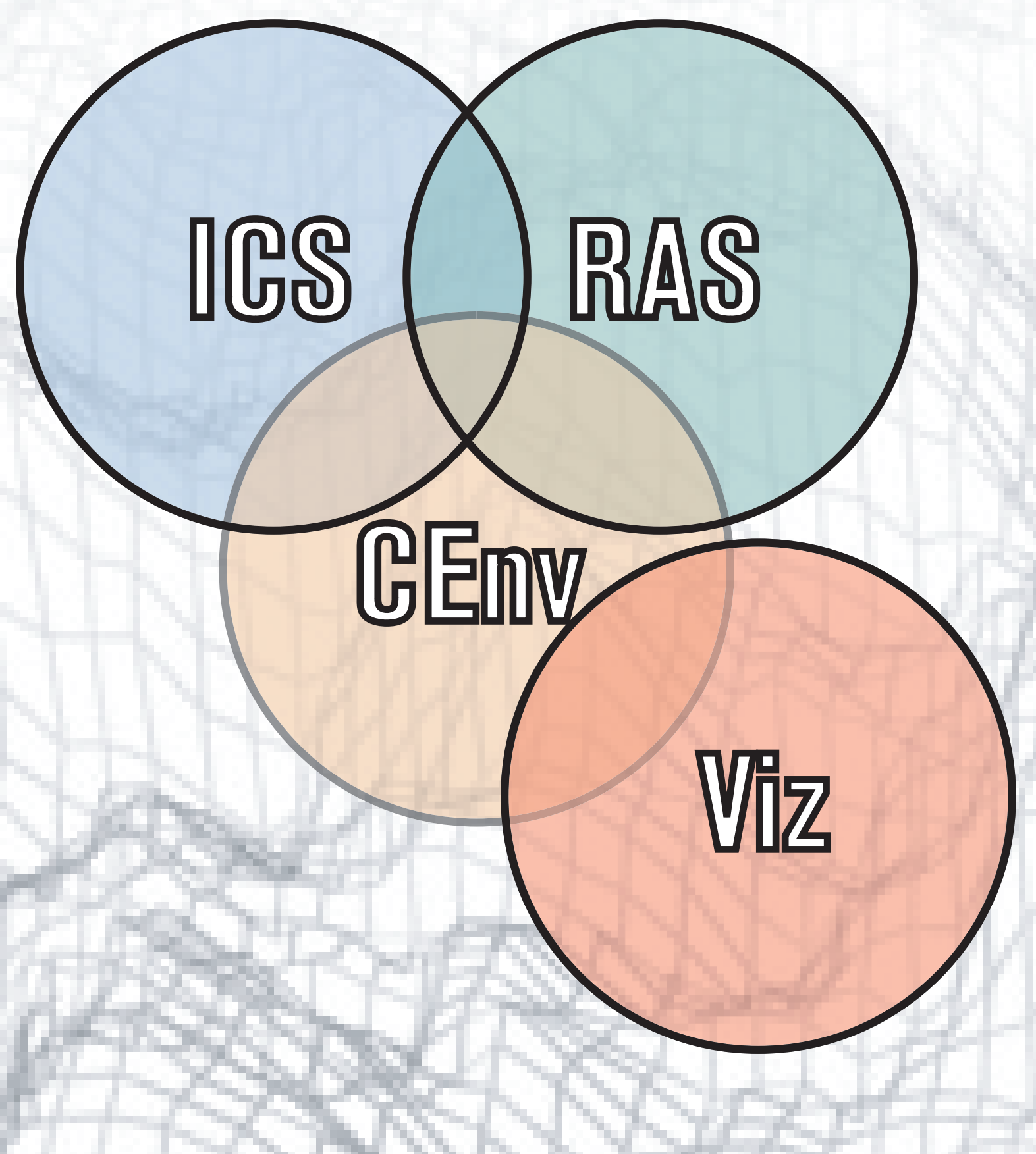
CHALMERS | GÖTEBORG UNIVERSITY

Distributed Computing & Systems

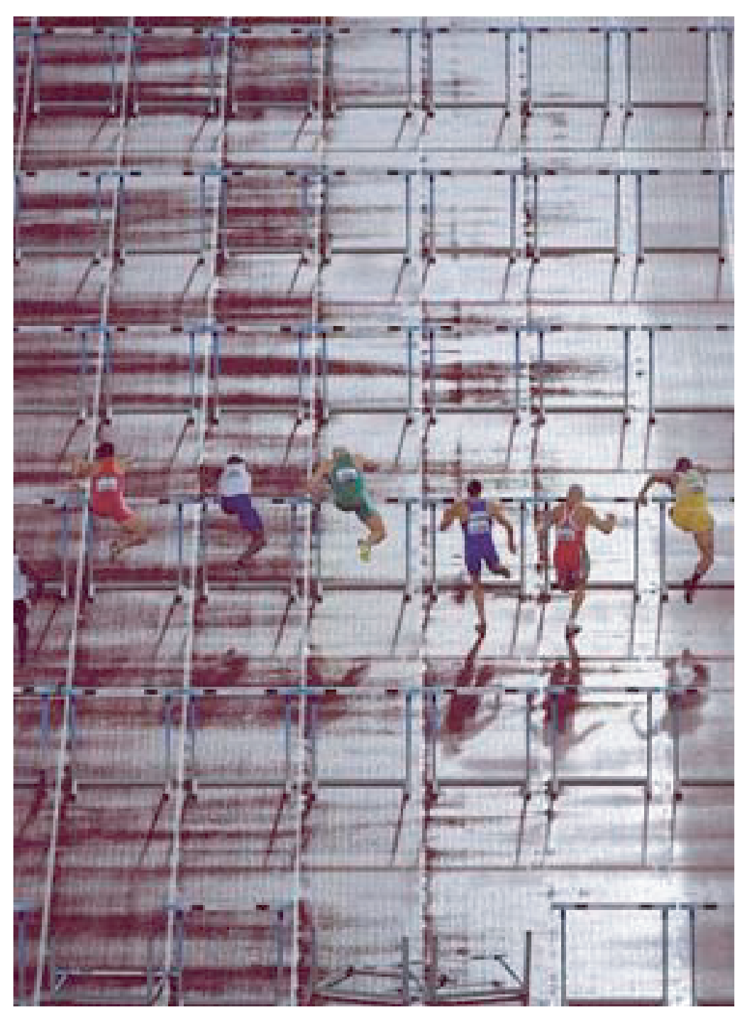
Department of Computing Science



In distributed systems, *sharing* is the key word; distributed systems share everything from information, cables and frequencies to printers and expensive computers. Our group studies efficient techniques for information sharing between different concurrent computer processes (ICS), but also efficient and fault-tolerant methods for sharing general resources on networks (RAS). Instead of sharing physical resources, many users would like to share views of virtual "worlds" that they can create together with colleagues or friends. Our group tries to see how to realize this in a way that is efficient and guarantees that people see a good approximation of what is happening in their worlds (CEnv). Now, if you think of how much information only our findings will generate on the network when implemented and try to imagine how much more digital information is generated in such complex systems, it is probably easy to get convinced that information visualization is the way to go to illuminate the information that the user is interested in (Viz).

Interprocess Communication & Synchronization (ICS)

When several processes run at the same time, there must be some way for them to inter-communicate and pass information. Processes can communicate and synchronize via shared data objects, conventionally implemented with locks to ensure consistency. However, this method has many disadvantages such as reducing parallelism, having weak fault-tolerance, risk for priority inversion, and more. On the other hand, *non-blocking synchronization* -- that avoids the use of locks -- can potentially maximize the parallelism in multi-processor systems.


Using wait-free synchronization, all concurrent operations can proceed independently of the others.

Locks may imply large overhead for small processes.

