

School of Science and Technology

### Methodology for Computer Science Research Lecture 1: Introduction

#### Andrey Lukyanenko

Department of Computer Science and Engineering Aalto University, School of Science T-110.6130@aalto.fi

September 13, 2012

## **Course overview**

Code: T-110.6130

Name: Methodology for Computer Science Research

Contact: T-110.6130@aalto.fi

Aim: Study of methods, tools, and development of reading and writing skills.

Structure: 6 Method lectures, 2 presentations, half lectures are removed for home study.



Lecture 1: Introduction September 13, 2012 2/49

# To pass the course...

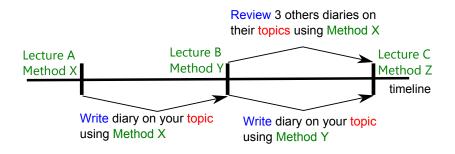
... during the course:

- 1. Pick *one research topic* of your interest: select one of the provided by us or choose yourself (be cautious!).
- Attend the lectures: Thu/Tue, 16:15-18:00 at T5 13 Sep, 20 Sep, 04 Oct, 11 Oct, 18 Oct, 30 Oct, 06 Nov.
- 3. Write diaries after each lecture related to the methods and your topic.
- 4. Review diaries of others in a week after each lecture.
- 5. Write an assignment on the topic you chose (here the diaries could help!).
- 6. Short presentations on your topic 11 Dec and 12 Dec.
- 7. Checkout English courses in Language Center if you need help?



## To pass the course...

... during the course:





Lecture 1: Introduction September 13, 2012 4/49

# **Credits and grading**

Credits: 5 credits

Grading:

- Diaries give 50% of the mark.  $(g_d)$
- Presentation gives 20% of the mark.  $(g_p)$
- Assignment gives 50% of the mark.(g<sub>a</sub>)

The final grade g will be calculated as

$$g=\min\left\{rac{50\cdot g_d+20\cdot g_
ho+50\cdot g_a}{100},5
ight\}.$$



Lecture 1: Introduction September 13, 2012 5/49

### **Assignment topics**

Each student have to pick up one topic. During the course produce analysis of the topic it with studied methods.

#### Topics are...

- Congestion control in TCP.
- Fairness int TCP.
- Distributed Hash Tables (DHTs).
- Unstructured Peer-to-Peer (p2p).
- Cloud computing Systems.
- Mesh Networks.
- Sensor Networks.
- Ad-hoc Networks.
- Social media.
- Delay tolerant networks.





Lecture 1: Introduction September 13, 2012 6/49

# Assignment topics (cntd)

- Security in DHT.
- Internet of things.
- Datacenter architecture.
- BitTorrent protocol (tit-for-tat).
- Routing protocols in the Internet.
- Publish/Subscribe systems.
- P2P reputation systems.
- Energy consumptions in Wireless systems.
- Denial-of-Service attack.
- Multicast protocols.

OR you can choose your own topic.

It may be what you have as Master Thesis topic, or any topic your are interested in.



# **Structure of the Course**

Course outline:

- 1) 13.09 Introduction (this lecture).
- 2) 20.09 Computer Simulation.
- 3) 04.10 Data analysis.
- 4) 11.10 Mathematical modeling.
- 5) 18.10 Academic programming.
- 6) 30.10 Experimental research.
- 7) 06.11 Network business models.
- 8) 10.12 Presentation 1.
- 9) 11.12 Presentation 2.

Assignment deadline is 01.12.



Lecture 1: Introduction September 13, 2012 8/49

# Studying process (1/3)

- After this lecture pick up own topic or 3 of our topics,
- Send a topic or the list to T-110.6130@aalto.fi with title "T-110.6130 assignment topic" (easier to find).
- Inside e-mail write your own topic; also say few words about the reason why did you choose it,
- or list 3 topics by priority from the provided ones, e.g.

"My priority list of topics is:

- 1. TCP.
- 2. DHT.

3. DoS attack"

or "Own topic: <Title> (I choose it because it's my MSc topic)".

Before lecture 2 you will be provided with a unique topic.



# Studying process (2/3)

After each methodological lecture (lectures 2-7) you

- write a short diary note (1 page IEEE double column format, no need to write too much, no need for introduction, title or conclusion; see Diary Instructions in Noppa.)
- upload (pdf and tex) the diary to optima.aalto.fi

When logged in to optima, you will find T-110.6130 workspace with diary subsection. The diary on previously studied method should be uploaded before next lecture (or during one week)

review 3 other's diaries from previous lectures (Diary grade is a combination of your diary quality and your review quality; Reviews are given as comments in optima.)

In this short diary you write how to use exactly given research method for your topic.

Warning: Avoid unnecessary information in diaries. Abstract, Introduction and Conclusion will be only in final assignment.



# Studying process (3/3)

- ► Last 2 lectures are presentations. All students will have short presentations (≈ 5min) on what they have studied during the course on their selected topic.
- One week before the presentations is an assignment deadline. The assignment is to cover the topic you choose with methodological view.

Your paper should:

- contain a short introduction to the topic,
- clearly state all methods used to study the topic in literature,
- compare them (pro and con),
- present own thoughts: what in the study is missing and why?

