TIME AND COST-EFFICIENT RESOURCE ALLOCATION FOR REAL-TIME APPLICATION IN HIGH PERFORMANCE COMPUTING SYSTEMS

BY

MUHAMMAD SHUAIB QURESHI

A thesis submitted in fulfilment of the requirement for the degree of Doctor of Philosophy in Computer Science

Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

AUGUST 2021

ABSTRACT

High Performance Computing (HPC) is the de-facto platform for deploying real-time applications due to the collaboration of large-scale resources operating in cross-administrative domains. HPC resource scheduling and allocation is a crucial issue in achieving efficient utilization of available resources, especially when resource-intensive applications have real-time deadlines and need data files replicated over the data storage resources. Such scheduling engages both computing and data storage resources to carry out application execution in a timely manner. Traditional approaches are sufficient only when data storage resources are coupled with the computing resources in HPC environment, since data is available at the computing resources for application execution. However, the said domain leaves gaps for deadline miss when data is transferred from remotely located data storage resources to the computing resources where application is being executed. The deadline miss mainly occurs due to the unavailability of the required data files, inadequate scheduling and allocation mechanism of the HPC resources. The problem becomes more complicated when some of the data files are pre-fetched while some post-fetched during application execution which usually results in delayed processing and in turn deadlines miss. The allocation of such resources by considering different optimization criteria such as makespan minimization, cost and energy efficiency, respecting application deadlines, etc. in the aforementioned scenario can be gracefully addressed by designing a scheduling strategy which can result in improved resources utilization while predicting application feasibility. It has always been of interest to the research community to pose the abovementioned situation to determine if the existing scheduling theory and resource allocation strategies are mature enough to accommodate the challenges presented with the emergence of the latest HPC platforms. In this thesis, we explore and analyze the existing resource-allocation techniques for scheduling real-time applications with temporal constraints on HPC platforms (grid, cloud, edge, fog, and multicore systems). This study further compares the resource allocation mechanisms based on different performance parameters and based on existing gaps, a model is proposed which predicts the application schedulability by analyzing time and data constraints before actually dispatching the application to the HPC resources. The main advantage of the prediction-based model is to save time by declining further analysis on unsuitable resources which improve resource utilization by considering application workload in advance. Furthermore, this research thesis devises time and cost-efficient variants of HPC resource allocation with provably correct formulations to cope with the aforementioned problems so that both the user and real-time application constraints are respected. The most celebrated results affirm the supremacy of the proposed techniques in obtaining the desired level of service.

