

Research Article

Augmenting High-Performance Mobile Cloud Computations for Big Data in AMBER

Muhammad Munwar Iqbal,¹ Muhammad Ali,¹ Mai Alfawair,² Ahsan Lateef,³
Abid Ali Minhas ,⁴ Abdulaziz Al Mazyad,^{4,5} and Kashif Naseer ⁶

¹Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan

²Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa' Applied University, Salt, Jordan

³Department of Computer Science, University of Agriculture, Faisalabad, Pakistan

⁴College of Computer and Information Systems, Al Yamamah University, Riyadh, Saudi Arabia

⁵King Saud University, Riyadh, Saudi Arabia

⁶Department of Computer Engineering, Bahria University, Islamabad, Pakistan

Correspondence should be addressed to Abid Ali Minhas; a_minhas@yu.edu.sa

Received 30 August 2017; Accepted 30 January 2018; Published 2 April 2018

Academic Editor: Syed H. Ahmed

Copyright © 2018 Muhammad Munwar Iqbal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Big data is an inspirational area of research that involves best practices used in the industry and academia. Challenging and complex systems are the core requirements for the data collation and analysis of big data. Data analysis approaches and algorithms development are the necessary and essential components of the big data analytics. Big data and high-performance computing emergent nature help to solve complex and challenging problems. High-Performance Mobile Cloud Computing (HPMCC) technology contributes to the execution of the intensive computational application at any location independently on laptops using virtual machines. HPMCC technique enables executing computationally extreme scientific tasks on a cloud comprising laptops. Assisted Model Building with Energy Refinement (AMBER) with the force fields calculations for molecular dynamics is a computationally hungry task that requires high and computational hardware resources for execution. The core objective of the study is to deliver and provide researchers with a mobile cloud of laptops capable of doing the heavy processing. An innovative execution of AMBER with force field empirical formula using Message Passing Interface (MPI) infrastructure on HPMCC is proposed. It is homogeneous mobile cloud platform comprising a laptop and virtual machines as processors nodes along with dynamic parallelism. Some processes can be executed to distribute and run the task among the various computational nodes. This task-based and data-based parallelism is achieved in proposed solution by using a Message Passing Interface. Trace-based results and graphs will present the significance of the proposed method.

1. Introduction

Big data challenges are opportunities for the academic as well as for the industry. Data processing and analytics of high-performance computing machinery provide the possibilities for designing complex business models, highly scalable and reconfigurable systems. The main interest of this research is to provide researchers with desktop-based machines or virtual machines, having the capacity of massive processing to run the jobs that require high-end hardware like supercomputers.

Cloud computing can be used to fulfil the requirements for the computation-intensive jobs.

Cloud computing running programs that run across homogeneous platforms on multiple nodes consists of multiple virtual machines. MPI provides parallel computing software infrastructure, task-based and data-based parallelism. This solution is cost-effective and efficient as compared to previous ones. Cloud computing data security and privacy are enhanced by using the Single Sign-On approach. This approach uses identity management techniques like OAuth,

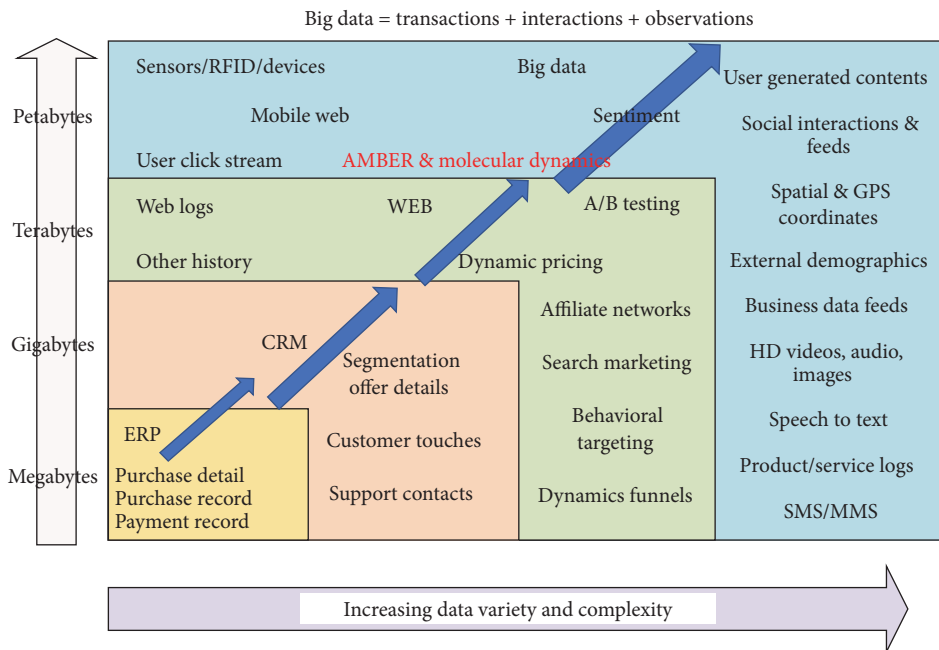


FIGURE 1: Multimedia big data particular levels [4, 5].

TABLE 1: Big data challenges with cloud computing.

Big data/cloud computing	Elasticity	Pooled	On-demand	Self-service	Pay-as-you-go
Volume		x			x
Velocity	x		x		
Variety	x	x		x	
Veracity				x	x
Value	x		x		x

OpenID, and SAML for the enhancement of cloud users' security and privacy [1].

The main challenges are content management, security, transmission, information retrieval, and mining in multimedia databases [2]. While redefining the possibilities of molecular dynamics and AMBER [3], cloud computing has engaged big data and enlightened potential solutions for various digital earth problems in geoscience and relevant domains such as social science, astronomy, business, and industry as shown in Figure 1.

In this research, energy calculation of AMBER force fields of molecular dynamics has been selected to prove the concept. This method is used to have an estimation of molecular structures and energies for confirmations of physical properties. The Internet is the primary source of the multimedia data produced on a large scale and freely available.

The features of cloud computing and their utilization to support characteristics of big data are summarized Table 1.

Molecular dynamics is a computer-based recreation of the developments of particles considering N -number of atoms as an N -body reenactment. The atoms and particles associated with each other for a characterized time frame, accordingly, furnish us with a perspective of the translational

developments of molecules. In the regular adaptation of reproductions, Newton's conditions of movement for an arrangement of N -communicating particles are unravelled numerically to watch the directions of iotas and atoms. The strengths between the particles and potential vitality are ascertained by utilizing any one sort of compelling fields of subatomic mechanics. It is a procedure of hypothetical physical science that was considered in the decade of 1950s, yet it is received right now for synthetic and physical investigation, materials science examination, and the displaying of biomolecules [6].