Remember: Your diaries on the same topic will help you with the final assignment!



# What is this course about?

This course is about Scientific Research in the field of Computer Science (more precisely, in the field of Data Communications).

The course tries to answer on the questions:

- How to do the Scientific Research?
- How to do the Scientific Research efficiently?
- How to do what a Scientific Community needs, in the form which the Scientific Community demands?
- How to present your Scientific Research to the Community?

Although, the above in context of Scientific Research, the same skills are useful in any kind of IT related work.



# What is Computer Science Research?

It is about studying an Idea: your Idea.

Novelty of the Idea.

Research is a study of new ideas in the field where the research belongs to.

Significance for the Community.

One of the most important questions of research is to understand what kind of idea is actually needed for the community "today".

Contribution from the Researcher.

The amount of efforts made by a researcher to study the idea.

But before...



Lecture 1: Introduction September 13, 2012 13/49

## But before...

... understanding *Novelty* and *Significance* you have to know the state-of-the-art of knowledge in Scientific Community.

#### How to be up-to-date?

- 1. Read recent journal articles, and conference papers. Almost all of them has "History", "Introduction" and "Future work" parts. (they correspond to "Past", "Current" and "Possible Future" of the research.)
- 2. Talk to colleagues and scientific advisers :) (they may suggest ideas and explain the field development, without studying).
- 3. Observe the business tendency and technology levels (news from industry).
- 4. Look through the visions of the future (Sometimes knowledgeable people publish their visions of the future).



#### Literature sources

#### The search engines (and sources) for scientific publications.

- Google Scholar: http://scholar.google.com
- Academic Microsoft: http://academic.research.microsoft.com
- ACM Portal: http://portal.acm.org
- IEEE xplore: http://ieeexplore.ieee.org

Especially, papers published in famous conferences, e.g.,

- ACM SIGCOMM: http://www.sigcomm.org
- INFOCOM: http://www.ieee-infocom.org

*Additionally*, many famous publications appear in less famous, but still important conferences.

AR - acceptance rates for the conferences and IF - impact factor for the journals.



# Accessing the publications

1. Traditional way: Go to the library and get an article or order one (an obsoleted way).

Unfortunately, the articles and conference books in the library are quite old. Some journals are available in the coffee room.

- 2. Internally: Inside Aalto University ACM, IEEE, Springer, etc websites allows to fetch articles freely.
- 3. Remotely: Outside Aalto University you can fetch them
  - directly from the Internet, some of them are publicly available
  - indirectly using the search site nelliportaali.fi or adding the proxy libproxy.aalto.fi, e.g. portal.acm.org.libproxy.aalto.fi



## **Reading as a part of Research**

The reading refers to the studying of the field (remember *Significance* and *Novelty*?).

Reading:

- adds knowledge about the field.
- adds the confidence in own knowledge about the field.
- helps new Research Ideas to pop up in the mind.

Do not underestimate the Reading as a part of Research:

- Even if you have the full confidence in the new Idea, check the literature, search for it.
- If the Idea popped up after reading some paper, check who citing this paper. May be the Idea was already developed.

Remember: the previously mentioned paper search engines are able to search by criteria: "cited by".



# Writing as a part of Research

The writing refers to the production of own Research (recall *Contribution*!).

Writing:

- allows you to document your work for own needs.
- allows others to see your work, to see that you are actually working.
- putting Ideas on a paper allows to polish them and even invent new or extend existing Idea.

Writing is always hard in the middle of research, but it will greatly help you later if you put on the paper even small Ideas, points, thoughts.



Lecture 1: Introduction September 13, 2012 18/49

# $\textbf{Reeding}\longleftrightarrow \textbf{Writing}$

Question: When should I switch from reading to writing?

Answer: Never.

- Starting the research you mainly read.
- Finishing the research you mainly write.
- In between, you write, but continue to keep abreast of the development of the Community.

Conferences happen all the time, papers appears. If you produce your research based on other authors paper, always check who is citing it.

Question: When to switch from mainly reading to mainly writing?

Answer: Whenever you have confidence in the field and double checked the Idea.



Lecture 1: Introduction September 13, 2012 19/49

# What is an outcome of the Research?

Accomplished research is determined by written results.

Outcome of the Research may be

- a survey of the field, if it is necessary overview, timely and shows new facets of the field,
- a new algorithm/protocol, if it gives some benefits compared to already existing ones,
- a mathematical model of a protocol/algorithm, if it is better predicts different features of the protocol/algorithm,
- a performance measurement of existing protocols, with additional analysis
- many more...

All these are "Scientific Findings". Your written results should address it clearly.



Lecture 1: Introduction September 13, 2012 20/49

## How to develop a new "clever" idea?

There is no rule for Idea generation process, but when you have *an Idea* remember:

"There is nothing new except what is forgotten" (c) Rose Bertin



- Whenever you have a new idea, double check that it was not studied previously. Even in close/related fields.
- It can be an old idea from a different field, but was forgotten and the time for it has come.

The checking allows to skip waste of time to study something that already was studied and will concentrate on application of the idea to the field.