خلاصة البحث

الحوسبة عالية الأداء هي عبارة عن منصة تقوم بالتخصيص المنسّق للموارد واسعة النطاق على التطبيقات الحالية التي يتم تشغيلها ضمن المجالات ذات الإدارات المتداخلة. تعتبر عملية جدولة وتخصيص الموارد ضمن مجال الحوسبة عالية الأداء مسألة في غاية الأهمية حيث ترتبط بتحقيق الاستغلال الأمثل للموارد المتاحة، وخصوصاً عندما تكون هناك مواعيد محددة للتطبيقات ذات الاستهلاك الكثيف للموارد، والتي تتطلب استنساخ الملفات عبر موارد تخزين البيانات. تقوم الجدولة بالتنسيق بين عمليات الحوسبة وبين موارد تخزين البيانات بما يضمن تنفيذ التطبيق في الوقت المحدد. تعتبر الأساليب التقليدية كافية فقط في حالة كان هناك اقتران بين موارد تخزين البيانات وبين موارد عمليات الحوسبة ضمن بيئة الحوسبة عالية الأداء، بحيث يتم توفير البيانات اللازمة لتنفيذ التطبيق ضمن موارد الحوسبة. إلا أن الأساليب المذكورة تؤدي إلى حدوث فجوات تتسبب في تفويت المواعيد المحددة وخصوصاً عندما يتم نقل البيانات من موارد التخزين النائية إلى موارد الحوسبة التي يتم فيها تنفيذ التطبيقات. بشكل عام، فإنه يتم تفويت المواعيد المحددة بسبب عدم توفر ملفات البيانات المطلوبة، وبسبب عدم كفاءة عمليات الجدولة وآليات تخصيص الموارد ضمن بيئة الحوسبة عالية الأداء. كما تصبح المشكلة أكثر تعقيداً عندما يتم جلب ملفات البيانات مسبقاً وجلب البعض الآخر لاحقاً أثناء عملية تنفيذ التطبيق، حيث يؤدي ذلك في العادة إلى تأخر عمليات المعالجة، والذي يؤدي بدوره إلى تفويت المواعيد المحددة. يمكن معالجة مشكلة تخصيص الموارد من خلال إعادة النظر في المعايير المختلفة للتحسين بما في ذلك: تقليل زمن التنفيذ الكلي، وكفاءة التكلفة والطاقة، والالتزام بالمواعيد المحددة، وغيرها ، ضمن السيناريو السابق، حيث يمكن معالجة ذلك بشكل فعال من خلال إعادة تصميم استراتيجية الجدولة بحيث تعمل على الاستغلال الأمثل للموارد، وفي ذات الوقت، تقوم بالتنبؤ بجدوى التطبيق. لطالما كان هناك اهتمام من الباحثين بالجانب أعلاه وذلك لتحديد ما إذا كانت نظرية الجدولة الحالية واستراتيجيات تخصيص الموارد فعالة بما فيه الكفاية لمواجهة التحديات التي برزت مع ظهور منصات الحوسبة عالية الأداء. سنقوم في هذه الأطروحة باستكشاف وتحليل الأساليب الحالية لتخصيص الموارد وجدولة التطبيقات الحالية ذات القيود الزمنية على منصات الحوسبة عالية الأداء بما في ذلك (الأنظمة الشبكية، والسحابية، والضبابية، والطرفية، والأنظمة متعددة الأنوية). كما تقوم هذه الدراسة أيضاً بالمقارنة بين آليات تخصيص الموارد القائمة على أساس معايير الأداء المختلفة، والقائمة على أساس الفجوات، حيث يعمل هذا النموذج المقترح على التنبؤ بالجدوى من جدولة التطبيق عن طريق تحليل القيود الزمنية والبياناتية قبل إرسال التطبيق فعلياً إلى موارد الحوسبة عالية الأداء. تتمثل الميزة الرئيسية للنموذج القائم على التنبؤ في توفير الوقت من خلال رفض طلبات التحليل الإضافية للموارد غير المناسبة، حيث يعمل ذلك على تحسين استغلال الموارد من خلال النظر مقدماً إلى حجم العبء المجدول على التطبيق. علاوة على ذلك، تعمل هذه الأطروحة على ابتكار متغيرات لتخصيص موارد الحوسبة عالية الأداء ذات كفاءة أكبر من ناحية الوقت ومن ناحية التكلفة، وذات إعدادات مناسبة للتعامل مع المشاكل الآنفة الذكر، وبحيث يتم الأخذ في الحسبان كلاً من القيود الآنية للمستخدم والقيود الآنية للتطبيق. تشير أبرز النتائج المنشورة إلى تفوق الأساليب المقترحة في تحقيق المستوى المطلوب من الخدمة.

APPROVAL PAGE

The thesis of Muhammad Shuaib Qureshi has been approved by the following:

Asadullah Shah
Supervisor

Amelia Ritahani Bt. Ismail
Co-Supervisor

Rizal Bin Mohd. Nor
Co-Supervisor

Amir ‘Aatieff Bin Amir Husin
Internal Examiner

Ali Selamat
External Examiner

Radwan Jamal
Chairman

DECLARATION

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Muhammad Shuaib Qureshi

Signature ..... Date ……

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

TIME AND COST-EFFICIENT RESOURCE ALLOCATION FOR REAL-TIME APPLICATION IN HIGH PERFORMANCE COMPUTING SYSTEMS

I declare that the copyright holders of this thesis are jointly owned by the student and IIUM.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Muhammad Shuaib Qureshi

……..…… …….