2. 7 Vs of Big Data

The cloud and IoT data are growing exponentially, that is, previously in kilobytes, megabytes, and gigabytes. The big data that was previously comprised of 4 Vs, but now it is pretty well changed into 7 Vs data, which are velocity, variety, variability, veracity, visualization, value, and volume. Molecular dynamics contains the massive volume of unstructured and structured data referred to as big data. The term "big data [3]" is originated from the web companies and they had to query the loosely structured large shared data. Big data is composed of seven principal terms, which signify

its importance. These terms are velocity, variety, variability, veracity, value, volume, and visualization. Velocity shows the speed of data generation, processing, and storage. It depicts the data that can be accessed with enough speed.

Variety is the most significant challenge in big data and is a range of structured, semistructured, and unstructured data. These data sources have many data types like XML, audio, video, SMS, and sensor data. The organization of the data is not an easy task; mainly when the data types are different, data format is heterogeneous; multifactor data values and probabilistic data are also available for processing. Variability is different from the variety and veracity like the taste of six different blends of coffee at the shop. Since data meaning is continually changing, metadata also changes accordingly on homogenization of data.

Veracity means to ensure the data that is required for processing is accurate and keep away bad, dirty data from collecting into systems for execution, like entering false and incorrect names and wrong contact information into databases. Visualization is an essential and critical element of big data. The sophisticated data visualization uses graphs, charts, modeling tools, and techniques with reported chock-full formulae and numbered values.

Value is the most effective element of the big data after putting whole effort and addressing the volume, velocity, variety, variability, veracity, and visualization. The organization must be getting values from data which is beneficial for setup. Big data now possibly define the data in zettabytes and yottabytes. The volume of data is currently increasing in gigabytes per minutes.

The big data are retrieved from heterogeneous and multiple resources, as well as their ingredients. Therefore the complexity of data is the core concern. The big data must be cleansed, matched, transformed, and linked into chosen formats before processing [7]. The other examples are Facebook and Twitter, which are generating data through user social interaction [8].

There are numerous techniques to figure out the energies and constraints; one path is by computing the traditional compelling fields or a quantum hypothesis at close level. The number count and the speeds of particles are introduced with starting self-assertive qualities that match the aggregate entirety of dynamic energies of the framework, which, thus, is further managed by the required temperature of reproduction. By having the estimations of the drive on every molecule, we can ascertain the increasing speed of every particle in the entire framework. Count of the integral of Newton's conditions of movement then decides a direction that depicts the quickening, position, and speed of the iotas and atoms which shift with time. The standard estimations of properties can be computed by keeping in view these directions. It is known as a deterministic strategy for figuring. In this technique, the speeds and positions of every single iota are utilized. By running the reenactment, these qualities can be ascertained for any moment of time. The time moment can browse the future or the past. Subatomic elements recreations are tedious and computationally costly if accomplished for expensive frameworks [9].

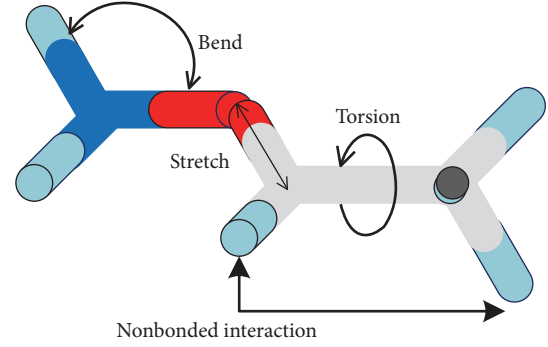


FIGURE 2: Force fields bonded and nonbonded interactions.

Molecular dynamics reproduction is one of the fundamental techniques in the domain of subatomic rebuilding and is utilized for subatomic examination [10, 11]. Atomic reenactments give us dependable association in force fields. They help us to learn the plainly visible thermodynamic properties of particles, for example, compressibility, inward vitality, weight, warm extension, particular warms, pliable and shear modulus that can be assessed. It is finished by utilizing the infinitesimal level data acquired by the after-effects of reenactments [10].

Specific commands are utilized to extricate force field information from the prototypical model. It is also used for the vitality expression or capacity calculation of the particular model. This specific vitality expression is the condition that communicates the possible vitality on the surface of a specific model. This is portrayed as an element of its nuclear directions. The sum of valence or reinforced and nonfortified associations can be communicated as the potential vitality of a framework. The corner ordinarily represents the energies required in valence connections to corner terms. These energies are named as bond extend term (E_{bond}), valence point twist term (E_{angle}), dihedral edge torsion term (E_{torsion}), and reversal ($E_{\text{inversion}}$) terms. These vitality terms are a piece of all constraint fields vitality estimations for covalent frameworks as shown in Figure 2.

The E_{val} is represented the by

$$E_{\text{val}} = E_{\text{bond}} + E_{\text{angle}} + E_{\text{torsion}} + E_{\text{inv}}. \quad (1)$$

The interactions' energy of the nonbonded atoms is calculated by considering van der Waals (E_{vdW}), electrostatic (E_{Coulomb}), and hydrogen bond (E_{hbond}) terms as shown in

$$E_{\text{nonbond}} = E_{\text{Coulomb}} + E_{\text{vdW}} + E_{\text{hbond}}. \quad (2)$$

AMBER is a method with force fields energy calculation for molecular dynamics. The equation for this type of force fields energy calculation defines the total potential energy of the molecules. The AMBER in minimization of the bond stretching energy is shown in Figure 3. This force is dependent concerning the position and its potential derivative of the equation to calculate this energy is called a functional form of AMBER.

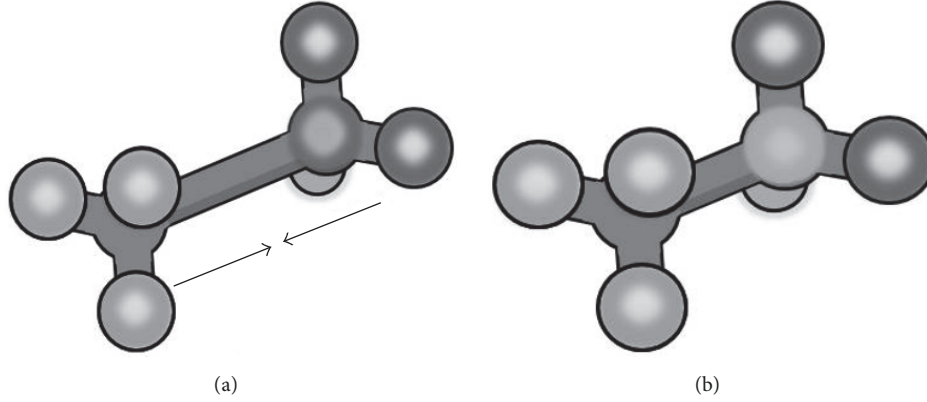


FIGURE 3: AMBER in minimization of the bond stretching energy.

