Also presented as a Key Note Pape in the
1st International Conference on Information and Communication Technology, ICICT 2005
Karachi, Pakistan, August 2005.

Emerging Era of Cooperative Empowerment: Grid, Peer-to-Peer, and Community Computing

Javed I. Khan

Media Communications and Networking Research Lab
Department of Computer Science
Kent State University, Kent OH 44242
javed@kent.edu

Abstract: *In this paper we discuss an emerging trend in high performance computing- the social or community computing. The emergence of screensaver computing, grid computing, peer-to-peer systems, and their demonstrated ability to garner computing power as potent as the supercomputers seems to be auspicious. In this paper we discuss the new phenomenon of community computing- including their potential applications in solving a new set of grand challenge problems. We also discuss the previously uncharted technological challenges architects of these new paradigms are addressing.*

1 Introduction

The 25th list of Top500.org released in June 2005 has just announced that the topper IBM BlueGene/L has again topped itself. Sitting at Lawrence Livermore National Lab, this mammoth's current configuration is capable of computing at 136 TFLOPS. Just six months back it has replaced past six list's topper Japan's Supercomputer NEC Earth Simulator- which itself has remarkable computing capacity of 36 TLOPS. It seems to be the news. However, on second thought, is that the news anymore?

Indeed, the *Super Computer* page on wikipedia.org, (version, 8:28pm 29th July, 2005) contains another piece of statistics- SETI@home- a distributed computing project has reached average processing power of 72.53 TFLOPS. On May 16, 2005, another distributed computing project FOLDING@home reported a processing power of 195 TFLOPS. On February 2, 2005, the same system recorded 207 TFLOPS. What is remarkable about these new projects is that all these are using simple inexpensive ordinary computers like our desktops and laptops connected via regular Internet, and using only the idle cycles on them at less than $5 compared to the thousands of dollars cost per GFLOP [1] on the Super Computers!

The real news is that a new era has emerged where intelligent management of simple resources can open up the door of tera computing to a much wider and poorer public. What is also remarkable is the way how it is being done- by garnering the goodwill of hundreds of thousands of people. While these new distributed systems (grid.org, distributed.net, seti@home, etc.) are now routinely superseding supercomputer heavyweights- these also are exploiting new technological horizons- social engineering previously uncharted territories of computing architecture [4]. We are probably seeing emergence of a new era of community computing- where computing will be performed for the people and by the people- in a little more emphasized senses of the prepositions. In this paper we will visit the major strands of the emerging community computing paradigms and discuss the new technical issues surrounding this latest wave of computing. As we will see the very new page wikipedia.org it-self is a harbinger of this new wave.

2 Paradigm Shift: Supercomputing to Social Computing

It will be interesting to trace the evolution of social computing beginning from the parallel computing paradigms. Current **super computers** are made of multiple hardware units running under one operating system (OS) environment and the entire system is run under single ownership. All parallel units are mostly replica in architecture. The OS does not have to worry about the architectural variations- rather the entire system is designed to takes advantage of the uniformity. Computing resources- memory, storage, CPU all are connected by dedicated hardware interconnections. **Distributed computing system** (DCS) also emerged in the same period. These also have multiple computing units. Each however, has its own complete resource set and own operating system. All computing units are generally inside same administrative domain, and are guided under one management policy. A key difference is however, that the resources are connected by a shared LAN. **Cluster computing** is a hybrid between the two. Clusters too have multiple computing units each with its own hardware and own operating system. The computing system is uniform and identical in a stronger sense than distributed computing systems. All computing units are in the same administrative domain, and are connected by high performance dedicated LAN owned by the same administrative domain as the CPU and other systems. All are managed by one resource sharing principle. Several clusters today rank within top 500 supercomputers.

Parallel computing seems to have grappled well with the multiplicity of hardware CPUs, storage units, network infrastructure, memory or computing device, etc. Soon after, however, we are seeing new paradigms where the notion of multiplicity no longer emphasizes the multiplicity of physical resources. Rather these emphasize exploitation of resources under multiple ownership domains with greater heterogeneity in the usage, sharing policies, and principles associated with ownership right. This is the tipping point towards the era of social computing. Since the notion of architectural multiplicity has taken a back seat- community computing system today may employ parallel hardware at the inner core but its super structure can be architecturally centrist-such as the Wikipedia or obsessively distributed like Nutella.

