Cheese Factory Project

Chedar - User Stories

Mikko Vapa
mikvapa@jyu.fi
Matthieu Weber
mweber@mit.jyu.fi

June 24, 2002

Abstract

Chedar is a platform for finding and using resources in a distributed network of workstations. User Stories -document describes the usage scenarios told by the potential users of Chedar platform. Users are considered to be software architects that implement a system using Chedar's middleware services. In this document users state their platform requirements which in the future will be used as a basis for designing features and writing technical specification. Each user story has been ranked by its priority to define in which part of the implementation process the story will be implemented.

Contents

1 Introduction

Chedar platform uses peer-to-peer communication to find resources across the network of peering nodes. The nodes form a community concentrating on solving a specific problem. Such kind of problems are found for example in distributed computing where the problem is to find idle workstations or in distributed data sharing where the aim is to find free disk space.

The peer-to-peer network consists of nodes routing query messages about resources. Nodes have an ad hoc (dynamic) behavior meaning that nodes can move, new nodes can enter in the network and existing nodes may disappear from the network at any time. When nodes enter, move and leave the network the topology continuously evolves and nodes seek for a stable and predictable neighbours to optimize their state of operation. Nodes also seek for short distance connections to avoid network load. The network structure does not have any central point and therefore is more resilient to targeted attacks and failures.

The document is divided into seven chapters each describing different aspects of the user stories. User stories are described in brief and potential technological solutions needed to fulfill each user story is marked in parenthesis. These solutions will be defined in detail in Chedar's technical specification in the future. Also a priority for each user story has been allocated to illustrate the importance of the feature and to determine when the feature will be implemented.

2 Entering and Leaving the Network

The Chedar platform should support a usage scenario where user logs in the system, submits a task for processing and logouts. After a while the same user would log in again at any place of the network and retrieve the results from the processing and logout. (permanent user id, authentication, distributed database for results, *high priority*)

The user should dock into the system automatically. This means that no prior knowledge of existing Chedar nodes would be present but the nodes would by theirselves find a node to connect. (address auto configuration, search for similar IP address, *low priority*)

After logging out from the system the user should continue from the same point he was earlier connected when logging in again into the network. (list of neighbours and the earlier status of the links are stored in the history, *high priority*)

When attaching to the network the node should find close by neighbours to connect based on a metric thus avoiding unnecessary network load coming from long distance connections. (neighbour discovery with qualityparameters, *high priority*)

3 Connectivity

Chedar should locate all the nodes in the network (limited by a available bandwidth) such that queries reach the maximum number of hits. (broadcast search, logical search nets for different resource types, query dropping based on the node's throughput capability, *high priority*)

Chedar should give direct access (point-to-point connections) to nodes for reserving resources. Direct access is needed after resources are found and started to be used. (Chedar for searching resources and IP for using these resources, like in Gnutella, *high priority*)

Firewalls block connection queries from outside the protected intranet. Nodes behind these kinds of firewalls should also be able to connect to the network and a communication between two nodes which both are behind firewall should be possible. (use of intermediate node outside the firewall as a gateway link, *high priority*)

Chedar should use every kind of available transmission medias for the connections including air, cable and fibre. (multiple network interfaces, ordered use of interfaces e.g. GSM if Ethernet does not work, but not before, *low priority*)

4 Resources

Chedar should present a list of resource types the neighbouring nodes can support. Resource types might be for example processing power, storage space, file and a printing or displaying service. Resource types should also have different characteristics for example if the resource type can be moved or duplicated and whether the resource changes or is always in a static state. In general Chedar should support many types of applications e.g. discussion groups, file transfer, problem solving, data storage and computing power consumers. (neighbour discovery with resource types, attributes for different resource types, *high priority*)

The nodes should be able to have different roles based on their capabilities. For example one node can act as a task processor and the other as a result file illustrator. (nodes will form different roles based on the usage and provision of resources during the network evolution, *high priority*)

After sending a resource query a list of found resources should be given to the user that can be used to choose the best alternative. (broadcast routing of queries and their reply packets to a source node, *high priority*)

The search should be fast and effective and the results should not be out of date for example realtime information about computing resources must be up to date rather than minutes old. (logical topology follows most effective search results based on history, topology is designed to have no unnecessary loops, *low priority*)

Some of the resources might be constantly changing and there might not be a checkpoint for a final version. Such kind of resources might be for example an air situation picture that is constantly updated. Nodes that use result files describing the air situation picture might want a constant stream of data and avoid delays caused by a temporal storage. (stream subscription, multicast, joining into a stream and leaving it, *low priority*)

The user should be able to choose what resources he wants to publish for common use. (access list for node's own resources, *high priority*)

File resources for example computing job or result data should be transferrable between different workstations using a direct point-to-point connection with standard transfer protocols. (Globus file transfer service GridFTP, *high priority*)

Computing resources should be accessbile by different workstations using a direct point-to-point connection (Globus job allocation using Resource Specification Language, Globus job execution, *high priority*)

5 Network Performance and Scalability

Chedar network should use minimal amount of physical network bandwidth. (logical connections should follow physical connections as much as possible, limited broadcast, nodes should prefer short distance connections and high bandwidth links by continuously measuring neighbour connections, physical local area network clusters should be treated also as logical clusters with one clusterhead, *low priority*)