The AMBER force fields functional form is described as the equation mentioned below [12, 13]:

$$\begin{aligned}
 V(r^N) = & \sum_{\text{bonds}} k_b (l - l_0)^2 + \sum_{\text{angles}} k_a (\theta - \theta_0)^2 \\
 & + \sum_{\text{torsions}} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] \\
 & + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \left\{ \epsilon_{i,j} \left[\left(\frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{0ij}}{r_{ij}} \right)^6 \right] \right. \\
 & \left. + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}, \quad (3)
 \end{aligned}$$

where k_a is the Angle Constant; θ is the angle bend; ϵ is the well depth; k_b is the bond constant; ω is the torsion angle; V_n is the torsion constant; l_0 is the natural bond length; l is the bond length; q_i is the charge on i ; q_j is the charge on j ; and γ is the phase factor. The expression on the right-hand side of AMBER force fields functional form is described in detail as follows: [14, 15]

- (i) Summation over bonds is the first term, used to calculate the energy of the atoms that are bonded covalently. This harmonic force is also known as ideal spring force. It gives us good approximation close to the bond length equilibrium, but this force becomes poorer as the distance between atoms increases.
- (ii) Summation over angles is the second term, used to present the energy, which is due to the geometrical placement of electron in orbitals involved in covalent type bonding.
- (iii) Summation over torsions is the third term, used for the representation of the energy due to twisting a bond as a result of the bond order, double bonds, and neighbouring electrons in lone pairs or bonds. A solo bond can have one or more than one term; for example, the total torsional energy can be expressed as a Fourier series.
- (iv) Double summing for i and j is the fourth term, used to express the nonbonded energy among all the pairs

of atoms. This term can further be decomposed into two terms: the first term of the summation is van der Waals forces and the second term of the summation is electrostatic energies.

The equilibrium distance (r_{0ij}) with depth (ϵ) is used for calculating the van der Waals energy, and equilibrium distance is r_{0ij} ensured by the factor of 2, in the calculation. Terms of σ , where $r_{0ij} = 2^{1/6}(\sigma)$, are sometimes used for the reformulation of energy. It may be used for example, in the implementation of the soft-core potentials [16].

The researcher reported in this investigation that the computation-intensive architecture was used free from geolocations for the computation of the Assisted Model Building with Energy Refinement (AMBER) with the force fields calculations for molecular dynamics. The proposed scheme is also useful for providing mobility to the calculation of the computation-intensive jobs, where Internet connectivity is not possible. The proposed methodology is providing software based instead of previously proposed solutions, which are costly and hardware based and setup is more critical. This proposed solution is composed of open source platform which minimized the computation cost.

3. Literature Review

OpenCL is embraced by Apple, Intel Xeon Phi, Qualcomm, Samsung, Vivante, Advanced Micro Devices (AMD), Altera, Nvidia, and ARM Holdings [17, 18]. The force fields figuring is an experimental computation technique proposed to give evaluations of structures and energies for adaptations of particles. The enhancement and conformities required to run the computation-intensive jobs need a testbed composed of very costly hardware. The investigation is also helpful for the exascale calculation in future [5]. The technique depends on the presumption of "common" bond lengths and points, deviation from which prompts strain, and the presence of torsional connections and appealing or potentially terrible van der Waals and dipolar powers between nonfortified molecules [19]. The strategy is called experimental drive field gullible.

Molecular dynamic (MD) progression is a computer recreation of physical developments of particles and atoms with regard to N -body reproduction [20]. The atoms and particles are permitted to collaborate for a timeframe, giving a perspective of the movement of the molecules. In the most widely recognized adaptation, the directions of atoms and atoms are numerically dictated by measuring Newton's conditions of movement for an arrangement of interfacing particles, where subatomic mechanics characterizes drives between the atomic particles and potential vitality compelling fields. The strategy was initially considered inside theoretical physical science in the late 1950s, yet is connected today for the most part to synthetic material science, materials science, and the displaying of biomolecules [21].

Electrostatic energy can be used here to assume single point charge in the equation. It means that a single point charge can represent both electrons and protons in an atom or in the case of parameter sets that employ point charges and electron lone pairs. If we consider a conventional molecular dynamic (MD) simulation, the most processor-hungry job is the calculation of the force field energies as a function of the internal coordinates of atoms and molecules that are under considerations. In the evaluation of that energy, the noncovalent and nonbonded energy calculations are a most computationally intensive part. In the view of complexity, big "O" notation of the MD calculations is scaled by $O(n^2)$. This expression is valid only if we explicitly consider all pair-wise electrostatic and van der Waals interactions for force fields energy calculation [22].

One may assess the energies and powers utilizing traditional drive fields or a chose level of quantum hypothesis. By and by, the atoms are allocated beginning speeds that comply with the aggregate dynamic vitality of the framework, which, like this, is directed by the carved reproduction temperature. From the information of the compel on every molecule the increasing speed of every particle in the framework can be decided. Incorporation of the conditions of movement then yields a direction that depicts the positions, speeds, and increasing speeds of the particles as they differ with time. From this direction, the normal estimations of properties can be resolved. The technique is deterministic; once the positions and speeds of every molecule are known, the condition of the framework can be anticipated, whether later on or in the past. Subatomic elements reproductions can be tedious and computationally costly [23]. Atomic progression (MD) recreation is one of the essential strategies in the subatomic reproduction world [24]. Through atomic reenactments with reliable communication compelling fields, the naturally visible thermodynamic properties, for example, weight, inner vitality, warm development, compressibility, tractable and shear modulus, particular warms, can be assessed by utilizing the minute level data produced through recreations [25].

Mixture parallel computing is one approach for addressing the issues of processor-hungry molecular dynamics recreation. Its columns depend on basic heterogeneous processing framework, climate CPU/GPU blend, or Intel's Xeon Phi coprocessor. Heterogeneous figuring refers to frameworks that use more than one sort of processors. These are multicenter frameworks that pick up execution by including centers,

as well as by fusing specific preparing abilities to handle specific assignments. Heterogeneous System Architecture (HSA) frameworks use various processor sorts (typically CPUs and Intel Phi coprocessors), for the most part on similar silicon pass on, to give you the best of both spaces. First is vector handling; aside from its outstanding parallel preparing abilities, it can likewise perform numerically concentrated calculations on huge information sets. Second is CPUs that can run the practical framework and perform conventional serial undertakings [26, 27].

Seven frameworks on the top 500 rundowns were at that point utilizing the Intel chip. The top supercomputers, Intel's quickening agent share is still a minute, machines utilizing Nvidia's GPUs dwarf, those with Intel's coprocessors 31 to 11. Tianhe-2 has significantly affected the scene. In the event of counting up all the peta-tumbles on the rundown, "the collected execution conveyed by Intel Xeon Phi coprocessors is currently greater than the execution conveyed by GPU quickening agents," says Intel representative Radoslaw Walczyk. "It is a major win," says Sergis Mushell, an examiner at the innovation inquiries about firm Gartner. Intel's chips contain up to 61 centers and are assembled utilizing the organization's 22-nanometer fabricating process, which is an era in front of the opposition. The organization says its coprocessors have a couple of focal points over GPGPUs: they can work freely of CPUs and they do not require a unique code to program [28].

Intel gives C and C++ languages compiler to Intel Xeon Phi coprocessor. OpenCL is summed up design for programming accessible on all top-of-the-line gadgets (mobile GPUs, AMD GPUs, Intel gadgets). C and C++ languages are recommended to meet the necessities of hybrid parallel computing as its streamlining instruments, for example, V Tune is likewise accessible by Intel to upgrade our application [27, 29, 30].