The most spectacular of this new genre undoubtedly is the **screensaver computing**. The best known among them is the SETI@home. It is a simple signal analysis program that aims at finding extra terrestrials. Since its appearance in the May of 1999, it has spread like wildfire perhaps due to its sheer romantic appeal. The number of participants today exceeds 5 million. The system originated from the motivation of utilizing idle computing cycles of many simple computers. The entire program was packed as a screensaver. This indeed was a brilliant social engineering- which essentially told everything about the program to an average PC user. Though SETI is the best known screensaver computing to-date, however, this is not the first one, nor the last. Condor in later 80's begun scavenging cycles from idle UNIX systems. Distributed.net was another pioneer in the area which distributed codes for the sheer joy of breaking apparently unbreakable cryptographic encodings. Today, tens of similar initiatives are in progress (see distributed.net, grid.org) solving various scientific problems ranging from code breaker, determining the digit's of pi, cancer drug discovery, proteome folding, etc. Each has tens of thousands of participants and are enjoying near supercomputing power at the cost of few servers.

Architecturally, in screensaver computing model, the participating computers have separate owners. Each is completely independent. The resources are connected via the commodity Internet requiring very basic connectivity. Each computing system contributes its idle cycles to the assigned task. A remote application server sends the tasks and collects the results. But, when to run the task is decided by the host computers. Computers do not interact among each other. Individual user donates the computer time and derives no tangible benefit from the computing it helps.

While the screensaver computing systems were evolving depending on the pure goodwill of the good hearted PC owners, at the same time, another very interesting model of social computing erupted- **peer-to-peer (P2P) systems** lead by Napster and likes. However, not from quite such selfless motivation- some even calls this the intention to 'steal and enjoy'! The participation here derives quite tangible benefit.

Architecturally, P2P systems are organized as massively distributed system which enables sharing of digital files among a community of users where each user commits small storage space and computing power over the Internet. The individual systems are fully independent. The community uses a uniform resource discovery, and exchange protocol to search, find and exchange the files. The storage of primary material and their directories are all distributed. The participants donate the resource to gain free access to these files. P2P computing originated as a grass-root movement out of tangible expectation of benefit by the participants.

The P2P computing has a very special significance in social computing. It is the first to seriously challenge any social order. Also, serious social engineering considerations have guided its core architecture. Its design objective is to seek how a digital service can be implemented with minimum centralized components. Any universal service tends to require some centralized components- whether it is a complete application master server or a master directory server or just user authentication database. P2P systems minimize any such centrality from architecture, thus it tends to make all individual participants potentially capable of taking over the task of other thus introducing the concept of 'equal peer'. This is to reduce the vulnerability of the overall system not only from individuals but also against threats from social forces of an organized society that tends to attack a target. P2P removes targets from design. It is estimated that there are at least three P2P systems (BitTorrent, eDonkey, Gnutella, etc.) each has more than two million followers currently (Ref: slyck.com). P2P now poses a challenge to the business models of content ownership. Another particular interesting genre in P2P crowd is the censorship-resistant systems such as Entropy or Freenet. It works by pooling the contributed storage space of the members and distributing the documents in those storage spaces without any user controllability. Also, all files and data are transferred in an encrypted form. Thus it is virtually impossible to track the originating site of any particular document. Its architecture thus ensures complete anonymity of the author and preserves his/her right to publish without fear of retribution from the forces of organized society.

The latest name of community computing is **Grid**. The success of screensaver computing in gathering CPU cycles from millions of ordinary computers and its ability to exceed or equal that of the largest supercomputers has also inspired major systems vendors (United Devices, IBM, Sun, HP, Parabon Computation, Data Synapse, Entropia are some of the leaders) and the supercomputing community to pool even high value resources together. This commercial model of Grid [2] computing now generalizes the concept of screensaver computing and is aimed at providing a platform for solving large scientific problems in a more complex community setup which otherwise was only possible with expensive supercomputers.

