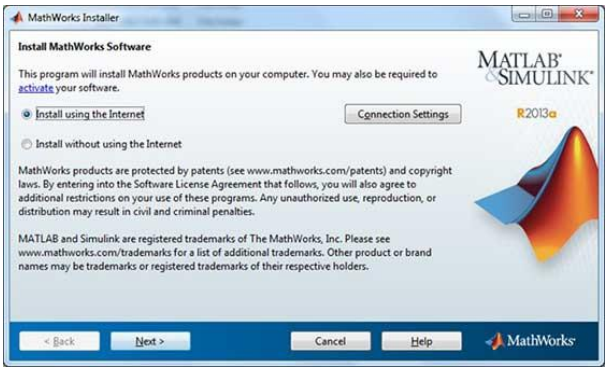

Fig 1 MTLAP

Understanding MATLAB Environment

MATLAB development IDE could be launched from icon created on desktop. The main working window in MATLAB is called desktop. When MATLAB is started, desktop appears in its default layout

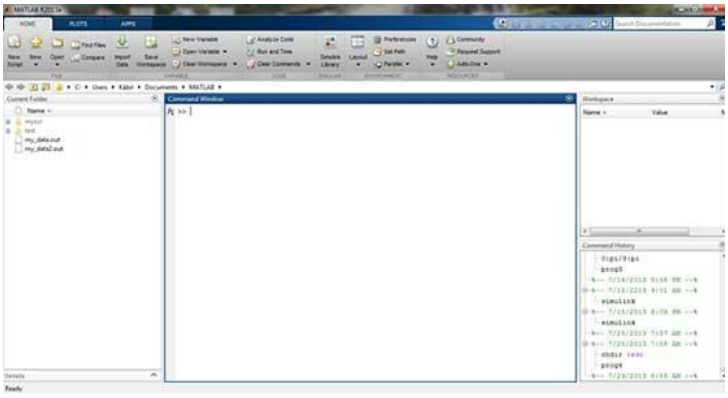

Fig 2 The desktop

[4] PROPOSED WORK

The proposed work is to analyse performance of Test Suites in different cases such as if we increase number of Core Processors and if number of Ram is increased and if Complexity of code is reduced

Impact of Core Processors on Performance

A **multi processor** is a single computing with two or many independent actual processing units (called "cores"), which are units that read & execute program instructions.[1] The instructions are ordinary CPU instructions but multiple cores could run multiple instructions at same time, increasing overall speed for programs to parallel computing.[2] Manufacturers typically integrate cores onto a single integrated circuit or onto multiple dies in a single chip package.

Advantages

The proximity of multiple CPU cores on same die allows cache coherency circuitry to operate at high clock rate than what is possible if signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves performance of cache snoop operations. Put simply, this means that signals CPUs travel shorter distances, & therefore those signals degrade less. These higher-quality signals allow many data to be sent in a given time, since every person signals could be shorter & do not need to be repeated as often.

Impact of Ram on Performance

RAM contains multiplexing & demultiplexing circuitry, to connect data lines to addressed storage for reading or writing entry. Usually many than one bit of storage is accessed by same address, & RAM devices often have multiple data lines & are said to be '8-bit' or '16-bit' etc. devices.

In today's technology, random-access memory takes form of integrated circuits. RAM is normally associated with volatile types of memory (such as DRAM memory modules), where stored information is lost if power is removed, although many efforts have been made to develop non-volatile RAM chips.[1] Other types of non-volatile memories exist that allow random access for read operations, but either do not allow write operations or have other kinds of limitations on them. These include most types of ROM & a type of flash memory called *NOR-Flash*.

[6] Need of Research

Testing of **all work flows, all fields, all negative scenarios** is time & cost consuming.It is **difficult to test for multi lingual sites manually.** Automation does **not** require **Human intervention**. You could run automated test unattended (overnight).Automation **increases speed of test execution.** Automation helps **increase Test Coverage. In future performance of test suites could be analysed on bases of Cache & Remote Network performance.**

[7] Conclusion

Testing of all work flows, all fields, all negative scenarios **is time & cost consuming.It is** difficult to test for multi lingual sites manually. **Automation does** not **require** Human intervention**. You could run automated test unattended (overnight).Automation** increases speed of test execution. **Automation helps** increase Test Coverage. In future performance of test suites could be analysed on bases of Cache & Remote Network performance.

