Advances in Parallel Computing Algorithms, Tools and Paradigms
D.J. Hemanth et al. (Eds.)

doi:10.3233/APC220052

The Functional and Storage Risks Associated to the Size Estimation of Parallel Computing Applications

John T Mesia Dhas[a,1] and Midhunchakravarthy[b]
[a] *Post-Doctoral Research Fellow (PDF), Department of CSE,*
Lincoln University College, Malaysia
[b] *Associate Professor, Faculty of Computer Science and Multimedia,*
Lincoln University College, Malaysia

Abstract. The software size estimation is the basic and essential process in the Software Development Life Cycle (SDLC) for budgeting, developing and delivering in the scheduled manner. The modern computer software like Artificial Intelligence (AI), Machine Learning (ML), Internet of Things (IoT) and Data Analytics are highly dynamic, distributed and parallel processing. To identify the size of this versatile environment is highly difficult and risk based. A small fault may cause negative consequences and also suspension of the project. This article gives a study on the functional, social, economic, political, organizational and nonfunctional risks associated to the parallel computing applications.

Keywords. Parallel Computing Applications, Software Size, Artificial Intelligence, Machine Learning, Internet of Things, Software Risks.

1. Introduction

The modern software system is highly dynamic, self-learned and motivated from environment, ability to do automatic decision making and fault tolerance, capable to process a huge amount of data for making single decision, manipulating and testing complex calculations and algorithms, to work with multiple output for single problem, to make fuzzy based decisions, highly influenced with socio, economic and political behavior of the society, highly distributed and parallel processed, based on complex and variable architecture, vast storage and cluster based, and so on.

To develop or measure the size of this versatile software environment is highly difficult compare with the traditional software environment. An existing single method or the current metrices are not sufficient for identifying the actual size of the modern software system.

[1] John T Mesia Dhas, Post-Doctoral Research Fellow (PDF), Department of CSE, Lincoln University College, Malaysia.

To quantify the size of the software at the beginning of SDLC is highly essential for identifying cost, time and effort for the software development.The size of the software is playing crucial role for deriving cost, time and effort of the software development process.

2. Software Risk Management

Software risk management is the process of identifying and cleaning adverse effects present in the software development process. The rate of change of adverse effect of the risk may be from negligible to kill the entire process. To identify the risks and analyze the best treatment methods for that risk. The measure of the probability of severity of adverse effect is risk. Financial risks, Health risks, Environmental and ecological risks, and Security risks are the different types of risks existing in the software industry.

Risk Process

The risk processes are, Identify the risk., Analyze its implications., Determine treatment methods, and Monitor performance of treatment methods.

Software Risk Management

The process of identifying, addressing and eliminating risks in a software development process is software risk management. It is necessary that some form of measurement is undertaken to determine and classify the range of risks a software development project faces and to identify areas where significant exposure exists.

Software Risk Management Steps and Techniques

The steps of Software risk management are shown in Figure 1.

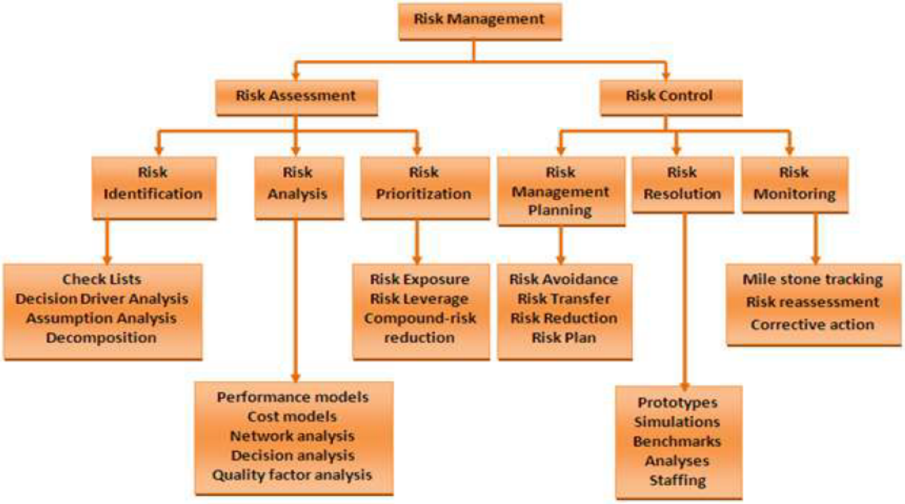

Figure 1. SRM Steps

3. Risks Associated to Parallel Computing System's Size Estimation

The risks of the modern software system are classified into two broad categories. They are, i. Functional Risks ii. Social Risks

3.1 Functional Risks

The functional risks of size estimation are based on the ambiguity present in the identification of functional units.

The Versatile Behaviour of Modern Software

The modern software is versatile using distinct programming languages, operating systems, file formats, topologies, SDLC and application tools. The functional unit of one software may differ from other software. It may lead to confusions in identifying functional units.

The Variable Behaviour of Functional Units

In the same software, the same functional unit will behave differently in different modules. It also increases the difficulties for identifying functional units of modern software systems.