The concept of resource now includes all forms of computing elements- supercomputing power, massive storage, databases, instruments, on-line tools, etc. The high value resources can also be connected via high performance communication infrastructure. The concept of ownership domain also has been generalized to accommodate institutions and corporations which can contribute cluster of resources with multiple/hierarchical administrative parties. Individual parties can participate with all or part of their resources to a Grid with complex accounting and sharing policies. The motivation for contributing resource in this pool is also complex. The participants donate the resource based on institutional or corporate interest for strategic cooperation however, with guarded trust. A set of protocols and middleware are being developed (mostly under open source alliance Globus.org) to help this complex business of resource discovery, allocation and resolving security and access policies, and to make life easier for end-users who can view a grid as a gigantic virtual computer. Current grids are solving scientific problems such protein folding, financial modeling, earthquake simulation, climate/weather modeling, etc. There are also several projects solving math problems such as computing large prime numbers, or digits of pi. Currently groups such as grid.org, distributed.net have hundreds of thousands of contributors spanning across 200+ countries. The vision of Grid is now vigorously expanding. It has inspired a great many ventures- such as Semantic Grid, Sensor Grid, TeraGrid, DataGrid, Access Grid, etc. Grid now symbolizes ways to pool together computing resources of various organizations, and thus potentially can solve big-science applications which no single super computer can solve.

So what is next? It seems one or two more generalizations are waiting on the wings. One of the imminent new mode computing is socially inclusive **interactive community computing**. In screensaver computing a group or individual contribute the computing power for e-science. There is however no direct connection between the contributors and the beneficiaries. Its social implication is that what gets computed is chosen by a centralized entity. It will soon be important that a contributing community also sees immediate benefits. Architecturally, it means that contributors will soon interact with their screensavers! An example of such application is a digital disease management system. Indeed one early system is being made now!

It is called Diabetes PHD system [6]. The computing helps a person with Diabetes to better manage his or her lifestyle. When a person provides information about him or herself, it is fed into an extensive mathematical model called Archimedes. This model connects an extensive detail array of anatomy, physiology and pathology models of a person including models of virtual organs and tissues such as heart, pancreas, kidneys, eyes, blood. Archimedes then creates a simulated person who has the same characteristics such as sex, age, race/ethnicity, height, weight, blood pressure, same laboratory test results such as glucose, cholesterol, the same past medical history, family history, symptoms, complications, and treatments etc. Archimedes then takes this simulated version of the person and creates a thousand “identical looking” people. It then calculates a future for those people – what is going to happen to them over the next 30 years, and tells the patients the types of complicacy he/she may face, and what type of change in lifestyle, diet might improve the life. In live trials, Archimedes [7] has been found to have magical accuracy! Archimedes however requires humongous computing power. It solves a massive array of differential equations, regression models, decision trees, flow charts, and accounting models. In an interesting community computing initiative diabetes patients and their friends can now contribute the idle CPU times via a grid system now being organized to support this massive computation. While it cannot say that a person will have a heart attack on any particular day, but it can say with quite surprising accuracy for example, that there is an 80 percent chance the person will have a heart attack sometime in the next 5 years. Archimedes can also calculate what if scenarios if the user decides to change their lifestyle. Overall it can be a very effective self-help tool in improving the longevity and quality of life for diabetes patients.

The potential of such a project however, extends much beyond. It can be further coupled within the fabric of the community. The Diabetes MD requires clinical trial data from diabetes patients and careful formulation and perfection of all the associated models. In a community computing system- then entire process can be potentially automated. The data fed by patients can again be reused to perfect the underlying science models. It can also be used to build such systems for other diseases- perhaps by incorporating learning algorithms.

