# Design Issues in Distributed Software

December 2018

DOI:10.1109/PDGC.2018.8745919

Conference: 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)

**Authors:**

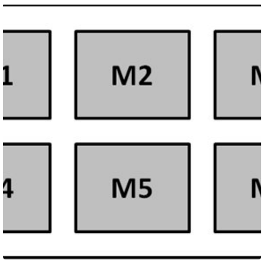
**Ashish Kumar Maurya**
Motilal Nehru National Institute of Technology Allahabad India

**Dipty Tripathi**
Indian Institute of Technology (Banaras Hindu University) Varanasi

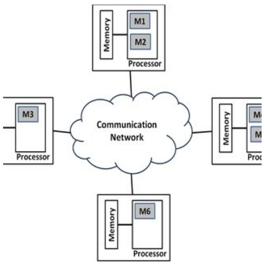
**Amit Biswas**
Indian Institute of Technology (Banaras Hindu University) Varanasi

**Anil Kumar Tripathi**
Indian Institute of Technology (Banaras Hindu University) Varanasi

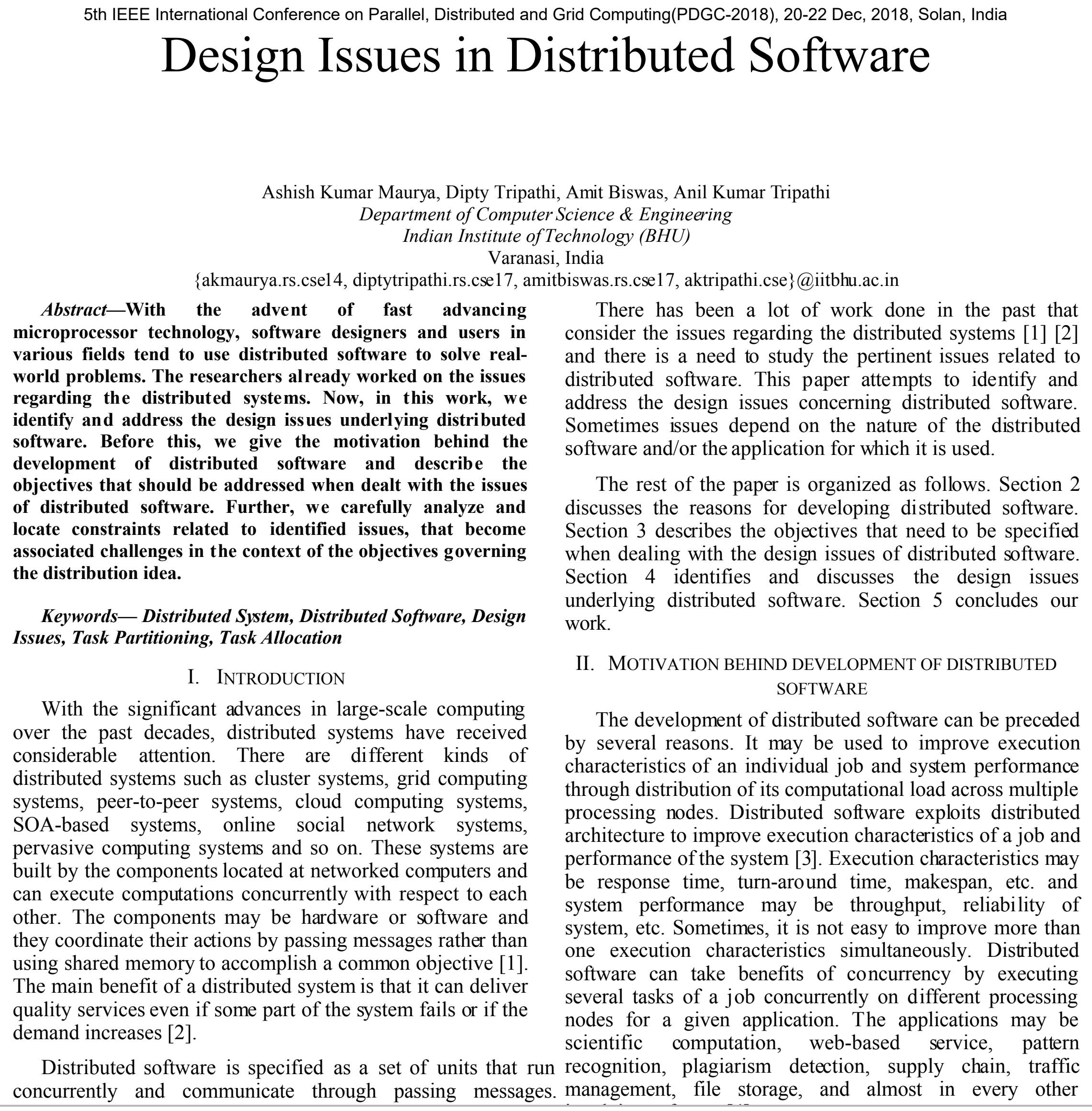
Figures



An instance of distributed softwar…



An instance of software when it is…



5th IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC-2018), 20-22 Dec, 2018, Solan, India

# Design Issues in Distributed Software

Ashish Kumar Maurya, Dipty Tripathi, Amit Biswas, Anil Kumar Tripathi
*Department of Computer Science & Engineering*
*Indian Institute of Technology (BHU)*
Varanasi, India
{akmaurya.rs.cse14, diptytripathi.rs.cse17, amitbiswas.rs.cse17, aktripathi.cse}@iitbhu.ac.in

***Abstract*—With the advent of fast advancing microprocessor technology, software designers and users in various fields tend to use distributed software to solve real-world problems. The researchers already worked on the issues regarding the distributed systems. Now, in this work, we identify and address the design issues underlying distributed software. Before this, we give the motivation behind the development of distributed software and describe the objectives that should be addressed when dealt with the issues of distributed software. Further, we carefully analyze and locate constraints related to identified issues, that become associated challenges in the context of the objectives governing the distribution idea.**

***Keywords*— *Distributed System, Distributed Software, Design Issues, Task Partitioning, Task Allocation***

## I. Introduction

With the significant advances in large-scale computing over the past decades, distributed systems have received considerable attention. There are different kinds of distributed systems such as cluster systems, grid computing systems, peer-to-peer systems, cloud computing systems, SOA-based systems, online social network systems, pervasive computing systems and so on. These