Lecture 1: Introduction September 13, 2012 21/49

#### Before the research...

There is a big difference what you see when you look inside your paper, and what others see!

Good to know from the start: Peer Review a common practice in the Scientific Community.

An Example: Peer-review process in SIGCOMM: Papagiannaki, K. and Rizzo, L. 2009. *The ACM SIGCOMM 2009 technical program committee process*. SIGCOMM Comput. Commun. Rev. 39, 3 (Jun. 2009), 43-48

Find and read the paper yourself, you know how and also read the one which is citing it!!



Lecture 1: Introduction September 13, 2012 22/49

# **TPC Review System (An Example)**

Traditional points of review process:

- \*\*\* Contribution: Rate the evaluation of work and contribution.
- \*\*\* Significance: Rate the significance to theory and practice.
- \*\*\* Novelty: Rate the originality and novelty.
- \*\*\* Relevance: How relevant is the paper to the call for papers?
- \*\*\* Readability: Rate the readability and organization of the content.
- \*\*\* Overall recommendation: Would you recommend this paper for conference?
- \*\*\* Best paper award: Do you consider the paper a candidate for a best-paper award?
- \*\*\* Detailed comments



# When you are finally ready!

Ready?

- You have the Idea.
- You have the confidence in the idea (novelty and significance).
- You need Contribution!!!

#### This is what the course is about.

The optimal process of the idea study is not unique and is fully dependent on the case, however, it has a set of known study methods:

- Mathematical Modeling.
- Computer Simulation.
- Experimental research.
- Data Analysis.
- Software Development (demo or product).



# Methods: Mathematical Modeling

*Mathematical modeling* is a research method that performs the problem abstraction, when different properties of a system are defined using a set of parameters and interactions of these properties are defined with functions and inequalities over the parameters.

Mathematical modeling provides a set of features, it allows to

- investigate properties of the whole system, based on a subset of measured parameters.
- see the system's asymptotic behavior.
- find optimal conditions for a system.



Lecture 1: Introduction September 13, 2012 25/49

# **Methods: Computer Simulation**

*Computer Simulation* is a research methods, when a small sample program representing the studying algorithm/protocol is created for an existing toolkit (or seldom from the scratch), which simulates the network, computer or system work.

Computer Simulation allows to

- produce "cheap" evaluation research.
- do research, when the development time is crucial.
- do research, when the sources (money, number of devices) are limited.
- do research even in the black-box architecture, real development is limited.



Lecture 1: Introduction September 13, 2012 26/49

# Methods: Experimental research

*Experimental research* is a research method, which is mainly based on an experimentation.

Experimental research allows to

- produce a research even in case if the modeling is difficult.
- acquire result, when a simulation may be a very slow process.
- see non-trivial dependencies between parameters.



Lecture 1: Introduction September 13, 2012 27/49

# Methods: Software development

*Software development* is a part of the research, when a product-like software (demo) is produced. Sometimes it is even in the form of commercial product, i.e., this methods allows to show that the research idea is fully feasible.

Software development allows to:

- create a proof-of-concept.
- see design errors/pitfalls in the idea.
- produce the research with most realistic environment.



Lecture 1: Introduction September 13, 2012 28/49

# Methods: Data Analysis

*Data Analysis* is a research method, that in a form of a bridge, connects together other methods, allowing to compare results, make the research consistent in different aspects and produce estimations for parameters.

Data Analysis allows to

- match mathematical models and experimental research.
- give estimations on the parameters for models and algorithms.
- prove some properties with high probability rates.
- provide reader with easily readable and understandable data.



Lecture 1: Introduction September 13, 2012 29/49

# Methods: Networking business methods

*Networking business research methods* are the research methods examines telecommunications from the business point of view.

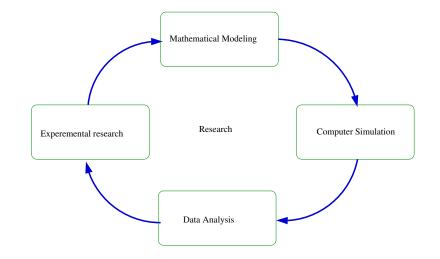
Networking business methods allows to

- study commercialization of the research.
- focus on the factors affecting the commercial success.
- predict future trends of the research and industry.



Lecture 1: Introduction September 13, 2012 30/49

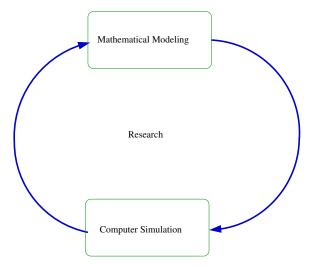
# Iterative process: Big Cycles





Lecture 1: Introduction September 13, 2012 31/49

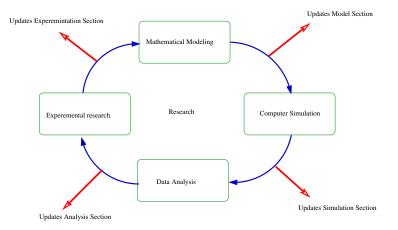
## Iterative process: Short Cycles





Lecture 1: Introduction September 13, 2012 32/49

### Iterative process: with paper outcomes





Lecture 1: Introduction September 13, 2012 33/49

## On the paper.

Section	Content
-	Title
-	Abstract
1	Introduction
2	History (Related work)
3	Idea, Algorithm
4	Model
5	Simulation, Measurements
6	Evaluation, Data Analysis
7	Implementation (Demo)
8	Discussion (Results), Future work
9	Conclusion
-	Reference