Signature Date

ACKNOWLEDGEMENTS

I offer heartiest “Darood-O-Salam” to Holly Prophet “MUHAMMAD” (Peace Be Upon Him). I am grateful to almighty ALLAH who is merciful and beneficent, and who enabled me to work on this research successfully. Accomplishment of a research thesis requires the help of many people who steer, guide, give confidence and help you. First, I would like to express my sincere gratitude to my supervisor, *“Prof. Dr. Asadullah Shah”* for his esteemed supervision, encouragement and guidance for successful completion of this research work. I will be forever grateful to him for being a great mentor in my professional life as well as in my personal life. He made my time a precious and joyful memory. I am thankful to my co-supervisors, *Dr. Amelia Ritahani Bt. Ismail and Dr. Rizal Bin Mohd. Nor* for their useful comments, suggestions and guidance. I am thankful to my colleagues, friends and family members who encouraged me to complete this work. My special thanks to my brother *Dr. Muhammad Bilal Qureshi*, who encouraged me at every step of my research work. I am really grateful for his sincere and unconditional help throughout my educational career. Last but not the least, special thanks to my parents for their unconditional, unwavering, and untiring support, wishes, and encouragement. They have been my strongest motivation throughout my thesis.

.

Muhammad Shuaib Qureshi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBRIVIATIONS

HPC	High Performance Computing
QoS	Quality of Service
RA	Resource Allocation
DVS	Dynamic Voltage Scaling
RDTA	Real-time Data-intensive Tasks Allocation
GA-SS	Steady State Genetic Algorithm
GA-ST	Struggled Genetic Algorithm
GA-EG	Elitist Generational Genetic Algorithm
PPDPS	Partition Problem based Dynamic Provisioning and Scheduling
DSB	Dynamic Scheduling of Bag of Tasks
MAHP	Modified Analytic Hierarchy Process
BATS	Bandwidth Aware Tasks Scheduling
LEPT	Longest Expected Processing Time
AGA	Adaptive Genetic Algorithm
EFTD	Earliest Finish Time Duplication
PRS	Proactive and Reactive Scheduling
ATS	Allocation-aware Task Scheduling
DGP	Developmental Genetic Programming
DCLS	Dynamic Cloud List Scheduling
DCMMS	Dynamic Cloud Min-Min Scheduling
EDF	Earliest Deadline First
DVFS	Dynamic Voltage Frequency Scaling
HGCS	Hybrid Genetic and Cuckoo Search
PTPS	Purely Time-driven Periodic Server
WCPS	Work-Conserving Periodic Server
CRPS	Capacity Reclaiming Periodic Server
USG	Unfair-Semi Greedy
EDZL	Earliest Deadline until Zero-Laxity
EARH	Energy Aware Rolling Horizon
GCSM	Green Cloud Scheduling Model
FDHEFT	Fuzzy Dominance sort based Heterogeneous Earliest Finish Time
DA-EDF	Data Aware – Earliest Deadline First
EDA-EDF	Enhanced Data Aware – Earliest Deadline First
RS	Resource Scheduling
RM	Resource Matching/Rate Monotonic
HCS	Hybrid Cuckoo Search
LFS	Least Feasible First
CATP	Compatibility Aware Task Partition
G-CATP	Group-wise Compatibility Aware Task Partition
SCRUP	Simple Combined Resource Usage Partitioning
ENCAP	Enhancing Shared Cache Performance
TDA	Time Demand Analysis
DEFT	Dynamic Fault-Tolerant Elastic

NOTATIONS AND DESCRIPTION

T	Task set
r_k	Release time of task *k*
e_k	Execution time of task *k*
d_k	Deadline of task *k*
CP_y	Computing power of resource *y*
PNP	Positive-negative points set
DF_k	Data files set required by task *k*
EET_{ky}	Execution time of task *k* on resource *y*
Υ_x	Tasks group *x*
GU_{Υ_x}	Group utilization of tasks group *x*
TU_i	Utilization of task *i*
CD	Compute and storage resource pair
CR	Computing resource set
t_P	Positive point
t_N	Negative point
TT	Total execution time
DR	Data storage resource set
$\mathbb{R}_w$	Response time of the storage resource *w*
f_{kz}	File *z* needed by task *k*
FT_{fkz}	Transfer time of the file f_{kz}
$card(T)$	Cardinality of task set *T*
dr_w	Data storage resource *w*