Non-blocking Synchronization

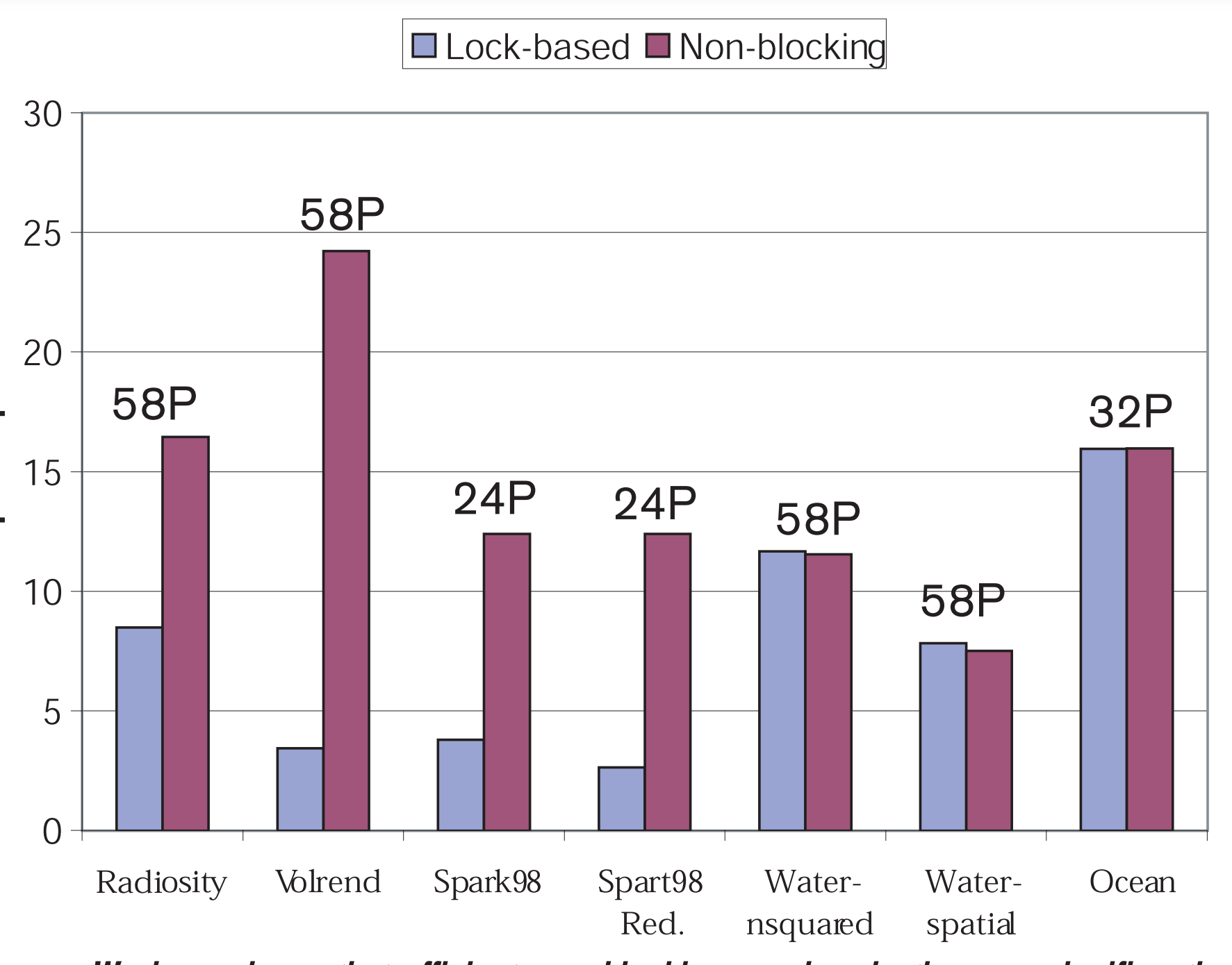
Non-blocking synchronization can be realized using *wait-free* or *lock-free* methods. Wait-free methods give better guarantees regarding fairness and predictability, but may result in relatively more complex constructions. However, by exploiting information about the execution environment, e.g. task characteristics in periodic real-time systems, it is possible to have significant improvements in this respect.