#### On the paper: General.

#### Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MIT Laboratory for Computer Science chordiftics mit adu http://pdos.ics.mit.edu/chord

Abstract

A fundamental problem that confronts new to peer applications in a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key'data stem pair at the node to which the key many. Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is contanously channing. Result from theoretical analysis, simulations, and ennetments dony that Chord is scalable with communication cost and the state maintained by each node scaling logarithmically with

#### 1 Introduction

Peer-to-peer systems and applications are distributed systems the software remains at each node is examplent in functionality a long hot redundant storage, permanence, selection of nearby servers, anonymity, search, authentication, and hierarchical man ing. Descript this rich set of features, the core countion in most peer-to-neer systems is efficient location of data iteres. The contripeer-to-peer system with frequent node arrivals and departures

The Chord protocol supports just one operation: given a key, it maps the key onto a node. Depending on the application using Chord, that node might be responsible for storing a value associated attigg keys to Chord nodes. Consistent hashing tends to balance load, since each node seceives roughly the same number of keys, "University of California Redulay interaction backalay adu

Authors in peverse alphabetical order

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Sys-tems Center, San Diego, under contract N66001-00-1-8933.

Permission to make digital or hard expires of all or part of this work for personal or characeen new is grazed without the perceluid that copies new not made or durationed for profit or commercial advantage and that copies ben this notices and the full citation on the fort pape. To copy otherwise, its Permission, Report and Servers or Personnellate to Loss, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-30113-411-8010008. 35100.

and involves relatively little movement of keys when nodes join and leave the system Previous work on consistent hashing assumed that nodes were

aware of most other nodes in the system, making it impractical to scale to large number of nodes. In contrast, each Cheel node needs "rooting" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-node cystem, each node maintains information only about (When N) other nodes, and resolves all lookups via O(log N) mensages to other nodes. Chord maintains its routing information as nodes ion and leave the system: with high probability each such event results in no more than O(log<sup>0</sup> N) Three features that distinguish Chord from many other peer-to-

peer lookup protocols are its simplicity, provable correctness, and mouthle nerformance. Chord is simple, routing a key through a se orange of Otlor N) other nodes treard the destination. A Chord node requires information about  $O(\log N)$  other nodes for efficient costing, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will ioin and leave arbitrarily, and consistency of even O(log N) state be correct in order for Chord to guarantee correct (hough slow) corring of essenter. Cherd has a simple algorithm for maintaining on in a dynamic enviro

The cest of this paper is structured as follows. Section 2 compares Chord to related work. Section 3 presents the system model that motivates the Chord motional. Section 4 presents the bare Chord motional and protect several of its momenties, while Section 5 presents extensions to handle concurrent joins and failures. Section 6 demonstrates our claims about Chard's performance through simulation and experiments on a deployed prototype. Finally, we outline items for fature work in Section 7 and summarize our con-

#### 2. Related Work

While Chord many keys onto nodes, traditional name and location services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleaver, the rest of this section assumes a Chord based service that maps keys onto values. DNS provides a host name to IP address mapping [15]. Chord

and the associated IP address representing the value. Chord requires no special servers, while DNS relies on a set of special root



Lecture 1: Introduction September 13, 2012

#### On the paper: Abstract.

Abstract:

When somebody finds out this article this is what is read first. Based on abstract a person should be able to preliminary know, should he/she read it further or not?



Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MIT Laboratory for Computer Science chordiftics mit adu http://pdos.ics.mit.edu/chord

#### Abstract

A fundamental problem that confronts neer-to-peer applications in to efficiently locate the node that stores a particular data item. This paper presents Chowl, a distiliated lookus protocol that addresses a key, it many the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key'data stem pair at the node to which the key many. Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is contanously channing. Result from theoretical analysis, simulations, and ennetments dony that Chord is scalable with communication cost and the state maintained by each node scaling logarithmically with

#### The second second second

Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software remains at each node is examplent in functionality a long hot redundant storage, permanence, selection of nearby server, ananymity, search authentication, and hierarchical naming. Descript this rich set of features, the core countion in most peer-to-neer systems is efficient location of data iteres. The contripeer-to-peer system with frequent node arrivals and departures

The Chord protocol supports just one operation: given a key, it maps the key onto a node. Depending on the application using Chord, that node might be responsible for storing a value associated attigg keys to Chord nodes. Consistent hashing tends to balance load, since each node seceives roughly the same number of keys, "University of California Redulay interaction backalay adu

Authors in peverse alphabetical order

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Sys-tems Center, San Diego, under contract N66001-00-1-8933.