An amazing drive field, situated on the atomistic depiction of the silicon dioxide statement on a melted silica substrate, has been produced and connected to the subatomic element recreation with the GROMACS bundle. The legitimacy of the created recreation approach is checked utilizing nuclear bunches comprising up to 106 molecules and having trademark measurements of up to 30 nm. The C and C++ languages advancement apparatus is bundled with Intel Composer XE. This bundle incorporates the libraries, debuggers, and the compilers. It has instruments to fabricate the offload and cross-incorporated form of the hotspot for Intel Xeon Phi coprocessors. It perceives compiler-upheld punctuation and its utilization concerning Intel Xeon Phi [30, 31]. The OpenCL Applications XE 2013 R2 uses Intel SDK for a product advancement environment for OpenCL applications for Intel Xeon processors and Intel Xeon Phi™ coprocessors [31]. The SDK gives documentation, improvement devices, and the OpenCL 1.2 runtime for Intel Xeon Phi coprocessors and Intel Xeon processors. The Intel SDK for OpenCL Applications XE 2013 for Linux is utilized. The SDK bolsters both the Intel Xeon server and Intel Xeon Phi coprocessor.

The Intel Xeon Phi coprocessor is the main item given, Intel Much Integrated Core Architecture (Intel MIC engineering), and it targets HPC sections, for example, oil

TABLE 2: Parameters required to calculate bond stretch terms and angle bend terms.

Sr. No.	Parameter	Description
(1)	“forceConstant”	Constant variable used in calculation
(2)	“equilibrium distance.”	Distance between two atoms
(3)	“angleBendTerms [ix5]”	All the four values of angle bend terms according to functional form
(4)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(5)	“dihedralAngleTerms [ix7]”	All the four values of dihedral angle terms according to functional form
(6)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(7)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(8)	“nonBondedTerms [ix4]”	All the four values of nonbonded terms according to functional form
(9)	“coulomb factor.”	It is calculated using permittivity and permittivity scale and a constant value 332
(10)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(11)	“bond length.”	The natural bond length
(12)	“energy”	The partial value of energy due to bond stretch terms is calculated and stored in it
(13)	“angleBendTerms [ix5]”	All the four values of angle bend terms according to functional form
(14)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(15)	“theta deg”	It is calculated in this function from coordinates
(16)	“term”	Intermediate value to calculate energy due to angle bend
(17)	“energy”	The partial value of energy due to angle bend terms is calculated and stored in it

investigation, logical research, money detailed examinations, and atmosphere reproduction, among numerous others. Intel MIC design joins various Intel CPU centers onto a single chip. Engineers keen on programming these centers can utilize standard programming techniques. Similar OpenCL source code composed of Intel Xeon processor can be reused with minor changes on the Intel Xeon Phi coprocessor [32, 33].

4. Proposed Mobile Cloud Methodology

Mobile Cloud Computing bases offered by VMware and open MPI innovation can be utilized to meet the necessities effectively [34]. C and C++ languages can be utilized to program the issue and execute the design of High-Performance Mobile Cloud Computing for proposed stages. Consolidated performance of VM machines is altogether higher than single equipment machine/processor, because of its high asset capacities. The devices given by versatile distributed computing, for example, VMware alongside open MPI, to run an issue in a parallel design can upgrade the execution of use on Mobile Cloud Figuring Infrastructure [35].

High computational tasks have been accomplished through hardware solution which is costly. The proposed solution is software based, one which comprises virtual machines having a hypervisor installed in it and programming utilities/software to execute the required computations. This software is installed on the VMware machines to run the distributed computational task.

The function “double calcBondStrechTerms (int begin, int finished)” takes two integer variables “start” and “end” as input and calculates results as a variable type double. The required parameters and constants are obtained from a file named “amber parameters.” It uses the parameters mentioned in Table 2.

The function is implemented and mentioned as “double calcAngleBendTerms (int begin, int finished)” in our code. This function takes the two integer variables “start” and “end” as input and calculates results as variable type double. Another function “calcDHAngleterms (int begin, int finished)” is mentioned as “double type.” This function takes the two integer variables “start” and “end” as input and calculates results as a variable type double. The required parameters and constants are obtained from a file named “amber parameters.” It uses the parameters mentioned in Table 3 from the parameters required to calculate the dihedral angle terms and half nonbonded terms as mentioned in Table 3.

The function “double calcHalfNBterms (int begin, int finished)” takes the two integer variables “start” and “end” as input and calculates results as a variable type double. The function named as “double calcNBterms (int begin, int finished)” takes the two integer variables “begin” and “finished” as input and calculates results as a variable type double. Another function that is mentioned as “double calcHBterms (int begin, int finished)” in our code takes the two integer variables “begin” and “finished” as input and calculates results

TABLE 3: Parameters required to calculate dihedral angle terms and half nonbonded terms.

Sr. No.	Parameter	Description
(1)	“dihedralAngleTerms [ix7]”	All the four values of dihedral angle terms according to functional form
(2)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array
(3)	“phiDeg”	Bend of angle represented in degree keeping in view radian to degree relationship
(4)	“app”	Intermediate variable to calculate phiDeg
(5)	“energy”	The partial value of energy due to dihedral angle terms is calculated and stored in it
(6)	“coulombFactor”	It is calculated using permittivity and permittivity scale and a constant value 116
(7)	“halfNonBondedTerms [ix4]”	All the four values of half nonbonded terms according to functional form
(8)	“charge <i>i</i> ”	Value of charge on atom <i>i</i>
(9)	“charge <i>j</i> ”	Value of charge on atom <i>j</i>
(10)	“coordinates [atomix3]”	X, Y, Z coordinates of atom stored in array

TABLE 4: Names and IPs of experimental machines.

Sr. No.	Machine	Hardware distribution	Name	IP
(01)	VM1	2 GB RAM, 2 processor, 40 GB HD	Server	192.168.2.50
(02)	VM2	1 GB RAM, 1 processor, 40 GB HD	Client 1	192.168.2.51
(03)	VM3	1 GB RAM, 1 processor, 40 GB HD	Client 2	192.168.2.52
(04)	VM4	1 GB RAM, 1 processor, 40 GB HD	Client 3	192.168.2.53
(05)	VM5	1 GB RAM, 1 processor, 40 GB HD	Client 4	192.168.2.54

as a variable type double. Both functions required parameters and constants obtained from a file named “zamberparameters.” This program can run as multiple processes on different cores of coprocessor and processor.

5. Experimental Equipment and Setup

Our experimental setup mainly up consists of the following hardware and software: the 2x laptops having Core i5 and Core i7 with VMware Workstation software and 1x WiFi network connection as shown in Figure 4. All these systems have Linux OPEN SUSE as the operating system on machines with NetBeans Development toolkit for the AMBER calculations. Experimental Setup details are given in Table 6: In the proposed solution, there are two laptops Core i5 and Core i7. There are five virtual machines (VMs) on two laptops and details are shown in Table 4. The hardware details of the machines are given in Table 6. The hardware and associated software are mentioned for the user understanding.