Unneeded neighbours should be overtaken to reduce network traffic meaning that node should try to get closer to the nodes with which it exchanges lots of data. In order to do that, it should drop an existing connection and open a new connection to another node, that would bring it closer to its destination by one hop. (network evolution using overtaking algorithm, *high priority*)

Chedar should give also predictions about the connection quality parameters (delay, jitter and bandwidth) that can be used for decisions about which resources to use. (nodes measure the end to end delay of responses to resource queries, *high priority*)

The network should scale to millions of nodes. (hierarchical and cluster-based routing schemes for nodes with high bandwidth connections and static behavior, *low priority*)

The physical network bandwidth usage should be limited for example to support low bandwidth connections in wireless networks. (the nodes should measure the outgoing traffic and reduce activities if limits are reached, *high priority*)

Chedar should be able to efficiently transfer huge files. (point-to-point transferring of data, peer-to-peer routing for metadata, *high priority*)

6 Security, Fault and Attack Tolerance

The network should sustain random failures and adapt fast to link failures. Topology should be kept as consistent as possible all the time. (neighbour connections are constantly monitored, existing connections dropped and new connections established, new connections are based on earlier history of known IP addresses, peer-to-peer network is by default resilient, *high priority*)

The network should be decentralized without any point that might be target for attacks. (clusterheads are elected if existing clusterhead fails based on their fitness/reputation, resource databases are located in each node, random hardware failures look like a node leaving a network, *high priority*)

Most of the attacks are targeted to the physical network, but Chedar should still continue its operation in a disjoint network and find if possible other alternatives to reach destinations. (parallel networks with IP layer might be used for backup connections e.g. radio links and GSM, disjoint network clusters operate like a smaller network compared to the original, nodes adapt their connections and form a new community, *low priority*)

The attacks to the network should have a minimal impact on Chedar. (logical connections follow as much as possible physical connections to avoid breakdown of all connections when only one link breaks, *low priority*)

Important nodes (for example air situation picture illustrators) should always keep backup connections if possible. (node might be forced to use ring topology with other half of the ring consisting of wired IP network and other wireless IP network, topology with backup connections, *low priority*)

Important data should be able to be replicated. (sleeping clones with heartbeat monitors and data synchronization, the resources are duplicated whenever they are needed resulting in a native replication of data, gossip architecture, *low priority*)

Chedar should be resistant to denial of service/network overload attacks. (inside Chedar connections producing too much traffic are dropped by default because they are ranked with low quality, if the physical network is overloaded the denial of service cannot be avoided with Chedar's mechanisms, *high priority*)

The users of the network should be authenticated. (distribution of certificates beforehand, *low priority*)

Nodes of the Chedar network should be accountable. (if resource usage does not result in an expected behavior for e.g. result files cannot be understood the node responsible for the computing is banned, this info might be also distributed further to other nodes, a reputation systems, *low priority*)

Some nodes might not be accountable so to suppress the effect of lying behavior two level of information reliability should be taken into consideration:

belief : information that is received from another node (which might have been lying or speaking thruth)

knowledge : information that has been verified and is considered as a thruth (the user has given a confirmation)

Nodes should prefer verifying if information is only a belief or if it should be treated as a knowledge. (decisions based at first on belief, but after verification knowledge will be the main decision factor, information is classified in these two categories)

Not all resources should be accessible to everyone. (credentials for using some of the resource types via certificates, ranking system of users, users might approve the usage of a particular resource, *low priority*)

Man-in-the-Middle attacks where a node in the middle spoofs its identity and acts as a false destination forwarding modified packets to the right destination should not be possible. (source and destination addresses can be spoofed, also the resources, is there a protection of this type of an attack other than strict authentication of Chedar nodes that no malicious nodes enter the network, fingerprints, signatures, *low priority*)

The network should produce theater traffic when needed to hide the real use of the network. For example nodes might be constantly sending some data to the network to make it more difficult to estimate what a particular event causes to the behavior of a network. Without theater traffic it might be easy to predict what kind of an information triggers sensors and how much data they produce. (nodes generate random data and aim to keep a constant bit stream between nodes, random data is dropped in every node, *low priority*)

7 Diagnostics

To investigate the behavior of the network a log file about activities should be kept and sent to a central directory for further analysis. (nodes can collect information of their activities and evolution, network can be monitored to get a picture of whole network traffic, the attributes being logged could be selected from a list, an external management application, *high priority*)

The logging information should be able to be reduced and further processed for example by visualizing the evolution of network topology. (log files are written in standardized format that can be used by network topology visualisation tools, *high priority*)

8 Implementation

The system should be portable in various operating systems and different hardware platform for example to the wireless mobile devices. (implementation in Java and under JXTA framework will increase the portability, *high priority*)

The system should be compatible with heterogeneous network layers. (Internet Protocol as a network interface will ensure compatibility with most of the wired and wireless networks, Chedar is not actually tied to any network layer at all, *high priority*)

9 Conclusion

The user stories described in this document define the behavior of Chedar platform. These requirements will be taken into account when designing a technical specification about the software implementing Chedar. High priority stories are implemented in the first phase of development and the lower ones will be investigated later. The aim of this document is to help clarifying the structure of the network and should raise discussion about the features that have not yet been taken into account. Also the document should help finding a common understanding what kind of system we are about to build and if the requirements are aligned with Cheese Factory's client's needs.