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

High Performance Computing (HPC) is an attractive platform for both academic research and ICT trade to execute computational-intensive applications which need powerful resources for generating celebrated results. Many of such applications are time critical which needs in time response for completion. Such applications are known as real-time applications. In real-time applications, the correctness of the system not merely depends on the produced results but the time in which these results are obtained. Real-time systems are characterized by some parameters like computation time, period, and deadline. The computation time is the system's execution demand for the computing resource. Each real-time system generates infinite instances called as jobs. A job is generated after a specific time interval called as period. The time before which the application should complete its execution is called as deadline. Depending on the consequences of the missed deadlines, the real-time systems are broadly categorized into hard and soft real-time systems (Qureshi, Alrashed, Min-Allah, Kolodziej & Arabas, 2015). In hard real-time systems, a deadline meeting is the most critical constraint. Examples of such systems include railway switching system, air traffic control system, nuclear plant control system, and military system. The hard-real-time systems must respect the deadline constraints. In soft real-time systems, there is a gap for deadlines miss which may not result in catastrophic behavior but system's performance degradation. Examples of soft real-

time systems include automated teller machines, virtual reality, multimedia systems, interactive computer games, mobile robotics, and telecommunication networks.

A real-time system is composed of concurrent programs known as tasks (Laplante, 2004). A real-time task can be defined as an executable entity of work that is characterized by deadline and execution time which is the maximum estimated time needed by a processor to complete the task. This time is termed as worst-case execution time. By considering the aforementioned two basic characteristics, a real-time task i is denoted by $\tau_i(c_i, d_i)$, where c_i and d_i represent execution time and deadline of the task i respectively. Since, real-time systems are time sensitive systems, so scheduling such systems got massive popularity in the literature and numerous algorithms for different scenarios have been proposed (Zhang, Tian, Fidge, & Kelly, 2016). Some of the scheduling algorithms guarantee in advance that the application constraints will be met during execution. Such algorithms are known as static scheduling algorithms. In static priority assignment algorithms, the tasks priorities once set remain constant throughout the execution of the task. The well-known static priority assignment algorithm is rate-monotonic (RM) algorithm. The RM algorithm prioritize tasks on the basis of their rates: the higher is the rate, the higher is the priority. The rate of the task is inversely proportional to the period of the task i.e., $rate = \frac{1}{period}$. The other class of algorithms prioritizes application during execution. Such algorithms are known as dynamic scheduling algorithms. Both algorithms classes characterize tasks with additional parameters, which are used to analyze performance of the system.

The HPC paradigm is attractive platform for deploying real-time applications mainly for three reasons:

1. the time sensitive nature of the real-time application requires parallel processing on distributed powerful resources to generate results in a timely manner,
2. the concurrent execution on many high-speed interconnected nodes as compared to a single powerful CPU is economical to efficiently achieve the desired level of performance, and
3. the distributed systems are highly reliable in cases of system failure.

But the existing HPC platforms still face many challenges due to the heterogeneous nature of distributed resources like predicting system behavior in peak load conditions (when all tasks occur at critical instant), completing tasks with minimum total execution time and user budget, proper load balancing on resources, on-time resource provisioning, dealing with task and resource heterogeneity, fault tolerance, a-priori time management requirements, and so on. Such challenges pave the way for further investigations and need to be addressed properly by developing adequate scheduling mechanisms to execute real-time applications within deadlines while meeting user QoS criteria. A proper resource allocation (RA) mechanism improves performance of all HPC classifications (Hussain, Malik, & Khan, et al., 2013; Qureshi, Dehnavi, & Min-Allah, et al., 2014).

Currently, task scheduling and resource allocation techniques attracted researchers towards HPC platforms considering diverse optimization criteria of virtual machines (VMs) renting cost, makespan minimization, QoS maximization, energy efficiency, and so on according to predefined agreed SLAs (Liu et al., 2015; Sangwan et al., 2016; Awad. A et al., 2015; Chen. H et al., 2015; Wu. X et al., 2013; Panda et al., 2015; Satish & Reddy, 2018). In this thesis, we use system, application, task, and job interchangeably.

