

Ph.D. Projects (2003/2004)

CS programme

Algorithms for Understanding Protein Structural Flexibility

Student : Anshul Nigham
Thesis Advisor (Singapore) : Asst Prof David Hsu
Thesis Advisor (MIT) : Prof Tomas Lozano-Perez

Project Abstract :

Proteins play a fundamental part in biological processes. The ultimate goal of molecular biology is to understand the chemistry and physics of these macromolecules as they interact with each other and their environment in numerous ways. Although initially thought of as rigid molecules, it is quite clear that proteins adopt a number of different shapes or "conformations", and that this ability is crucial to their functions. The study of these changes and how they occur is crucial to understanding biological processes. Modern drug design also benefits heavily from such knowledge as the cellular targets of many drugs are proteins. The goal of this work is to develop efficient computational approaches for studying protein structural flexibility.

Adaptive Distributed Services based on Peers Technology

Student : He Yuxiong
Thesis Advisor (Singapore) : Assoc Prof Hsu Wen Jing
Thesis Advisor (MIT) : Prof Charles E. Leiserson

Project Abstract :

This research project will design new architectures for distributed resources management and develop provably efficient algorithms that adapt to available computational resources. The key issue is to make the dynamic resource management systems self-configurable, fault tolerable, adaptive to varying demands and scalable to the change of available resources. Recent developments in distributed computing, e.g. peer-to-peer systems such as Chord or Koorde developed at MIT, offer promising features. Peer-to-Peer systems, or P2P systems for short, are distributed, adaptive, self-organizing systems without any central authority. Nodes or peers in P2P systems have equivalent capabilities and responsibilities. Our aim is to design, evaluate and implement a practical P2P infrastructure offering a number of basic services such as communications, discovery, range query, publish/subscribe mechanism, etc. To evaluate the cost and performance of a design, factors like node degree, query latency and topology maintaining overhead are measured and analyzed. We will provide new architectures based on structured P2P systems for efficient services that adapt to dynamic fluctuations in the overlay networks due to nodes joining, departure, and failures.

Memory Conscious Object-Oriented Programs

Student : Nguyen Huu Hai
Thesis Advisor (Singapore) : Assoc Prof Chin Wei Ngan
Thesis Advisor (MIT) : Assoc Prof Martin Rinard

Project Abstract :

We present a new (size-)polymorphic type system (for an object-oriented language) that characterizes the sizes of data structures and the amount of heap and stack memory required to successfully execute methods that operate on these data structures. Key components of this type system include type assertions that use symbolic Presburger arithmetic expressions to capture data structure sizes, the effect of methods on the sizes of the data structures that they manipulate, and the amount of memory that methods allocate and deallocate. For each method, it can provide expressions that (conservatively) capture the amount of memory required to execute the method as a function of the sizes of the method's inputs. The safety guarantee is that the method will never attempt to use more memory than its type expressions specify. We have implemented a type checker for programs written with our type system and used this checker to verify the memory usages for a variety of programs. Our experience is that the type system can effectively capture the memory system needs of our programs.

Robotic Mapping

Student : Ong Chen Hui
Thesis Advisor (Singapore) : Assoc Prof Leong Tze Yun
Thesis Advisor (MIT) : Prof Leslie Kaelbling

Project Abstract :

An important capability that truly autonomous robots must be equipped with is the ability to map the physical environment they are moving in, even if the environment is unfamiliar to the robots. The field of robotic mapping attempts to devise algorithms that equip robots with this capability. Specifically, this requires the robot(s) involved to learn a good policy of how to move through the environment while localizing itself and building a spatial model of the environment. Robotic actions are assumed to suffer from stochastic actuator errors and robotic observations cannot precisely place the location of the robot (i.e. the environment is only partially observable). In the most general case, the environment of interest may be either indoors or outdoors and the robots may operate singularly or in teams.

A literature survey has been conducted to identify the progress in this field. Most of the work to date focuses on the use of single robots to perform mapping, usually in the indoors environment. One of the earliest techniques applied to robotic mapping is Kalman

filtering, which falls under the category of Simultaneous Localization And Mapping (SLAM) techniques. While Kalman filtering is easy to understand and apply, the technique cannot support environments where there are indistinguishable landmarks.

To overcome this limitation of Kalman filtering techniques, we will use partially observable Markov decision process (POMDP) framework to build a good policy for single-robot mapping. POMDP provides a sufficiently rich framework to model a robot's uncertainties about its actions and observations. However, exact POMDP algorithms suffer from the curse of dimensionality and cannot solve robotic mapping problems of reasonable sizes. Several approximation algorithms have been proposed to date. Our objective in this project is to improve upon current approximate algorithms.

P2P Data/Content Sharing Systems and Data Integration

Student : Yu Bei
Thesis Advisor : Prof Ooi Beng Chin (Singapore)
Thesis Advisor (MIT) : Prof Stuart Madnick

Project Abstract :

Peer-to-Peer (P2P) systems attract much research concern today as a new architecture for distributed computing and the Internet. P2P systems can be described informally as a system in which large number of sites cooperate and coordinate to perform specific tasks. The sites in P2P systems are called *peers*, due to their similarity in capability and function within the system. Typically, peers can be any PCs connected to the Internet or an intranet. The peers are organized as an overlay network having its own protocols and routing mechanisms, which we refer to as *P2P network*. P2P systems are dynamic in nature. The peers can join and leave the network continuously, but the overall system performance does not degrade too much.

P2P systems cover broad applications including data sharing, distributed computing, communication and collaboration, and platform services. Our focus is on P2P data sharing systems. The motivation of P2P data sharing system is the rapid growth of the amounts of digital information in the World Wide Web. A P2P data sharing system is for sharing huge amount of data in the P2P network, where each peer contributes its own data and storage capacity, and benefits from the resources shared by other peers as well. Many systems and prototypes have been built for data sharing in P2P networks, and some of them have become very popular web services. Our current research aims to provide richer functionality to P2P data sharing systems such as supporting structured data sharing and integration.

Dynamic Code Optimization

Student : Zhao Qin
Thesis Advisor (Singapore) : Assoc Prof Wong Weng Fai

Thesis Advisor (MIT) : Assoc Prof Martin Rinard

Project Abstract :

As a new technique to complement traditional optimization techniques, dynamic code optimization provides opportunities for finding program execution performance bottlenecks and deploying optimizations at run-time. The runtime overhead is the main obstacle to dynamic code optimization, so how to reduce the overhead of runtime system is one of the major challenges.

Our project is to develop a complete dynamic code optimization system, which is able to monitor the program run-time behaviors, detect the performance bottlenecks, and perform optimizations. In the project, several techniques, such as utilizing hardware like performance counter, are explored to reduce the overhead on monitoring program execution and performing optimizations. Besides reducing the overhead, how to choose optimization and evaluate optimization results are also important. Making the framework adapt itself to different behavior of applications by applying different optimizations to obtain the maximum performance gain is another goal of the project.