Difficult to Rank Function Points

Ranking of function points is differing from organisation to organisation. So, it increases the confusions in estimating the size of modern software systems.

Insufficient CAF

The existing IFPUG Function Point size estimation technique uses only fourteen CAF. It is not competent for sizing modern software systems.

Dynamic Function Points

The dynamic behaviour of function points generates difficulties in the size estimation of modern software systems.

3.2 Social Risks

The social risks of size estimation and project management mainly depends on industrial policies, socio-economic policies, political system, and universal standards of the client, developer and domain.

Economic and Financial Risks

The economic and financial status of the institutions also affects the development of software in its estimated period of time as the fund flow is essential for managing the needs of software development process.

Social and Environmental Risks

The social issues in the society like employees, institutional policies, working hours and ecological policies also affects the development of software. It creates major impact on size estimation.

Security Risks

The secured transfer and storing of information is highly essential till the life time of software system. So, efficient algorithms must be developed for those issues.

National Policies

The national policies of the client and developer may also affect the development process of the software system.

Universal Standards

The universal standards of software development, employees and organizational standards also affect the growth of software development process.

4. Findings in FPA

The traditional function point estimation techniques were using only five functional units and fourteen CAF (discussed in 3.4.3). These are not sufficient for estimating the size of modern software system. The following are the new functional units and CAF for attaining the accurate sizing of modern software to an extent.

Internal Input

It is an important essential functional unit for modern software. For example,

```
void main()
{
const float pi =
3.14;
int r = 10;
float k;
k=pi * r * r;
printf ("Area =
%f", k);
}
```

In the above function, the internal direct assignments (eg. variable r) and constants (eg. pi) are the examples for internal input. In the traditional FPA, the internal input is not considered as functional units. Therefore, it will reduce the size of the software product in FPA.

Internal Operations

The internal operations are not considered in the traditional FPA estimation. For example,

```
void main()
{
int i,j,k,l;
printf("Enter
the value of a
and b");
scanf("%d%d"
, &i,&j);
k=i+j;
l=k-i;
printf("Value
of l=%d",l);
}
```

In the above example, variables ‘i’ and ‘j‘ are EI, variable ‘l*’* is EO, but the internal operation ‘k=i+j’ is not considered as functional unit. The internal operations are playing very important role in scientific and AI software programs. To increase the accuracy of modern software size estimation, internal operations also can be considered as a functional unit for FPA size estimation process.

Indexed Data

The arrays and lists are very essential data variables for modern software. But all the indexed values are not getting importance. The indexed data variables also considered as the single valued variables. For example, in the existing FPA estimation ‘int a[10]’ will be getting equal weightage as that of ‘int a’. It reduces the effort level of developer at the time of estimation.

Multiple Forms of Output

Nowadays, the modern software are capable to create many forms of outputs like data report, crystal report, excel formats, GUI formats, database reports, etc. But the existing FPA methods are considering only one format of output. The exclusion of different forms of output affects the size of the software system. Therefore, the time and cost constraints are not accurate in the estimation.

Insufficient Metric Values

The size of the functions and the data handling with the functions are increasing by time. The existing metric values (low, average and high) are not sufficient. Hence, we have to add one more very high metric value.

Database and Text Files

The database and text files were not considered in the existing function point methods. The current technologies like machine learning, data mining, data analytics and big data are using a large amount of historical and primary data. The neglecting of database and text affects the actual effort level of the software system.

Multi-valued Function Points

A variable will act as one functional unit in one function and the same variable will act as another functional unit in another function is known as Multi-valued Function Points (MVFP). For example,

```
void get(void);     {              {             void add(int a,
void  add  (int,    get();         a=10;         int b)
int);               add(a,b);      b=20;         {
int a,b;            }              }             int c= a+b;
void main()         void get()                   }
```

In the above example, the variables ‘*a*’ and ‘*b*’ are as internal inputs in function get() and as EI in function add(). Similarly, ILF of one function becomes EIF of another function and EQ of one function is EI of another function. The importance of MVFP is not considered in FPA method.

Dependent Function Points (DFP)

Some functional units are identified based on some other functional units. The choice based functional segments are example for DFP. For example,

```
void main()             printf ("Enter a        {                       {
{                       and b values");         big = a;                big = b;
int  a,  b,  big,       scanf("%d               small = b;              small = a;
small;                  %d", &a, &b);           }                       }
                        if (a > b)              else                    }
```

In the above example, 'if' block and 'else' block will be chosen based on the variables '*a*' and '*b*'. If one block is chosen then all other blocks are omitted. The small applications won't give any impact on its size estimation. But in the large-scale systems, the choices are playing great role in size of the software. The choices and dependent function points were not considered in FPA method. Similarly, case () structure also is an example for DFP.

Composite Function Points (CFP)

A variable will get the characteristics of different functional units in the same function is known as CFP. For example,

```
void main()             printf("Enter  a        scanf("%d%d"            b=a-b;
{                       and b values");         ,&a,&b);                a=a-b;
int a,b;                                        a=a+b;                  }
```

In the above example, variable '*a*' and '*b*' are accepting the characteristics of External Inputs and Intermediate Results. Similarly, External Inquiries becomes Internal or External Inputs within the same function. The composite behaviors of functional units are not discussed in FPA method.