A generalized RA scenario in HPC systems is represented in Figure 1.1 High-level view of resource allocation in HPC system A user submits request for RA to the broker which finds the resources status form the HPC information service directory. This directory holds information about resources. Based on this information, a large pool of resources is searched and resources which fulfil user QoS criteria are selected for application execution. The user application is submitted to the selected resources for execution. After successful execution of the application, the results are returned to the user.

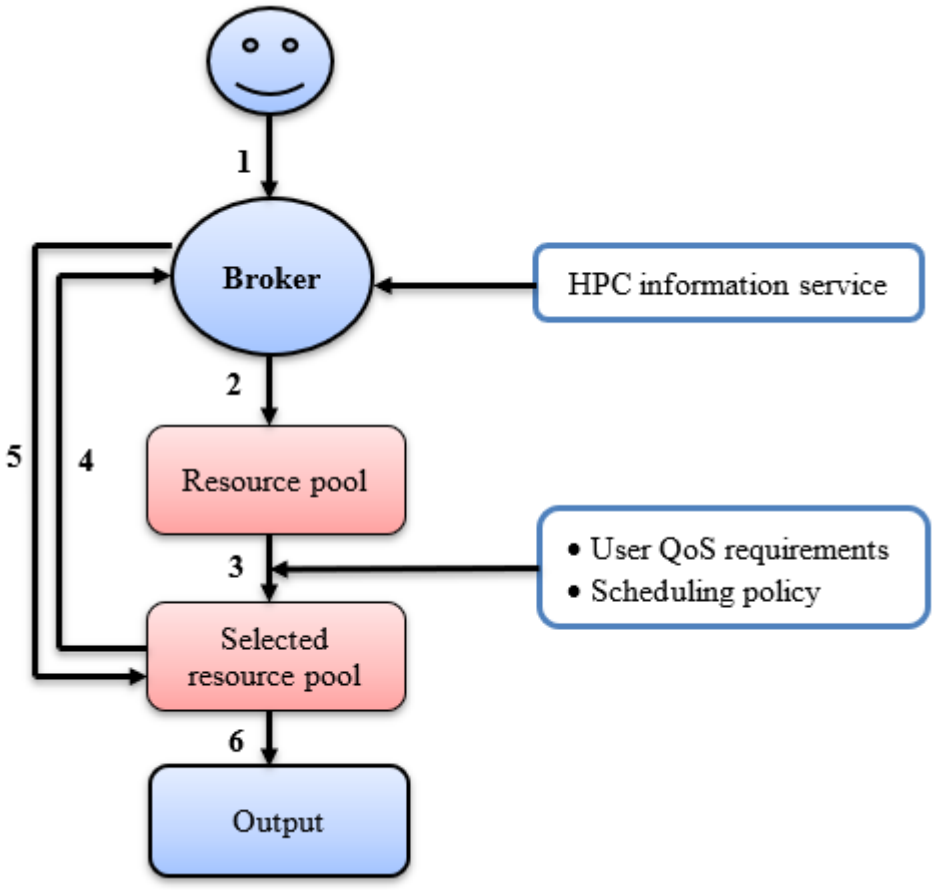

Figure 1.1 High-level view of resource allocation in HPC system

1.2 PROBLEM STATEMENT

It is observed from literature that most of the existing solutions to the real-time task's allocation problem in HPC domain provide room for deadline miss when such applications need data files for processing. This type of applications is called as data-intensive real-time applications. The data and communicational aspects between computing and storage resources are unaccounted in the existing resource optimization (scheduling and mapping) settings when data storage resources are located remotely from the computational resources. The feasibility testing of the real-time tasks under fixed priority scheduling technique has always been challenging in data-intensive real-time applications when some of the required data files are pre-fetched while some post-fetched during tasks execution. The literature also lacks comprehensive mechanism for predicting tasks feasibility prior to execution on different HPC resources when time and data constraints are considered. This gap provides opportunity for tasks scheduling on non-feasible resources and hence deadlines miss. From data files processing aspects, it has also been of interest to include the data files transfer time in deciding tasks feasibility on computational resources. The research community focused on the deadline's fulfilment concern of the real-time applications execution and ignores the user budget constraints.