The server is the central machine that distributes multiple tasks to various clients through MPI.

Open MPI is the core resource to distribute multiple tasks to various clients attached to the server of the virtual network. All machines have their SSH keys shared among one another to achieve passwordless connection to each machine for calculations. VMware assigns hardware resources to machines. NetBeans is installed on server for program compilation and execution. The screen of the server distributed computational process over the computing nodes as shown in Figure 5. Linux

Open SUSE OS is installed on each machine as a hypervisor. All machines are on the same network, that is, 192.168.2.0/42. VMs are connected through a virtual switch; in response, all clients send their tasks results to the main server. The server then calculates the Grande total result received from all clients. Computation node with completion time, CPU, PID, and memory is shown in Figure 5.

6. Results and Discussion of HPCC

High-performance cloud computing is an advanced technique that relies on virtualization and parallelism. To simulate this technique, multiple VMs are created in the simulated cloud. Resources are assigned to these VMs. One of the machines is the core machine named as a server. In Table 5, the comparison is being made among various solutions for molecular dynamics calculations. The comparison is carried out by HPC with Biomer. Biomer is software that runs on the single machine and lacks parallel mechanism like MPI. Hybrid parallel computing is a hybrid parallel technique that uses multiple hardware processing cores along Intel MPI to reduce the time for molecular dynamic tasks calculation. It splits and distributes tasks to other submachines named as clients. Clients in response send their result to the server. Table 5 shows that HPCC has much lower time as compared to the previous method.

The experiment consists of multiple numbers of processes like 1, 2, 12, 57, and 58. These numbers have been taken from HPC for apple to apple comparison. It also shows that, as none

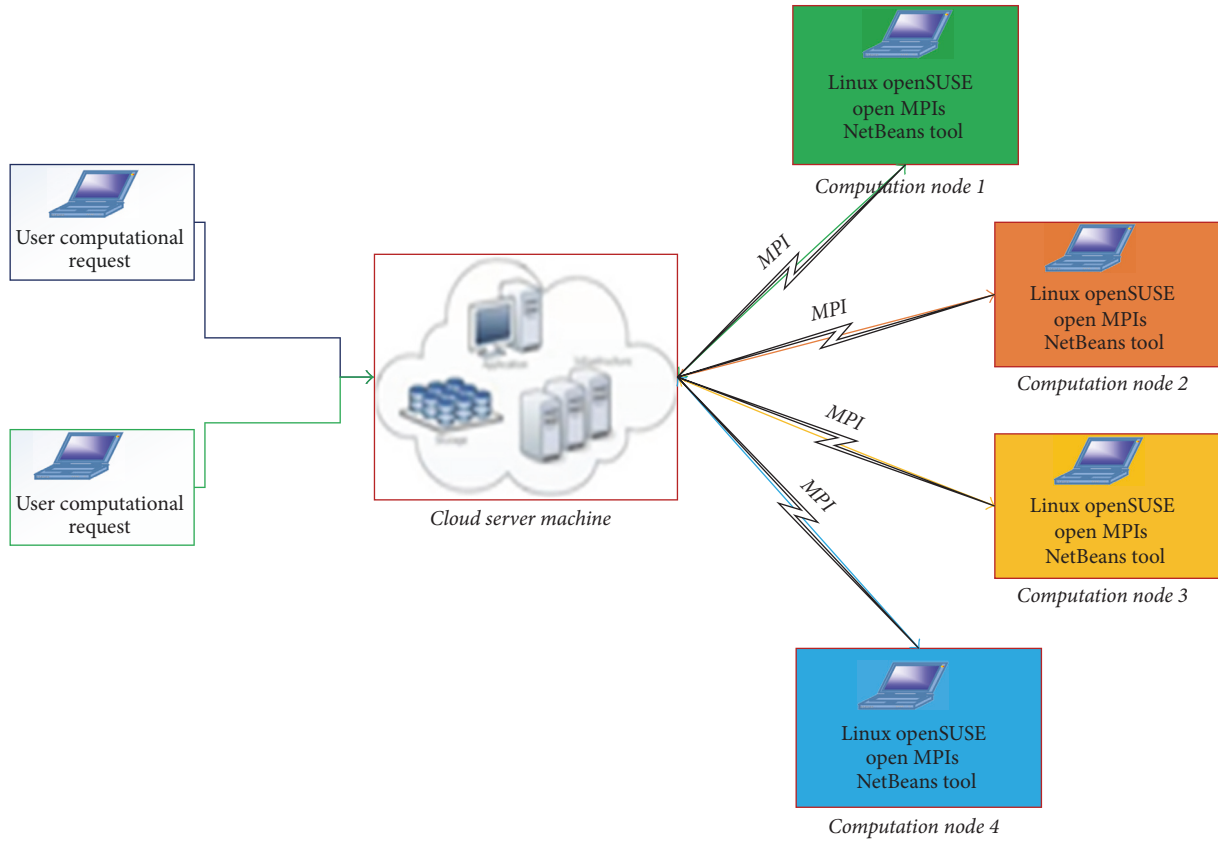


FIGURE 4: High-performance cloud computing simulation model.

TABLE 5: Comparison of processing time required for calculation of energy-based tasks.

No. of processes	HPC (second)	HPCC (second)	Biomer CPU (second)
1	N/A	N/A	5073
2	676	153	N/A
12	78	25	N/A
57	23	7	N/A
58	23	5	N/A

TABLE 6: Hardware and software comparison of HPC and HPCC.