Finally, I will present another level of generalization of social computing- **man/machine symbiotic community computing**. In all the previous examples the resource meant computing resources. But how about harnessing collective human intellect? There are still many tasks which are solved by human in much better ways then by computers. Is it possible to harness the collective intellectual ability of a community in solving large versions of few such extremely difficult problems via grid like ‘intellect extractor’ infrastructure? Let me try to define such a system. It will harness collective human computing power contributed by a community and will strike clever symbiosis between the participating human and the computer to solve an organized digital computing problem. If a group is involved in some form of 'computing' and/or the result is a 'digital resource' on which computation can be performed, the overall system can be justifiably called a computing system[1]. Licklider [8] identified elements of fine grained symbiosis between man/machine by studying the steps within the physical/mental endeavor of a scientist in what he called 'A Preliminary and Informal Time-and-Motion Analysis of Technical Thinking'. He observed the mini steps in thinking such as hypothesis formation, setting up experiments, plotting data, observations, correcting hypothesis, and observed 80% of time was in actuality spent 'to arrive at a position to think'- and hoped much of which be taken over by a future computer. In present day context this preparatory time is probably longer. Yet in network computing context this can be left more to computing than ever before. Also, in a community computing environment, more humans can intellectually participate- either with redundant or complementary human skills. But either case can enhance the quality of that other 20% of the thinking process- if just the proper 'intellect extractor' process is instituted.

Indeed, it seems we are moving towards such man/machine symbiotic community computing systems. A shining example is Wiki. *Wikipedia* is a community computing system- though not distributed, but enables a community to jointly perform a very complex organized task- which most computers possibly won't dare. Wikipedia.org has engineered one such 'intellect extractor' process though which the intellects of millions of people are procedurally harnessed to develop and evolve a gigantic digital encyclopedia. Here million participate via the Internet to contribute their knowledge to build this digital resource. The encyclopedia as a whole showcases how complementary expertise can increase the width of the task. The vigilant-stewardship of typical wiki pages by several volunteers showcases how redundant human expertise can improve precision and accuracy of the task. Though, at first glance it may seem that it lacks human computing- but indeed there is. The same if done by computers would require massive (yet potentially futile) computing. Also, once the digital product is there- all forms of semantic web applications are conceivable that can infer and compute on Wikipedia. Examples are Wikibook threads. Below are few additional examples of man/machine community computing project:

- Target Detection: Provide satellite imagery of a city and let computer system perform an initial pass. Then distribute the images to humans and let them perform the final detection. The same concept can be potentially extended to other detection and analysis scenarios (Sensor & Sensory based City Defense).
- Knowledge Excavation: Knowledge is information but in usable form. Though there are relatively better means for machine processing and machine inference of knowledge, but, there is no machine means for extraction and excavation of knowledge. Currently, only humans can extract new knowledge. Community computing can be used very successfully in knowledge extraction and subsequent knowledge engineering. Wikipedia is a good example of man machine mediated knowledge extraction application.
- Literature Translation/Aesthetics Processor: Machine translation is available only at limited technical quality only between limited languages. Literary translation that can retain the elegance and inner beauty and aesthetic quality of prose is beyond the realm of machine translation (consider the translation of Urdu 'sher'). A man/machine community computing scheme can electrify the e-literature and make linguistic beauty of various languages accessible to worldwide audience by harnessing human ability to process aesthetics.
- People's Journalism: Same is true for availability of multi-lingual news editing and gathering. News group appeared as one of the first applications of the Internet. However, currently there are several problems with the news-like elements on the Internet such as online diary, blog, travel recorder, public history, etc. However, it seems that more organized applications are poised to appear subject to sorting out some computer assisted social engineering.

3 Technical Challenges

The design of various Internet mediated process that can institute social computing is no longer a conventional networking or computing issue. It would require serious understanding in socio-psychological mechanics. However, before we discuss the elements of social engineering, I will briefly discuss some systems engineering issues that also must be addressed.