The aim of this research is to develop a novel and fine-tuned resource scheduling and allocation policy for real-time application on HPC resources to cope with the aforementioned problems. We believe that this attempt will result in remarkable contributions towards plethora of existing real-time systems scheduling literature. Based on the nature of research work and above discussed problems, it is mainly divided into three modules (portrayed in Figure 1.2) with clear objectives.

In Module – I, we study and analyze the real-time resource allocation (RA) strategies in HPC domain. The performance of each RA strategy is evaluated on the basis of common parameters. The Module – I describes and pictorially represents each studied strategy in detail which helps in understanding working of each mechanism.

Module – II devises a prediction-based resource allocation model for real-time data-intensive application. The developed model checks the feasibility of tasks before actually allocating and dispatching tasks to the computing resources. The research work in Module – II devises a new scheduling model for scheduling real-time application to enhance schedulability of tasks by considering different optimization criteria.

In Module – III, we propose a resource allocation strategy for real-time data-intensive tasks by ranking computational resources and classifying tasks into groups that guarantee tasks execution with minimum possible time and cost while deadlines are kept intact. The ranking technique helps in selecting the most appropriate resources for application scheduling. The obtained results affirm the supremacy of the proposed technique over the existing counterparts.

1.3 RESEARCH OBJECTIVES

The aims and objectives of this research are:

1. To collect and pictorially present the existing resource scheduling and allocation techniques in different HPC systems (grid, cloud, fog, edge, and multicore) at one place under the umbrella of real-time literature by considering common performance parameters.
2. To propose a two-stage model where the first stage predicts the feasibility of real-time application before actually dispatching to the HPC resources

and, the second stage schedule the application on the feasible resources by considering objective function.

3. To propose a time and cost-efficient resource allocation strategy for data-intensive real-time application in the HPC system which reduces the priority levels.