Sr. No.	Attributes	Hybrid parallel computing (HPC)	High-performance cloud computing (HPCC)
(1)	Name	Intel Xeon Phi™	2x Dell laptop
(2)	Processor	Intel3120A	Core i7, i5
(3)	RAM	6 GB Ram	8 GB
(4)	Clock speed	1.00 GHz	2.7 GHz~3.5 GHz
(5)	Cores	57 Cores	No extra hardware
(6)	Card/VM	2x 2.1 GHz CPU, 2 GB RAM	4x VMs
(7)	VM specific	No VM	3x (1 processor, 1 GB RAM) 1x (1 processor, 1 GB)
(8)	Power supply	Additional required	No additional
(9)	Watt	400 for Phi TM Card	0
(10)	OS	Cent OS	Windows 7 & 10
(11)	Kernel type	Linux 2.6.32-431	SUSE Linux
(12)	Libraries	IMPI	MPI
(13)	Version	Intel Cluster Studio XE 2013	Server machine has NetBeans
(14)	Tested	Simple, compiled using IMPI, running on the coprocessor	Simple, compiled using MPI, Intel CPU on SUSE Linux


```

top - 02:09:49 up 9 min, 3 users, load average: 0.16, 0.13, 0.07
Tasks: 134 total, 1 running, 133 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 0.0 sy, 0.0 ni, 99.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 4054156 total, 1050476 used, 3003680 free, 51424 buffers
KiB Swap: 2103292 total, 0 used, 2103292 free, 381468 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     TIME+  COMMAND
 1840 root        20   0   502m  43m  30m  S  1.0  1.1   0:03.97  konsole
2119 root        20   0   992m  1500 1020  R  0.7  0.0   0:00.06  top
   60 root        20   0     0     0     0  S  0.3  0.0   0:00.76  kworker/0:2
  980 root        20   0   400m  230m  36m  S  0.3  5.8   0:16.79  X
    1 root        20   0 45664  4420 2144  S  0.0  0.1   0:02.36  systemd
    2 root        20   0     0     0     0  S  0.0  0.0   0:00.01  kthreadd
    3 root        20   0     0     0     0  S  0.0  0.0   0:00.09  ksoftirqd/0
    5 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kworker/0:0H
    7 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kworker/u:0H
    8 root        rt    0     0     0     0  S  0.0  0.0   0:00.74  migration/0
    9 root        -2   0     0     0     0  S  0.0  0.0   0:00.33  rcuc/0
   10 root        -2   0     0     0     0  S  0.0  0.0   0:00.00  rcub/0
   11 root        -2   0     0     0     0  S  0.0  0.0   0:00.00  rcub/1
   12 root        20   0     0     0     0  S  0.0  0.0   0:01.04  rcu_preempt
   13 root        20   0     0     0     0  S  0.0  0.0   0:00.00  rcu_bh
   14 root        20   0     0     0     0  S  0.0  0.0   0:00.00  rcu_sched
   15 root        rt    0     0     0     0  S  0.0  0.0   0:00.00  watchdog/0
   16 root        rt    0     0     0     0  S  0.0  0.0   0:00.00  watchdog/1
   17 root        -2   0     0     0     0  S  0.0  0.0   0:00.29  rcuc/1
   18 root        20   0     0     0     0  S  0.0  0.0   0:00.10  ksoftirqd/1
   19 root        rt    0     0     0     0  S  0.0  0.0   0:00.75  migration/1
   20 root        20   0     0     0     0  S  0.0  0.0   0:00.00  kworker/1:0
   21 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kworker/1:0H
   22 root        rt    0     0     0     0  S  0.0  0.0   0:00.00  watchdog/2
   23 root        -2   0     0     0     0  S  0.0  0.0   0:00.43  rcuc/2
   24 root        20   0     0     0     0  S  0.0  0.0   0:00.06  ksoftirqd/2
   25 root        rt    0     0     0     0  S  0.0  0.0   0:00.79  migration/2
   27 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kworker/2:0H
   28 root        rt    0     0     0     0  S  0.0  0.0   0:00.00  watchdog/3
   29 root        -2   0     0     0     0  S  0.0  0.0   0:00.47  rcuc/3
   30 root        20   0     0     0     0  S  0.0  0.0   0:00.06  ksoftirqd/3
   31 root        rt    0     0     0     0  S  0.0  0.0   0:00.72  migration/3
   33 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kworker/3:0H
   34 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  cpuset
   35 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  khelper
   36 root        20   0     0     0     0  S  0.0  0.0   0:00.01  kdevtmpfs
   37 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  netns
   38 root        20   0     0     0     0  S  0.0  0.0   0:00.00  bdi-default
   39 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kintegrityd
   40 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  kblockd
   41 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  ata_sff
   42 root        20   0     0     0     0  S  0.0  0.0   0:00.02  khubd
   43 root         0 -20     0     0     0  S  0.0  0.0   0:00.00  md
   46 root        20   0     0     0     0  S  0.0  0.0   0:00.04  kworker/2:1
   47 root        20   0     0     0     0  S  0.0  0.0   0:00.09  kworker/3:1
   48 root        20   0     0     0     0  S  0.0  0.0   0:00.00  khungtaskd
   49 root        20   0     0     0     0  S  0.0  0.0   0:00.00  kswapd0
   50 root        25   5     0     0     0  S  0.0  0.0   0:00.00  ksm

```

FIGURE 5: Computation node with completion time, CPU, PID, and memory.

of processes increases, time for calculating tasks reduces just because of parallel techniques as shown in Figure 6.

In Table 6, there is the comparison of hardware resources used in HPCC and HPC solution. Hybrid parallel computing (HPC) is much resource consuming as compared to high-performance cloud computing which is based on virtualization. HPC hardware is much costly as compared to HPCC.

Table 6 shows that HPCC mostly relies on software as compared to HPC which relies on costly hardware. The processing time has been obtained by executing the same energy calculation, and complexity of Biomer Java-based software is 5073 sec which is reduced to hundreds of seconds by using cloud computing in combination with parallel computing. It is because Biomer software is Java-based and