Permission to make digital or hard expires of all or part of this work for personal or characeen new is grazed without the perceluid that copies new not made or durationed for profit or commercial advantage and that copies ben this notices and the full citation on the fort pape. To copy otherwise, its Permission, Report and Servers or Personnellate to Loss, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-30113-411-8010008. 35100.

and involves relatively little movement of keys when nodes join Previous work on consistent hashing assumed that nodes were

aware of most other nodes in the system, making it impractical to scale to large number of nodes. In contrast, each Cheel node needs "rooting" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-mode cystem, each node maintains information only about (When N) other nodes, and resolves all lookups via ()(log N) mensages to other nodes. Chord maintains its routing information as nodes ion and leave the system: with high probability each such event results in no more than O(log<sup>0</sup> N) Three features that distinguish Chord from many other peer-to-

peer lookup protocols are its simplicity, provable correctness, and mouthle nerformance. Chord is simple, routing a key through a se orange of Otlor N) other nodes treard the destination. A Chord node requires information about  $O(\log N)$  other nodes for efficient costing, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will ioin and leave arbitrarily, and consistency of even O(log N) state be correct in order for Chord to guarantee correct (hough slow) corring of essenter. Cherd has a simple algorithm for maintaining on in a dynamic enviro

The cest of this paper is structured as follows. Section 2 compares Chord to related work. Section 3 presents the system model that motivates the Chord motional. Section 4 presents the bare Chord motional and protect several of its momenties, while Section 5 presents extensions to handle concurrent joins and failures. Section 6 demonstrates our claims about Chard's performance through simulation and experiments on a deployed prototype. Finally, we outline items for fining work in Section 7 and summarize our con-

#### 2. Related Work

While Chord many keys onto nodes, traditional name and location services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleaver, the rest of this section assumes a Chord based service that maps keys onto values. DNS provides a host name to IP address mapping [15]. Chord

and the associated IP address representing the value. Chood requires no special servers, while DNS relies on a set of special root



Lecture 1. Introduction September 13, 2012 36/49

# On the paper: Introduction.

Introduction describes more

precisely current state of the

research field, significance of

the paper and relevance to it.

gives some basic assumptions

outlines own achievements.

and limitations, as well as briefly

## Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MT Laboratory for Computer Science chordiftics mit actu http://pdos.ics.mit.edu/chord

Abstract

A fundamental problem that confronts neer-to-neer applications in to efficiently locate the node that stores a particular data item. The paper presents Chowl, a distributed lookup protocol that addresse a key, it maps the key onto a node. Data location can be easi implemented on top of Chood by associating a key with each da item, and storing the key/data stem pair at the node to which key many. Cherd adapts efficiently as nodes ioin and leave system, and can answer queries even if the system is contanto netments show that Chord is scalable with communication and the state maintained by each node scaling logarithmical the number of Chord nodes.

### Introduction

Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software remains at each node is examplent in functionality long hit redundant storage, permanence, selection of nearby servers anonympty search sufpentication and hiersecharal man peer-to-over systems is efficient location of data items. The contripeer-to-peer system with frequent node arrivals and departures The Clovel protocol supports just one operation: given a key,

it maps the key onto a node. Depending on the application using Chood, that node might be responsible for storing a value associated attigg keys to Chord nodes. Consistent hashing tends to balance load, since each node receives roughly the same number of keys,

University of California Barkalay, intrication barkalay adu Anthors in peverse alphabetical order



Permission to make digital or hand copies of all or part of this work for Personations to make signified or hard request of all or part of this work her personal or choracous tas is granted without the provided that copies are not made or durabuted for profits or commercial advantage and that copies bear this notice and the full citation on the farst page. To copy otherwise, to permission, to post an operation of reasonable to inter, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-30113-411-8010008. \$5100.

d involves relatively little movement of keys when nodes join Previous work on consistent hashing assumed that nodes were

aware of most other nodes in the system, making it impractical to scale to have number of nodes. In contrast, each Chord node needs 'ronting" information about only a few other nodes. Because the pouring table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-node cystem, each node maintains information only about (When N) other nodes, and resolves all lookups via O(log N) mensages to other nodes. Chord maintains its routing information as nodes ion and leave the system: with high probability each such event results in no more than  $O(\log^2 N)$ Three features that distinguish Chord from many other peer-to-

peer lookup protocols are its simplicity, provable correctness, and mouthle nerformance. Chord is simple, routing a key through a seorange of Otlog N) other nodes toward the destination. A Chord node requires information about  $O(\log N)$  other nodes for efficient costing, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will ioin and leave arbitrarily, and consistency of even O(log N) state be correct in order for Chard to guarantee correct (hough slow) contine of energy. Cherd has a simple algorithm for maintaining on in a dynamic envir

The cest of this paper is structured as follows. Section 2 comparen Chord to related work. Section 3 presents the system model that motivates the Chord protocol. Section 4 presents the base Chord motocol and proves several of its mometies, while Section 1 presents extensions to handle concurrent joins and failures. Sec tion 6 demonstrates our claims about Chard's performance through simulation and experiments on a deployed prototype. Finally outline items for future work in Section 7 and summarize

#### 2. Related Work

While Cherd many keys onto nodes, traditional name and location services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleaver, the rest of this section assumes a Chord based service that maps keys onto values. DNS provides a host name to IP address mapping [15]. Chord

and the associated IP address representing the value. Chood requires no special servers, while DNS relies on a set of special root



and Technology

Lecture 1. Introduction September 13, 2012 37/49

# On the paper: Introduction.

### Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MT Laboratory for Computer Science chord@lics mit adu http://pdos.ics.mit.edu/chord