3.1 Systems Engineering

When a large number of entities contribute resources to a common pool- a set of new system level challenges appears. The system need to maintain a **dynamic directory service** via which scheduling middleware can discover, find, locate, and manage resources. Because of multiple ownership domains a second set of major issues emerges regarding **authentication and access control** policies. For example, a particular contributor may

[1] I will leave out discussion groups and chat rooms and consider them more as medium of communication than computing.

want to provide access only to a specific group of users, or specific type of applications. On the other hand, there might be policy restriction from the point of view of the application owner- where the application owner might want to restrict the computation within certain domain for legal or financial arrangements. Since, the applications may originate from third parties, therefore the **trust and security** problem also becomes much more complex. Since, the codes are no longer executed in the same authority domain it becomes problematic to ensure that a result returned is actually executed by the valid code. The connectivity topology, the timing of the connectivity, availability of processing power, local disk space, each becomes mostly an unpredictable dynamic quantity. Thus, the problems of **task scheduling, remote loading, execution, dynamic scheduling** becomes much more complex. The level of heterogeneity also increases in a multi-owner pool. Thus, another major issue becomes the code portability and protocol **interoperability**. In a communal environment, each contributing organization or person may use different operating system and middleware to manage their own part of the resource. Not only the policies may vary, even the language or the framework may also vary. A communal system must overcome the compatibility issues both at syntactic and semantic level across many platforms.

3.2 Social Engineering

However, the challenging new territory of social computing is the social engineering. This is an unfamiliar turf for computing sciences- even to social sciences. Any system which depends on voluntary participation by millions of humans must be built on sophisticated socio-psychological understanding of this virtual society. Though, there is much to learn from existing body of social science knowledge, but in reality the virtual communities did not simply exist before. Therefore much of the social engineering must happen as the system evolves. Procedural mechanisms have to be developed where groups of people can work jointly and participate as an overall man machine symbiotic computing system and finally achieve a joint computing objective. There are already some architectural trends. It seems that in each of the system architecture, the social engineering evolves around a fabric of observed **sociomic phenomena**. The system is then engineered for harmonic management of these. Management of each phenomenon is often based on one or more *sociometric measure(s).* It will be interesting to see few such measures and associates issues in few such systems.

- **Trust Phenomena:** One of the first sociometric measures to appear and to become an indispensable part of many communal systems is the rating and reputation system. The objective of the reputation system in e-business is to overcome the trust/distrust phenomenon that arises in online transactions specifically buying and selling activities. The first generation reputation systems however, suffer from various limitations. For example many of these are susceptible to group attack, where a group of users can artificially send wrong ratings targeting a particular seller or buyer. Another potential problem is mutual rating boosting, where two users can come to a collusion of increase each other ratings. It is possible to develop improved algorithmic techniques and rating mechanism which however can mitigate the potential of such attacks. Ratings are now associated with roles (buyer, seller, authors, etc.), products, web-sites. It has evolved into a complex feedback system, although we have very little formal understanding about it.

- **War of Opinions:** "Wikipedia" has observed several interesting phenomena. A particularly interesting one is the **Edit war**. Where a two or more users continuously engage in editing and reediting each other's version. It has been identified that there are quite a few forms of this war. The original version is a protracted war where two people simply do not like each others opinions or definitions. In another version it can be simply the "revert-war", where instead of editing- two sides engage in simply reverting back to a version from the history page that they like. An ongoing revert war can very fast make the history mechanism useless. However, there is also another type- which appears as a war but indeed is not a war at all. Here two users begin updating a page simultaneously. The one who saves later his/her version appears to be final and to the other it looks like an edit-war. This is now called "edit-conflict". Wikipedia is experimenting with several sociometric measures to assuage each. One such technique is 'three-revert-rule' which bars reversion of any single page more than three times within a period of 24 hours by one user. If for some reason such edit is necessary then a third person have to be found. For serious conflict with the content, Wikipedia also provides a "discussion page" with each page published. Algorithmic approaches such as slowing down the allowed edit counts, providing an easy merge tool, diff tool allows community to deal with edit wars.

- **Vandalism:** Vandalism is however a different problem. Occasionally, there will be attacks when documents are changed with a deliberate attempt to compromise the integrity of the encyclopedia. It has very little to do with content or opinion. Wikipedia has identified about 15 sub-types of vandalisms. Wikipedia has also engineered various means to deal with them. Apparently, an IBM study [5] has found that as a result most vandalized pages in Wikipedia is reverted back within 5 minutes by other users and thus most readers may not even notice the act of vandals.