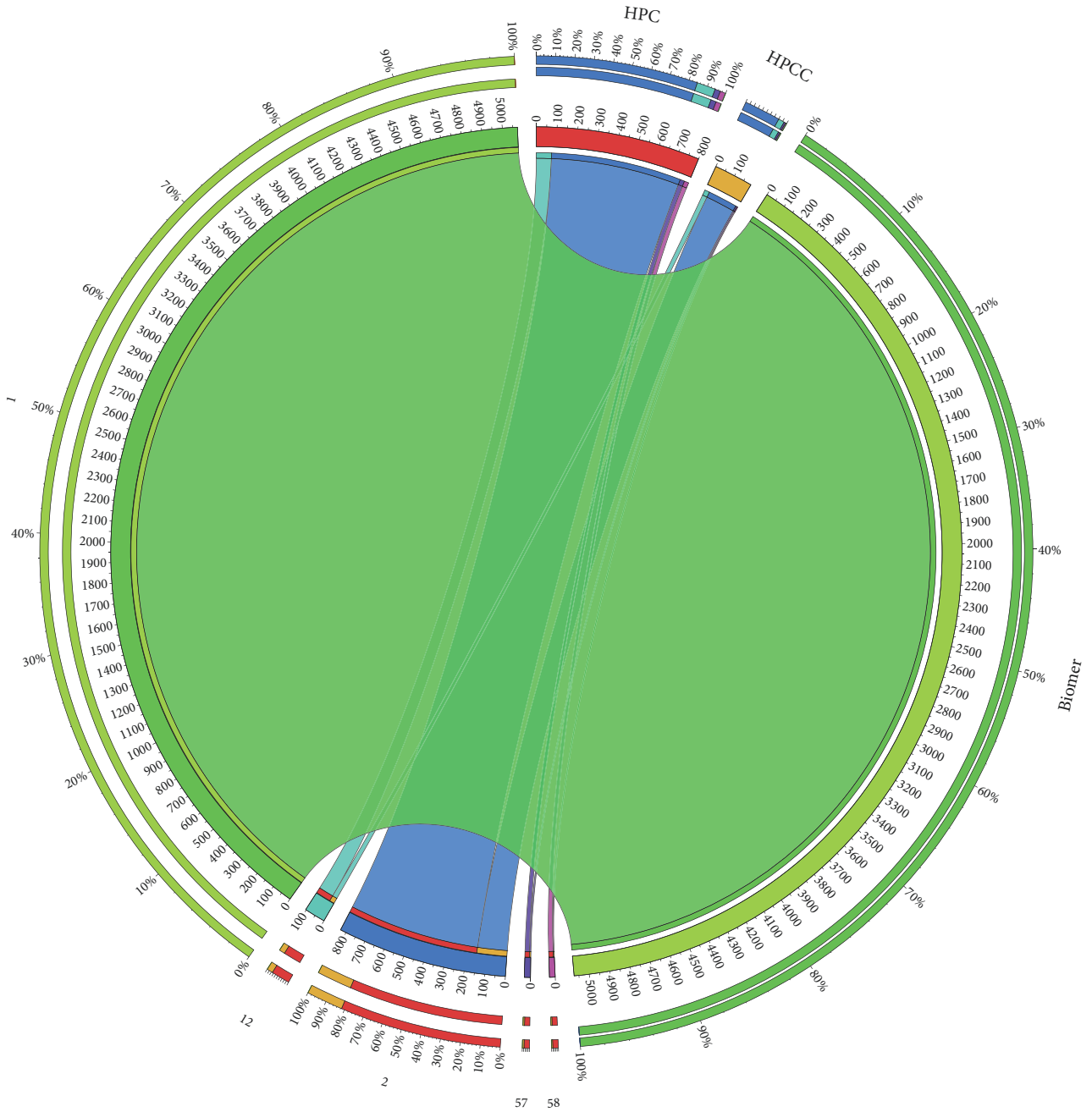


FIGURE 6: Comparison of HPC and HPCC time required for calculation of energy.

runs as a single process without using the mechanism of the parallelism. It has further to reduce calculation time by increasing number of process and machines resources. Figure 7 shows the comparative result of HPCC and HPC techniques.

It would get more efficient result by increasing number of the process to be run using parallelism and using high-end desktop machines and optimized network media. In another comparison depicted in Figure 8, when number of processes has been increased to 100% of previous value, calculation time is reduced to 50% of previous value. Values for number of processes are 10, 20, 40, and 80, processing time is reduced

to one-half of previous value, and efficiency is increased to 100% as shown by the graph in Figure 8.

If X is the number of processes to be selected for calculation and y is the calculation time for number of processes of the previous value, then X will be calculated through the equation $X = (1/2)Y$. Figure 8 shows that when number of process increased up to double of the previous value, its calculation time is reduced up to 1/2 of the previous value of calculation time. The result is verified in Table 5. In the table, there is the comparison of software resources used in HPCC and HPC solution. HPCC uses VMware Workstation as virtualization and cloud computing tool. Physical machine

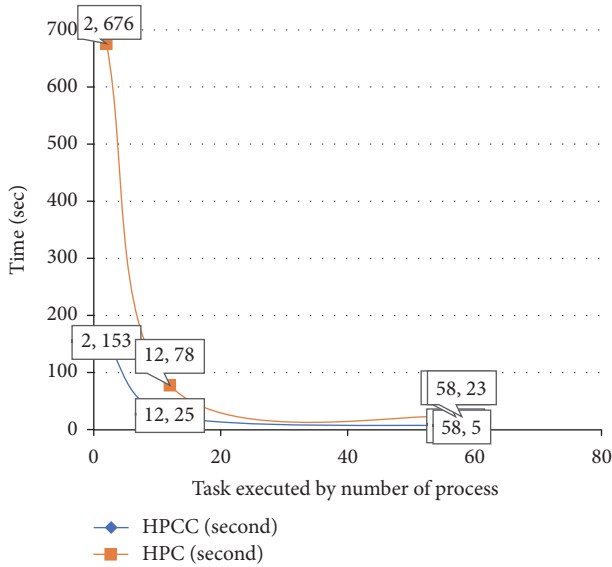


FIGURE 7: Comparison chart of HPCC and HPC processes.

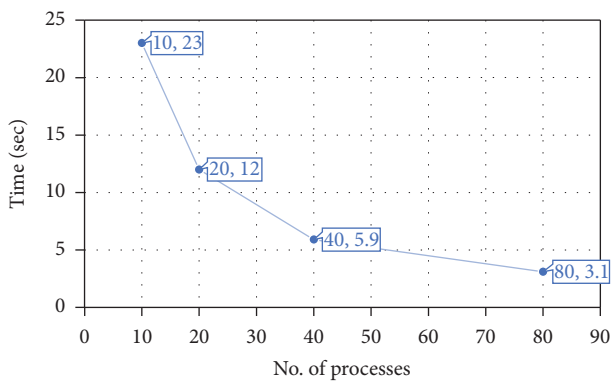


FIGURE 8: Comparison of HPCC in terms of number of processes versus execution time.

has Windows OS and VMs have open SUSE Linux flavour. Open MPI is used for the parallel tool. HPCC mostly relies on software as compared to HPC which relies on costly hardware.

7. Conclusion and Future Work

It is concluded from this research that big data in the field of molecular dynamics is a high computational task and requires large computational resources and advanced techniques to meet the requirements. Data can be extracted, processed, and analyzed in AMBER required cloud computing techniques which are suitable for the parallel execution of tasks. It provides flexibility and scalability of resources for calculation of heavy computational tasks. It is a homogeneous mobile cloud platform consisting of laptops and virtual machines as processing nodes in combination with dynamic parallelism. Multiple numbers of processes have been chosen at run time to distribute the task among the computing nodes. Message Passing Interface is used to attain task-based and

data-based parallelism. It used shared hardware resources for VMs for achieving better results. In future high-end mobile and smartphone using fast and efficient networking media can be used for better results. MPI can further be tuned and programmed for dynamic parallelism. MPI can also be integrated with another Linux flavours as well. Moreover, it is also implementable for the big data for the enterprises and analytics.

Conflicts of Interest

All the authors confirm that there are no conflicts of interest regarding the publication of this research article.

Acknowledgments

This research is supported by the Department of Computer Science, University of Engineering and Technology, Taxila.