Research Perspectives

- Efficient use of non-blocking synchronization to enhance performance of parallel applications and algorithms.
- Enhancing performance by cooperative scheduling and synchronization.
- Exploiting information from the execution environment to provide real-time guarantees and simple and practical wait-free synchronization.
- Design and implementation of a library of non-blocking shared data objects for parallel and distributed systems. See NOBLE, *http://www.cs.chalmers.se/~noble*

Applications

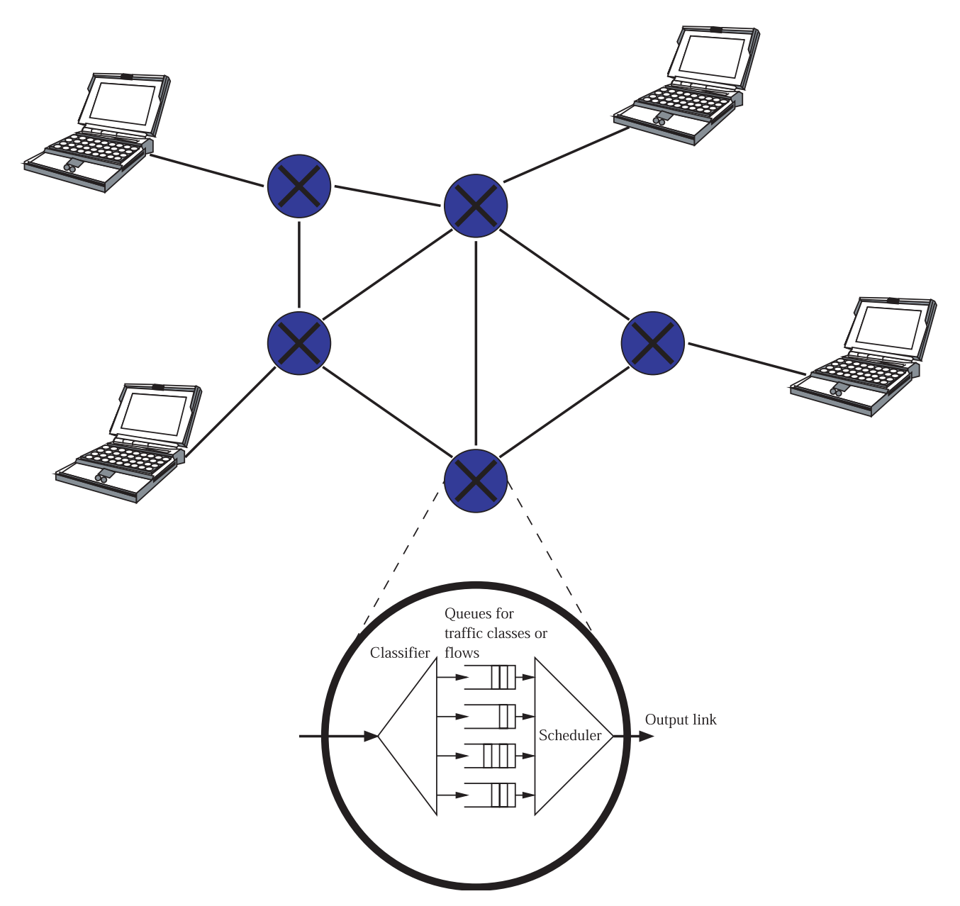
- Real-time (operating) systems, general and special-purpose: synchronization and scheduling
- Operating systems, programming languages: memory management; garbage collection, synchronization libraries
- Synchronization in embedded systems
- Parallel applications: Volume Rendering, unstructured finite element simulations, etc



We have shown that efficient non-blocking synchronization can significantly improve the performance of many parallel applications.

Resource Allocation & Sharing in Networks (RA)

In allocating and sharing network resources, key goals are *efficiency* and *fairness*. Besides, a wide variety of applications, which may have very different requirements on the quality of the provided service, must be taken into account. The ability to provide a set of different service levels instead of just best effort is referred to as *Quality of Service* (QoS).



Applications

- Router packet scheduling
- Admission control in networking
- Services for multimedia applications
- Resource sharing in mobile communication: data, frequencies, computational units/capacity

Research Perspectives

- Exploiting locality in sharing and decision-making, aiming at improving performance and fault-tolerance
- Addressing trade-off issues and decisions in parametrized Quality-of-Service.