### Abstract

A fundamental problem that confronts new to peer applications in to efficiently locate the node that stores a previous per appoint in poper presents Chord, a distributed lookup protocol that addresses a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key'data stem pair at the node to which the key many. Cheed adapts efficiently as nodes join and leave the system, and can answer queries even if the system is contanously channing. Result from theoretical analysis, similations, and ennetments dony that Chord is scalable with communication cost and the state maintained by each node scaling logarithmically with

#### Introduction

Peer-to-peer systems and applications are distributed syste without any centralized control or hierarchical organization, where the software ramains at each node is eggevalent in functionality a long hit redundant storage, permanence, selection of nearby servers anonympty search sufpentication and hiersecharal man ing. Descript this rich set of features, the core countion in most peer-to-neer systems is efficient location of data iteres. The contrientered property and one operation, given a

it maps the sea or

attigg keys to Chord nodes. Consistent hashing tends to balance load, since each node seceives roughly the same number of keys, "University of California Badulay, intrication badulay ada

Anthors in peverse alphabetical order

This research was sponsored by the Definite Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Sys-tems Center, San Diego, under contract N66001-00-1-8933.

Permission to make digital or hard expires of all or part of this work for personal or chromous naw is guarant without the perceluid that copies now not made or durationed for profit or commercial advantage and that copies bene this notices and the full citations on the fort paper. To copy otherwise, its permission, to post an operation of reasonable to inter, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-30113-411-8010008. \$5100.

and involves relatively little movement of keys when nodes join and leave the system Previous work on consistent hashing assumed that nodes were

aware of most other nodes in the system, making it impractical to scale to large number of nodes. In contrast, each Cheel node needs 'ronting" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-node cystem, each node maintains information only about (When N) other nodes, and resolves all lookups via O(log N) mensages to other nodes. Chord maintains its routing information as nodes ion and leave the system: with high probability each such event results in no more than  $O(\log^2 N)$ Three features that distinguish Chord from many other peer-to-

peer lookup protocols are its simplicity, provable correctness, and mouthle nerformance. Chord is simple, routing a key through a se grappe of (blog N) other nodes treard the destination. A Chord node requires information about  $O(\log N)$  other nodes for efficient outing, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will ioin and leave arbitrarily, and consistency of even O(log N) state be correct in order for Chard to guarantee correct (hough slow) contine of energy. Cherd has a simple algorithm for maintaining

The cest of this paper is structured as follows. Section 2 compares Chord to related work. Section 3 presents the system model that motivates the Chord motional. Section 4 presents the bare Chord motional and protect several of its momenties, while Section 5 presents extensions to handle concurrent joins and failures. Section 6 demonstrates our claims about Chard's performance through simulation and experiments on a deployed prototype. Finally, we outline items for fining work in Section 7 and summarize our con-

#### 2. Related Work

While Chord many keys cuto nodes, traditional name and location services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleaver, the rest of this section assumes a Chord based service that maps keys onto values. DNS provides a host name to IP address mapping [15]. Chord

and the associated IP address representing the value. Chood requires no special servers, while DNS relies on a set of special root

The first part of Introduction is hardest part, it always is difficult to find what words/sentence to use in the beginning.

Aalto University School of Science and Technology

## Lecture 1. Introduction September 13, 2012 38/49

## On the paper: Introduction.

## Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MT Laboratory for Computer Science chordiftics mit actu http://pdos.ics.mit.edu/chord/

### Abstract

A fundamental problem that confronts new to peer applications in to efficiently locate the node that stores a particular data item. This paper presents Chowl, a distillated lookus protocol that addresses a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key/data item pair at the node to which the key many. Cheed adapts efficiently as nodes join and leave the system, and can answer queries even if the system is contanuously channing. Result from theoretical analysis, significant, and exnetments dony that Chord is scalable with communication cost and the state maintained by each node scaling logarithmically with the number of Cheed nodes

#### 1 Introduction

Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software remains at each node is examplent in functionality a long hit redundant storage, permanence, selection of nearby servers anonympty search sufpentication and hiersecharal man peer-to-neer systems is efficient location of data iteres. The contriof this paper is a scalable protocol for lookup in a dynamic

The Clovel protocol supports just one operation: given a it maps the key onto a node. Depending on the application usin and, that node might be responsible for storing a value associa . Chord men a variant of consistent l

anigs keys to cannot note. Comments assume which to balance load, since each node receives roughly the same number of keys, University of California Barkalay, intrication barkalay adu

Anthors in peverse alphabetical order

This research was sponsored by the Definite Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Sys-tems Center, San Diego, under contract N66001-00-1-8933.

Permission to make digital or hard expires of all or part of this work for personal or chromous naw is guarant without the perceluid that copies now not made or durationed for profit or commercial advantage and that copies bene this notices and the full citations on the fort paper. To copy otherwise, its permission, to post an operation of reasonable to inter, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-30113-411-8010008. \$5100.