References

- [1] Y. Saleem, M. Munwar Iqbal, M. Amjad et al., “High security and privacy in cloud computing paradigm through single sign on,” *Life Science Journal*, vol. 9, no. 4, pp. 627–636, 2012.
- [2] M. H. R. Pereira, C. L. de Souza, F. L. C. Pádua, G. D. Silva, G. T. de Assis, and A. C. M. Pereira, “SAPTE: A multimedia information system to support the discourse analysis and information retrieval of television programs,” *Multimedia Tools and Applications*, vol. 74, no. 23, pp. 10923–10963, 2015.
- [3] C. Yang, Y. Xu, and D. Nebert, “Redefining the possibility of digital Earth and geosciences with spatial cloud computing,” *International Journal of Digital Earth*, vol. 6, no. 4, pp. 297–312, 2013.
- [4] H. Chen, R. H. L. Chiang, and V. C. Storey, “Business intelligence and analytics: from big data to big impact,” *MIS Quarterly: Management Information Systems*, vol. 36, no. 4, pp. 1165–1188, 2012.
- [5] M. M. Iqbal, Y. Saleem, K. Naseer, and M. Kim, “Multimedia based student-teacher smart interaction framework using multi-agents in eLearning,” *Multimedia Tools and Applications*, pp. 1–24, 2017.
- [6] B. Joó, D. D. Kalamkar, K. Vaidyanathan et al., “Lattice QCD on Intel® Xeon Phi™ coprocessors,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7905, pp. 40–54, 2013.
- [7] A. Gupta, A. Mehrotra, and P. M. Khan, “Challenges of cloud computing & big data analytics,” in *Proceedings of the 2nd International Conference on Computing for Sustainable Global Development, INDIACom 2015*, pp. 1112–1115, ind, March 2015.
- [8] P. Malik, “Governing Big Data: Principles and practices,” *IBM Journal of Research and Development*, vol. 57, no. 3/4, pp. 1:1–1:13, 2013.
- [9] C. J. Newburn, S. Dmitriev, R. Narayanaswamy et al., “Offload compiler runtime for the intel® Xeon Phi coprocessor,” in *Proceedings of the 2013 IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, IPDPSW 2013*, pp. 1213–1224, usa, May 2013.
- [10] S. Ahmad, S. Xu, B. Li, and S. Ahmad, “Research and implementation of Hybrid Parallel Computing for force field calculation,”

- in *Electronics, Information Technology and Intellectualization: Proceedings of the International Conference EITI 2014*, Shenzhen, China, 2015.
- [11] J. C. A. Boeyens, "A molecular-structure hypothesis," *International Journal of Molecular Sciences*, vol. 11, no. 11, pp. 4267–4284, 2010.
- [12] W. D. Cornell, P. Cieplak, C. I. Bayly et al., "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *Journal of the American Chemical Society*, vol. 117, no. 19, pp. 5179–5197, 1995.
- [13] G. M. Tihfon, S. Park, J. Kim, and Y.-M. Kim, "An efficient multi-task PaaS cloud infrastructure based on docker and AWS ECS for application deployment," *Cluster Computing*, vol. 19, no. 3, pp. 1585–1597, 2016.
- [14] G. D. M. Seabra, R. C. Walker, M. Elstner, D. A. Case, and A. E. Roitberg, "Implementation of the SCC-DFTB method for hybrid QM/MM simulations within the Amber molecular dynamics package," *The Journal of Physical Chemistry A*, vol. 111, no. 26, pp. 5655–5664, 2007.
- [15] M. Allen and D. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, UK, 1987.
- [16] V. Gapsys, D. Seeliger, and B. L. De Groot, "New soft-core potential function for molecular dynamics based alchemical free energy calculations," *Journal of Chemical Theory and Computation*, vol. 8, no. 7, pp. 2373–2382, 2012.
- [17] S. Mittal, "A survey of techniques for architecting and managing asymmetric multicore processors," *ACM Computing Surveys*, vol. 48, no. 3, article no. 45, 2016.
- [18] B. D. Veerasamy and G. Nasira, "Exploring the Contrast on GPGPU Computing through CUDA and OpenCL," *i-manager's Journal on Software Engineering*, vol. 9, no. 1, pp. 1–8, 2014.
- [19] H. Zheng, K. M. Langner, G. P. Shields et al., "Data mining of iron(II) and iron(III) bond-valence parameters, and their relevance for macromolecular crystallography," *Acta Crystallographica Section D: Structural Biology*, vol. 73, no. 4, pp. 316–325, 2017.
- [20] L. F. L. Oliveira, N. Tarrat, J. Cuny et al., "Benchmarking Density Functional Based Tight-Binding for Silver and Gold Materials: From Small Clusters to Bulk," *The Journal of Physical Chemistry A*, vol. 120, no. 42, pp. 8469–8483, 2016.
- [21] A. Verma, O. Uzun, Y. Hu et al., "Surface-structure-regulated cell-membrane penetration by monolayer-protected nanoparticles," *Nature Materials*, vol. 7, no. 7, pp. 588–595, 2008.
- [22] Molecular_dynamics, ed: Wikipedia, 2013.
- [23] G. H. Fredrickson, V. Ganesan, and F. Drolet, "Field-theoretic computer simulation methods for polymers and complex fluids," *Macromolecules*, vol. 35, no. 1, pp. 16–39, 2002.
- [24] C. Delgado, S. Y. Stevens, N. Shin, and J. Krajcik, "A middle school instructional unit for size and scale contextualized in nanotechnology," *Nanotechnology Reviews*, vol. 4, no. 1, pp. 51–69, 2015.
- [25] Y. Duan, C. Wu, S. Chowdhury et al., "A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations," *Journal of Computational Chemistry*, vol. 24, no. 16, pp. 1999–2012, 2003.
- [26] F. Magoulès, M. Parsons, and L. Smith, *Innovative Algorithms for Extreme Scale Computing*, SAGE Publications Sage, London, England, 2015.
- [27] J. Vetter, A. Almgren, P. DeMar, K. Riley, K. Antypas, D. Bard et al., *Advanced Scientific Computing Research Exascale Requirements Review. An Office of Science review sponsored by Advanced Scientific Computing Research, September 27–29, 2016, Rockville, Maryland, Argonne National Lab. (ANL), Argonne, IL, USA, 2016, Argonne Leadership Computing Facility 2017.*
- [28] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, and C. Simmerling, "Comparison of multiple amber force fields and development of improved protein backbone parameters," *Proteins: Structure, Function, and Bioinformatics*, vol. 65, no. 3, pp. 712–725, 2006.
- [29] J.-s. Shyr, *Adaptive scheduling of function cells in dynamic reconfigurable logic*, Google Patents, 2005.
- [30] R. A. Bridges, N. Imam, and T. M. Mintz, "Understanding GPU power: A survey of profiling, modeling, and simulation methods," *ACM Computing Surveys*, vol. 49, no. 3, article no. 41, 2016.
- [31] D. Barr, A. Basden, N. Dipper, and N. Schwartz, "Reducing adaptive optics latency using Xeon Phi many-core processors," *Monthly Notices of the Royal Astronomical Society*, vol. 453, no. 3, pp. 3222–3233, 2015.
- [32] E. Wang, Q. Zhang, B. Shen et al., *High-Performance Computing on the Intel Xeon Phi*, Springer International Publishing, 2014.
- [33] J. Kim, K. Jang, K. Lee, S. Ma, J. Shim, and S. Moon, "NBA (network balancing act): A high-performance packet processing framework for heterogeneous processors," in *Proceedings of the 10th European Conference on Computer Systems, EuroSys 2015, France, April 2015*.
- [34] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig, "Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization," *Intel Technology Journal*, vol. 10, 2006.
- [35] P. Raj, A. Raman, D. Nagaraj, and S. Duggirala, "High-Performance Grids and Clusters," in *High-Performance Big-Data Analytics*, pp. 275–315, 2015.



Hindawi

Submit your manuscripts at
www.hindawi.com