systems are built by the components located at networked computers and can execute computations concurrently with respect to each other. The components may be hardware or software and they coordinate their actions by passing messages rather than using shared memory to accomplish a common objective [1]. The main benefit of a distributed system is that it can deliver quality services even if some part of the system fails or if the demand increases [2].

Distributed software is specified as a set of units that run concurrently and communicate through passing messages.

There has been a lot of work done in the past that consider the issues regarding the distributed systems [1] [2] and there is a need to study the pertinent issues related to distributed software. This paper attempts to identify and address the design issues concerning distributed software. Sometimes issues depend on the nature of the distributed software and/or the application for which it is used.

The rest of the paper is organized as follows. Section 2 discusses the reasons for developing distributed software. Section 3 describes the objectives that need to be specified when dealing with the design issues of distributed software. Section 4 identifies and discusses the design issues underlying distributed software. Section 5 concludes our work.

## II. Motivation Behind Development of Distributed Software

The development of distributed software can be preceded by several reasons. It may be used to improve execution characteristics of an individual job and system performance through distribution of its computational load across multiple processing nodes. Distributed software exploits distributed architecture to improve execution characteristics of a job and performance of the system [3]. Execution characteristics may be response time, turn-around time, makespan, etc. and system performance may be throughput, reliability of system, etc. Sometimes, it is not easy to improve more than one execution characteristics simultaneously. Distributed software can take benefits of concurrency by executing several tasks of a job concurrently on different processing nodes for a given application. The applications may be scientific computation, web-based service, pattern recognition, plagiarism detection, supply chain, traffic management, file storage, and almost in every other