5. Conclusion

Parallel computing application system is an amalgamation of a huge structured, semi structured, unstructured and multi structured database with multi-dimensional secured analytical application software. That is why; Parallel computingsystem is a multipurpose one, so any single method or the existing metrics and measures won't give the actual software size. But the function point-based estimation technique is independent of the language, tools, or methodologies used for implementation; i.e., they do not obtain into consideration programming languages, database management systems, processing hardware, any otherdatabase technology or any platform. So, some updating with the existing metrics and measures of the function point will give

References

[1] Sfar AR, Zied C, Challal Y. A Systematic and Cognitive Vision for IoT Security: A Case Study of Military Live Simulation and Security Challenges. In: Proc. 2017 international conference on smart, monitored and controlled cities (SM2C), Sfax, Tunisia, 17–19 Feb. 2017. https ://doi.org/10.1109/sm2c.2017.80718 28.

[2] Gatsis K, Pappas GJ. Wireless control for the IoT: power spectrum and security challenges. In: Proc. 2017 IEEE/ACM second international conference on internet-of-things design and implementation (IoTDI), Pittsburg, PA, USA, 18–21 April 2017. INSPEC Accession Number: 16964293.

[3] Zhou J, Cap Z, Dong X, Vasilakos AV. Security and privacy for cloud-based IoT: challenges. IEEE Commun Mag. 2017;55(1):26–33. https ://doi.org/10.1109/MCOM.2017.16003 63CM.

[4] Sfar AR, Natalizio E, Challal Y, Chtourou Z. A roadmap for security challenges in the internet of things. Digit CommunNetw. 2018;4(1):118–37.

[5] Minoli D, Sohraby K, Kouns J. IoT security (IoTSec) considerations, requirements, and architectures. In: Proc. 14th IEEE annual consumer communications & networking conference (CCNC), Las Vegas, NV, USA, 8–11 January 2017. https ://doi.org/10.1109/ccnc.2017.79832 71.

[6] Gaona-Garcia P, Montenegro-Marin CE, Prieto JD, Nieto YV. Analysis of security mechanisms based on clusters IoT environments. Int J Interact MultimedArtifIntell. 2017;4(3):55–60.

[7] Behrendt F. Cycling the smart and sustainable city: analyzing EC policy documents on internet of things, mobility and transport, and smart cities. Sustainability. 2019;11(3):763.

[8] IoT application areas. https ://iot-analy tics.com/top-10-iot-proje ct-applicatio n-areas -q3-2016/. Accessed 05 November 2021.

[9] Zanella A, Bui N, Castellani A, Vangelista L, Zorgi M. Internet of things for smart cities. IEEE IoT-J. 2014;1(1):22–32.

[10] Khajenasiri I, Estebsari A, Verhelst M, Gielen G. A review on internet of things forintelligent energy control in buildings for smart city applications. Energy Procedia. 2017; 111:770–9.

[11] Internet of Things. http://www.ti.com/technologies/internet-of-things/overview.html Accessed 02-09-2021.

[12] John T Mesia Dhas, C.R. Bharathi," Relative Analysis of Sizing Methods in the senseof E-Commerce system", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015), pp. 39808-39816.

[13] John T Mesia Dhas, C.R. Bharathi, "Risks Associated to Size Estimation of E-Commerce System using Function Point based Estimation Techniques", Indian Journal of Science and Technology, Vol 9(7), DOI: 10.17485/ijst/2016/v9i7/85148, February 2016.

[14] John T Mesia Dhas, C.R. Bharathi, "E-Commerce System Size using User BasedFunction Points", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017), pp. 6115-6122

[15] Heer T, Garcia-Morchon O, Hummen R, Keoh SL, Kumar SS, Wehrle K. Security challenges in the IP based internet of things. Wirel Pers Commun. 2011;61(3):527–42.

[16] T. S. Shiny Angel, Paul Rodrigues, "Size and Flexibility Metrics for E-Learning System", International Journal of Software Engineering and Technology, Vol. 4, No 7 , July 2012.

[17] T. S. Shiny Angel, Paul Rodrigues, "Limitations of Function Point Analysis in E-Learning System Estimation", International Journal of Computational Engineering Research, pp. 156 -161, July 2012.

[18] T.S. Shiny Angel, Paul Rodrigues, "Estimating the Size of E-Learning System using Learning Object Points Method", Indian Journal of Science and Technology, September 2016.

[19] T. S. Shiny Angel, Paul Rodrigues, "Comparative Analysis of Sizing Techniques in the sense of E-Learning system", International Journal of Applied Engineering Research, Volume 10, Number 16,pp 36303-36312, August 2015.

[20] T. S. Shiny Angel, Paul Rodrigues, "ELSE:E-Learning System Estimator", International Review on Computers and Software, Vol. 7 n. 6, pp. 3033-3036, Nov- 2012.