### and involves relatively little movement of ke

Previous work on consistent hashing assumed that nodes were rare of most other nodes in the system, making it impractical to e to large mucher of nodes. In and the other party Berning routing table is distributed, a node resolves the hash function communicating with a few other nodes. In the steady state, (When N) other nodes, and resolves all lookups via O(log N) mennodes ion and leave the system: with high probability each such

Three features that distinguish Chord from many other peer-to peer lookup protocols are its simplicity, provable concerness, and

question of the second seco be correct in order for Chord to guarantee correct (floored contine of energy. Cherd has a simple algorithm for maintaining

that increments in a systemic environment. The set of the steps is structured as follows. Section 2 com-parest Coord of Lated work. Section 3 presents the system model that more on the Chord protocol. Section 4 presents the base Chord

bandle concurrent joins and failures. Sec Descent and presents extended to an an an and transfer. Sec-tion 6 demonstrates on claims about Chard's performance through simulation and experiments on a deployed prototype. Finally, we outline items for fature work in Section 7 and supermative our con-

### 2. Related Work

While Chord many keys cuto nodes, traditional name and location services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleaver, the rest of this section assumes a Chord based service that maps keys onto values. DNS provides a host name to IP address mapping [15]. Chord

and the associated IP address representing the value. Chord requires no special servers, while DNS relies on a set of special root Normally, the first sentence of each paragraph is general sentence, which includes all the knowledge, while the remaining parts are details. which provides the around for the first sentence.

When you start writing an article, you can write down a set of general sentences and organize the article structure based on them, and later provide specific details about those.



Lecture 1. Introduction September 13, 2012 39/49

## On the paper: Related work.

## Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan MT Laboratory for Computer Science chordiftics mit ach http://pdos.ics.mit.edu/chord/

Abstract

A fundamental problem that confronts new to peer applications in a key, it maps the key onto a node. Data location can be easily implemented on top of Chord by associating a key with each data item, and storing the key/data item pair at the node to which the key many. Chord adapts efficiently as nodes join and leave the system, and can answer queries even if the system is continuously channing. Results from theoretical analysis, simulations, and exnetments dony that Chord is scalable with communication cost and the state maintained by each node scaling logarithmically with

#### 1 Introduction

Peer-to-peer systems and applications are distributed systems without any centralized control or hierarchical organization, where the software remains at each node is examplent in functionality a long hot redundant storage, permanence, selection of nearby servers, anonymity, search, authentication, and hierarchical man peer-to-neer systems is efficient location of data iteres. The contripeer-to-peer system with fbessent node arrivals and departures

The Clovel protocol supports just one operation: given a key, it maps the key onto a node. Depending on the application using Chord, that node might be responsible for storing a value associated attigg keys to Chord nodes. Consistent hashing tends to balance load, since each node seceives roughly the same number of keys, University of California Backalay, intrication backalay ada

Authors in reverse alphabetical order

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Sy tems Center, San Diego, under contract N66001-00-1-8933

Permission to make digital or hard copies of all or nart of this work Permanents to make signila or hard report of all or part of this rooks perconal or characous tas is granted without the provided that copies not made or characous tas is granted without divinitigs and that copies bear this notice and the full citetion on the first page. To copy otherwise permission, Report and Servers or Personnellate to Loss, requires press permission and/or a fee. 350COMM/31, August 27-31, 2001, San Diegn, California, USA, Conventer 2001 ACM 1-35113-411-8010008. \$5100.

and involves relatively little movement of keys when nodes join

Previous work on consistent hashing assumed that nodes were aware of most other nodes in the system, making it impractical to scale to large number of nodes. In contrast, each Cheel node needs 'ronting" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-node cystem, each node maintains information only about (When N) other nodes, and resolves all lookups via O(log N) mensages to other nodes. Chord maintains its routing information as nodes ion and leave the system: with high probability each such event results in no more than  $O(\log^2 N)$ Three features that distinguish Chood from many other peer-to-

peer lookup protocols are its simplicity, provable correctness, and mouthle nerformance. Chord is simple, routing a key through a se orange of Otlor N) other nodes treard the destination. A Chord node requires information about  $O(\log N)$  other nodes for efficient conting, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will ioin and leave arbitrarily, and consistency of even O(log N) state be correct in order for Chord to guarantee correct (hough slow) contine of energy. Cherd has a simple algorithm for maintaining

The cest of this paper is structured as follows. Section 2 compares Chord to related work. Section 3 presents the system model that motivates the Chord motional. Section 4 presents the bare presents extensions to handle concurrent joins and failures. Secsimulation and experiments on a deployed prototype. Finally, we outline items for fining work in Section 7 and summarize our contributions in Section 8.

#### 2. Related Work

While Chord many keys cuto modes, traditional name and lo cation services provide a divect mapping between keys and val-ues. A value can be an address, a document, or an arbitrary data item. Chord can easily implement this functionality by storing each key/value pair at the node to which that key maps. For this reason and to make the comparison cleater, the rest of this section assumes a Chood based service that maps keys onto values. DNS reprode a host maps to P address marging [15]. Chood

and the associated IP address percentations the value. Chood requires no special servers, while DNS relies on a set of special root Related work section contains the most significant knowledge a researcher have discovered concerning the topic.