The distribution of such software may be of two types: logically distributed and physically distributed. In logically distributed software, concurrent units of software share primary memory and execute on the same multi-core processors while in physically distributed software, concurrent units of software may execute on different processors that do not share primary memory. The physically distributed software incurs processor-to-processor communication cost, which is almost negligible in logically distributed software. When any software requires more processing power and memory, then it is beneficial to distribute software into many modules and execute parallelizable modules on to different processors of the network [3]. Figure 1 shows an instance of distributed software having six modules. Figure 2 and 3 show an instance of software when it is logically and physically distributed respectively. By distributing modules of software across multiple processors, or on different cores of a single processor, better performance can be achieved than without distributing those modules. involving software [4].

Distribution of computational load of software can also be used to ensure greater dependability by exploiting multiple processing nodes through reconfigurability and re-distribution rather than making use of one processing node. Due to this distribution and other arrangements, a system seems to give a single system image to the users. The applications that perform critical tasks such as patient monitoring, spacecraft navigation, chemical plant protection, and bank's customer accounting should not be disrupted by any failure. When executing critical tasks if any processing node fails then it is the responsibility of other processing nodes to perform disrupted computations [5].

The ultimate motivation for developing distributed software is to exploit inherent concurrency in the solution process of the given specific application. There are many applications that implicitly need concurrent execution of its tasks on multiple processing nodes. The applications that

978-1-5386-6026-3/18/$31©2018 IEEE

Citations (2) References (21)

... As distributed system has the capability to reschedule the job of a failed or compromised among the other non-faulty nodes node. This makes the system more fault-tolerant by avoiding a single point of failure [34]. In a recent study, the authors [27] also advocated that some sort of distribution is required to manage the functionality and security at physical and cyber level of CPSs. ...

**An integrated approach of designing functionality with security for distributed cyber-physical systems**

Article Full-text available

Sep 2022 · J SUPERCOMPUT

Dipty Tripathi · Amit Biswas · Anil Kumar Tripathi · Amrita Chaturvedi

View Show abstract

**Preselection Based Leader Election in Distributed Systems**

Chapter

Jan 2022

Amit Biswas · Anil Kumar Tripathi

View Show abstract

Recommendations

Discover more

Article

348 Object Oriented Design of CAE Software with Distributed Components

November 2001 · The Proceedings of The Computational Mechanics Conference

Yasuhiro Kanto · Seiichi Hiruta

Read more

Conference Paper

cTc - A Tool Supporting the Construction of cTLA-Specifications.

March 1996

Carsten Heyl · Arnulf Mester · Heiko Krumm

The design tool cTc has been developed to support the construction and refinement of formal specifications of distributed software systems. It concentrates on the design by stepwise refinement, where refinement steps correspond to the integration of predefined process patterns. cTc processes modules written in the compositional TLA specification style cTLA. It applies the cTLA specification ... [Show full abstract]

Read more

Article Full-text available

Effect of solid slabs on Loads and moments induced in columns of multistoried frames

July 2015

Balaji Kvgd

This paper compares the analysis of a multi-storied structure with and without considering the effect of flexural stiffness of the slab on the distribution of loads and moments in the columns. The study focuses the effect of flexural stiffness of slab in corner columns, penultimate columns and interior columns in the upper most storeys and on the ground storey. STAAD software has been used to ... [Show full abstract]

View full-text

Article

### Software---Practice And Experience

November 2003

Paolo Boldi · Bruno Codenotti · Massimo Santini

this paper we present the design and implementation of UbiCrawler, a scalable, fault-tolerant and fully distributed web crawler, and we evaluate its performance both a priori and a posteriori. The overall structure of the UbiCrawler design was preliminarily described in [2] , [5] and [4]

Read more

Article

### One Kind of Mould's Geometrical Modeling and Finite Element Analysis

March 2014 · Advanced Materials Research

Zhen Long Guo · Hong Sheng Zhao

This paper, by using large-scale finite element analysis software Ansys and assuming that the lining of a mould is under uniformly distributed load, analyses the stress and deformation and on the basis of the result gives an overall optimization design.

Read more