Our research in this field focuses on:

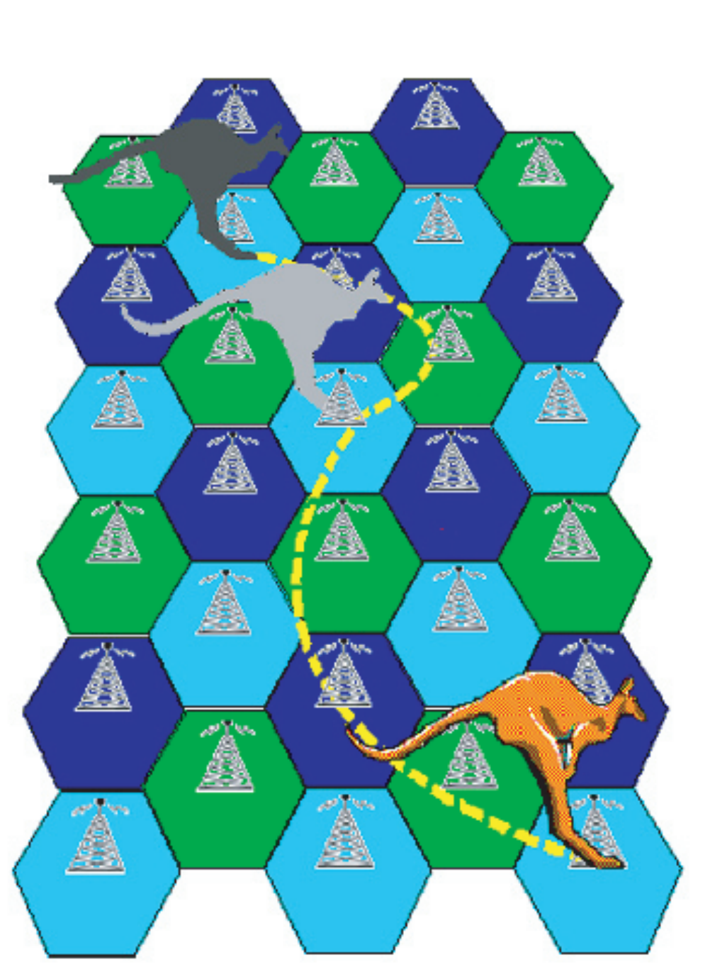
Admission Control and Traffic Scheduling in General-Purpose Networks

One of the keys to provide Quality of Service in a general-purpose network is for the routers to be able to decide for each packet which service it is entitled to and how to ensure that it receives that service. This involves proper classification and admission control of traffic, together with appropriate scheduling of packets:

- Packet scheduling addresses the problem of how the packets queued in a router should be serviced in order to meet the different quality of service guarantees associated with the packets. Different packet scheduling policies imply different bandwidth, fairness, and delay guarantees for traffic.
- Admission control ensures that the demand on the services is within the limits of what the system can handle.

Resource Allocation in Cellular Networks

A basic resource in cellular networks is bandwidth (frequencies, channels). To allocate channels in cellular networks one needs to meet a number of performance optimization criteria -- e.g. frequency reuse and the number of satisfiable requests should be maximized, connection setup time and communication complexity should be minimized -- and a number of interesting trade-offs involved in trying to meet them. Moreover, the impact of failures in the network should be as small as possible.



To achieve *dynamic assignment* and maximal *fault-tolerance*, distributed solutions are appropriate, where the base stations "negotiate" between themselves, instead of depending on some central stations/switches which may be some distance apart. We are working on dynamic solutions, including tuning *trade-offs* of performance parameters and quality-of-service (QoS) parameters, as these are crucial in determining the desired behaviour of the system.

Contact Persons
Marina Papatriantafilou, Philippas Tsigas | *Tel:* +46 31 772 5413 (5409)
Fax: +46 31 16 56 55 | *Email:* <ptrianta, tsigas>@cs.chalmers.se

Conceptual layout by Robert Karlsson -- Thanks!