Automation Testing saves time, cost & manpower. Once recorded, it's easier to run an preset test suite when compared to manual testing which would require skilled labour. If number of CPU & Ram Increases then performance of testing suites also increases.

The proximity of multiple CPU cores on same die allows cache coherency electrical to operate at a much higher clock rate than what is possible if signals had been to travel off-chip. Combining correspondent CPUs on a single die significantly improves performance of cache snoop operations. Put simply, this means that signals different CPUs travel shorter distances, & therefore those signals degrade less. These higher-quality signals allow many data to be sent in a given time period, since individual signals could be shorter & do not need to be repeated as often.

REFERENCES

1.Object Oriented software testing by Devid C. Kung
http://www.ecs.csun.edu/~rlingard/COMP595VAV/OOSWTesting.pdf

2. Automated Testing tools
http://www.guru99.com/automation-testing.html

3. Matlab Documentation
http://in.mathworks.com/help/matlab/matlab_oop/getting-familiar-with-classes.html

4.ML-Unit Matlab unit Test Framework
http://sourceforge.net/p/mlunit/mlunit/HEAD/tree/trunk/

5. Object Oriented programming in Matlab
http://www.ce.berkeley.edu/~sanjay/e7/oop.pdf

6. Artem, M., Abrahamsson, P., & Ihme, T. (2009). Long-Term Effects of Test-Driven
Development A case study. In: *Agile Processes in Software Engineering & Extreme*
Programming,10th International Conference, XP 2009,. 31, pp. 13-22. Pula, Sardinia, Italy:
Springer.

7. Bach, J. (2000, November). Session based test management. *Software testing & quality*
engineering magzine(11/2000),(
http://www.satisfice.com/articles/sbtm.pdf).

8. Bach, J. (2003). Exploratory Testing Explained, The Test Practitioner 2002,
(http://www.satisfice.com/articles/et-article.pdf).

9. Bach, J. (2006). *How to manage & measure exploratory testing.* Quardev Inc.,
(http://www.quardev.com/content/whitepapers/how_measure_exploratory_testing.pdf).

10. Basilli, V., & Selby, R. (1987). Comparing effectiveness of software testing strategies.
IEEE Trans. Software Eng., 13(12), 1278-1296.

11. Berg, B. L. (2009). *Qualitative Research Methods for Social Sciences (7th International*

Edition) (7th ed.). Boston: Pearson Education.

12. Bernat, G., Gaundel, M. C., & Merre, B. (2007). Software testing based on formal
specifications: a theory & tool. In:*Testing Techniques in Software Engineering, Second*
Pernambuco Summer School on Software Engineering. 6153, pp. 215-242. Recife:
Springer.

13. Bertolino, A. (2007). Software Testing Research: Achievements Challenges Dreams. In:
International Conference on Software Engineering, ISCE 2007, (pp. 85-103). Minneapolis:
IEEE.

14. Causevic, A., Sundmark, D., & Punnekkat, S. (2010). An Industrial Survey on
Contemporary Aspects of Software Testing. In: *Third International Conference on*
Software Testing, Verification & Validation (pp. 393-401). Paris: IEEE Computer
Society.

15. Chillarege, R. (1999). *Software Testing Best Practices.* Tehcnical Report RC2145, IBM.

16. Frankl, P. G., & Hamlet, R. G. (1998). Evaluating Testing Methods by Delivered
Reliability. *IEEE Trans. Software Eng., 24*(8), pp. 586-601.

17. Galin, D. (2004). *Software Quality Assurance: From theory to implementation.* Pearson
Education Ltd.

18. Butts, K., et. al., "Automotive Powertrain Control Development Using CACSD", Perspectives in Control: New Concepts & Applications, Tariq Samad (ed.), IEEE Press, 1999.

19. Butts, K., Toeppe, S., Ranville, S., "Specification & Testing of Automotive Powertrain Control System Software using CACSD tools", 1998, Proceedings of 17th AIAA/IEEE/SAE Digital Avionics System Conference

20. Toeppe, S., Ranville, S., "Model Driven Automatic Unit Testing Technology: Tool Architecture Introduction & Overview", 1999, Proceedings of 18th AIAA/IEEE/SAE Digital Avionics System Conference