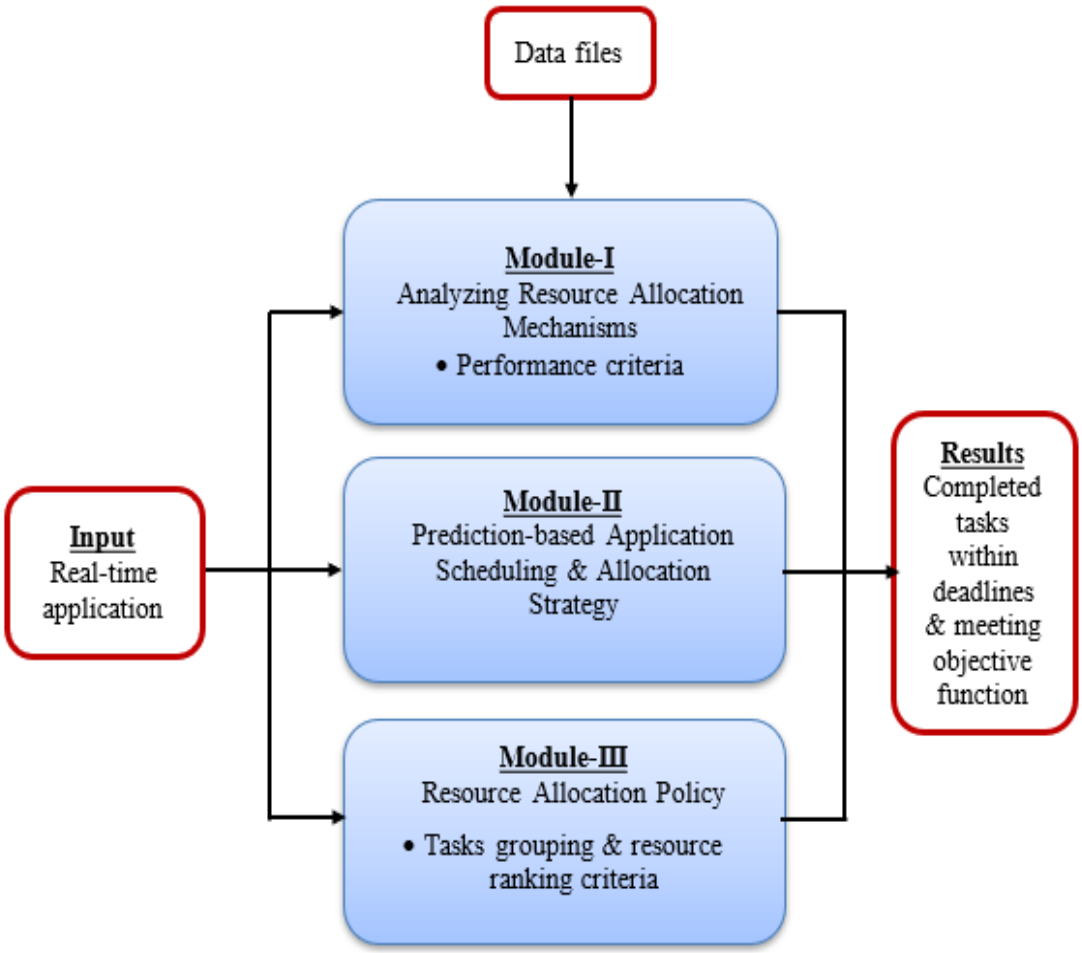

Figure 1.2 Proposed resource allocation scenario.

1.4 ULTIMATE RESEARCH QUESTIONS

The HPC platforms offer celebrated capabilities if the encompass distributed resources are managed in an efficient way across heterogeneous environments. The existing

literature is still immature in modeling the HPC resources in a comprehensive manner which leave some questions open ended that need to be answered. The ultimate questions that the proposed resource allocation research addresses are:

1. How HPC paradigm can help more efficiently in scheduling and processing real-time applications within deadlines as compared to other platforms?
2. What issues can arise during real-time application scheduling and mapping, specially, when both computing and storage resources are heterogeneous in nature and involved in resource allocation strategy?
3. How can it be verified in advance whether the real-time application constraints will be met during execution or not?
4. How the resource allocation strategy allocates resources and executes real-time data-intensive tasks in minimum possible execution time and limited budget constraints while respecting tasks deadlines?
5. What effects the transfer time put on the tasks total execution time and cost when data files are replicated on remotely located storage resources?

1.5 SIGNIFICANCE/EXPECTED OUTCOMES OF THE RESEARCH

The HPC is resources rich paradigm which is considered the most promising platform for deploying real-time applications. The increasing complexity of the real-time applications make HPC model a handy candidate to place such systems. The accuracy of generated results deliberately depends on the resource allocation mechanism. An efficient resource allocation mechanism can contribute to the best performance of the resources in handling time critical applications in an eminent fashion, especially, when resources are heterogeneous in nature.

This research delivers broadly to the scheduling theory that opens new dimensions for extending the existing mechanisms in broad ways which contributes to the society. Schedulers and resource allocation mechanisms are considered to be the core components of operating systems, parallel systems, and real-time embedded systems. It helps researchers how to take advantage of powerful resources in executing time-conscious applications in an efficient way such that the deadlines are respected. Moreover, the proposed model is flexible in order to adapt to the future paradigms and multiple domains by future researchers and scientists to immensely improve the throughput and reliability of the systems.

This research work advances the current state-of-the-art of real-time scheduling theory using HPC platforms as follows.

1. *Identification of the positive and negative scheduling points*

   The proposed work identifies the impact of negative points on deadlines miss during scheduling real-time application. Such points can be disjointed from the positive points in a set which need not to be checked during feasibility testing of the real-time application on HPC resources. This mechanism can help in reducing application completion time which in turn ensures fulfilling the deadlines.

2. *Specification of criteria for the selection of computing and storage resources for data-intensive real-time application*

   This research accumulates real-time application which needs data-files for complete execution. The data-files are replicated on distributed data storage resources connected to the computing resources by network links. The proposed technique designs a criterion for the selection of storage and