- **Free Riders:** Some users act selfishly by only consuming resources without contributing. This phenomenon is known as free riding [3]. For file sharing system- it means who only downloads without uploading. Typically, a system's viability is threatened if too many users are free-riding. Therefore all community systems need to control it. All P2P systems are now instituting some form of sociometric measure that tracks user contribution and then tie it down with the quality-of-service that user receives. The actual definition of 'contribution' depends on the service scenario. For example bitTorrent uses *share-ratio*- which measures how long a user keeps his/her client active after a download- as BitTorrrent encourages a downloader also to act as a distributor.

From system point of view, it seems all community systems are conceptually vulnerable to certain basic sociomic phenomena. Thus, it might be possible to develop holistic approach to them. Social engineering is very important for the sustainability, stability, and smooth operation of the system. The first generation systems are currently managing them in ad-hoc manners. We have very little formal understanding about it. It seems that very soon social engineering will become major part of computing architecture design.

4 Conclusions

Are there applications which are more suitable for supercomputing than social computing? Does the choice of architecture matters to an application? Consider simulation of n- body atomic structures. It requires intense intra-communication- thus it may not be easily separable into self-contained small task units. Such computation may appear to be more suitable for supercomputing- than for social computing. However, often same problems can be decomposed in a different ways with larger grains.

However, the real issue is often the economics of supercomputing. Only very rich countries and communities can afford such computation power. Some of the principal usage of supercomputers are weapon systems design, weather forecasting, cryptographic code breaking, drug design, share market profit prediction, etc. However, often groups with smaller wealth may have requirement for larger computation (such as Diabetes or AIDS patients). Also many extreme computations appeal mostly to human curiosity than to profit possibility (such as searching for divine message in the digits of pi or finding space aliens). A key reason for the Grid rush worldwide is that community systems such as grid.org, SETI@home has been capable of harnessing super computing power at a cost about 100 times cheaper. A supercomputer today cost 0.5-100 million dollars. At US$100 million price tag IBM BlueGene/L with its 136 TFLOP is a surprisingly low cost system costing only 735$/GFLOPS. Just a year back the fastest NEC's Earth Simulator with US$110 million offered 36.4 TFLOPS costing 3055$/GFLOPS. Compare this with SETI@home which cost US$500,000 [1] and at conservative estimate now offers around 100 TFLOPS, and that means the cost of computation is $5/GFLOPS!

Finally, there is another new factor in the war of applications in the teracomputing world. Community computing will have a mind of its own! Social computing requires thousands of people to be motivated to participate. With the increase in the number of applications the openness and the social goodness of the applications will be increasingly at issue. Is it more likely that the ultimate grand challenge problem- financial modeling for worldwide poverty reduction/elimination, will finally garner more computing power than the designer atom bomb? Time will tell.

5 References

[1] Antony Davies: Computational Intermediation and the Evolution of Computation as a Commodity, Applied Economics, June 2004

[2] Roger Smith: Grid Computing: A Brief Technology Analysis, CTO Network Library, 2005.

[3] Buragohain, C., D. Agrawal, and S. Suri (2003): A Game Theoretic Framework for Incentives in P2P Systems. Third IEEE International Conference on Peer-to-Peer Computing (P2P 2003), Linkoping, Sweden.

[4] Barry Wellman, "Designing the Internet for a Networked Society." Communications of the ACM [Association for Computing Machinery] 45, 5 (May 2002): 91-96.

[5] Martin Wattenberg, & Fernanda B. Viégas, "History flow: results: What can we learn from history flow?", IBM Research, 2003, http://researchweb.watson.ibm.com/history/contact.htm.

[6] Schlessinger L, Eddy DM. Archimedes: a new model for simulating health care systems - the mathematical formulation. Journal of Biomedical Informatics 35: 37-50, 2002.-

[7] Eddy DM, Schlessinger L. Archimedes: A trial-validated model of diabetes Diabetes Care 26: 3093-3101, 2003.

[8] J. C. R. Licklider, Man-Computer Symbiosis, IRE Transactions on Human Factors in Electronics, volume HFE-1, pages 4-11, March 1960.