It is the starting point of any article or any scientific work. You can start this section even during when you still study your field. It shows that you well aware of the topic.



Lecture 1. Introduction September 13, 2012 40/49

# On the paper: Finalizing the paper.

Figure 12: Lookup latency on the Internet prototype, as a function of the total number of nodes. Each of the ten physical sizes runs multiple indexendent coules of the Chord ande software.

copies of the Chool software at each site. This is different from manning  $O(\log N)$  without nodes at each site to provide good land balance, rather, this instention is to measure have well our implementation scales even though we do not have more than a small number of deployed modes.

The end matter of tasks shown in Figure 11, and physical meres 16 Colo (1000) spin tasks (1000) spin

The lesson from Figure 13 is that lookup latency gows slowly with the total number of nodes, confirming the simulation results that demonstrate Chord's scalability.

#### 7. Future Work

Based on our experience with the prototype mentioned in Section 6.6, we would like to improve the Chord design in the followmentation

There are enough its no to perform mechanisms to head participants in sings, such mign code loggest location constraints to the subhitzment procedure. One way to check global constraints; it for each nucle or to periodized in a dore moders to do a chall location for a " the location" does not be a substantiation of the substantiant. This ways to dona that lacestadap in the every mode to lance of the sums and in our distant substantiation where a substantian by the for moder and in our distant substantiation where a substantian by the for moder and in our distant substantiation in the substantiant of the substantiant of the substantiant lacestadap in the every mode to lance of the sums meconanced in the part, if a particular form, the reactions are in an exercition are life for the inducible code for dotte or deep neutrino.

particulate and a set of the set

that they are not seeing a globally consistent view of the Chord ring.

An attacker could target a particular data item by inserting a node into the Chord may with an ID immediately following the item's law, and having the node return more when added to reture the data. Requiring (and checking) that nodes use IDs derived from the SIA-1 bash of their IP addresses makes this struck hardes: Dees log?. We messages per locality may be no many for some

The start of a start of the st

A different approach to improving lockup listney might be to use server relevant. Each finger table entry could point to the first k nodes in flat entry's interval on the ID ring, and a node could measure the server's delay to each of the k nodes. The k nodes are generally equivalent for purposes of lockup, so a node could formal lockups to the case with lower delay. This special would be most effective with recurrice Chaol lockup, in which the nodes measuring the delays is also the node forwarding the lockup.

#### 8. Conclusion

Many distributed per-to-per applications used to determine the node that stress and the item. The there determines a data item. The there approaches (see the data) paraziver, general as legs (it determines the node responsible for streing the legs 'n streing the legs 'n subset of the size of the stress of the legs 'n subset of the size of

Attention features of Clored included in simplicity, provide cortentenes, and provide performance even in the fibe of concoursent node arrowing and departments. In continues to fanction consectly, alber at 6 ograded performance, where a node's information, and oppormental results confirm that Clared sciences in the number of nodes, recover; from large matchers of simultaneous node failures of joins, and assume must londing correctly even sharing sectorary joins, not assume must londing correctly even sharing sectorsectors.

We believe that Chord will be a valuable component for peerto-peer, large-scale distributed applications such as cooperative file during, sime-dured randiable strange systems, durinfund under for document and service discovery, and large-scale distributed computing platforms.

#### Acknowledgments

We thank Fanak Dabek for the measurements of the Chord prototype described in Section 6.6, and David Andersen for setting up the testbed used in those measurements.

### 9. References

- ANDERSEN, D. Raubant ovaday networks. Master's thesis, Department of EECS, MIT, May 2001.
- [2] BAREER, A., AMADE, E., BALLINTIN, G., KUZ, I., VERLARE, B. 199 No. 199 (1997) 1 (1997)



Lecture 1: Introduction September 13, 2012 41/49

# On the paper: Discussion.

In Discussion or Future Work section you touch the important questions related to the work, which though stayed out of the scope of the paper are still important to mention explicitly.

Additionally, here You can discuss extensions, limitations and problems of the paper, and how to improve or solve them in the future.

You can strengthen own paper, if you add in this section solutions for possible flaws of your work. This will show to the reviewers that you aware of these limitations. and at least know how to extend the work to overcome them, or why those scenarios are insignificant.



Figure 12: Lookup latency on the Internet prototype, as a fution of the total number of nodes. Each of the ten physical sirum multiple independent cupies of the Chood node software

copies of the Chood software at each site. This is different f running  $O(\log N)$  virtual nodes at each site to provide good b balancy: rather, the intentions in to measure heave well our implementation scales even though we do not have more than a small run of deployed nodes. For each number of nodes shown in Figure 13, each plays

where the CC set to denote the time of got 4.2, then prove one. The proph gives the setup. In S + 0.0, the S + 0.0, then of local particle is the setup. In S + 0.0, the S + 0.0, prove of local particle is the setup. In S + 0.0, the S + 0.0, prove of local particle is the setup. In S + 0.0, the S + 0.0,

### 7. Future Work

Based on our experience with the prototype mentioned in Section 6.6 we would like to improve the Chord design in the follow-

Cased eccentraly has no specific mechanisms to heat particined, many, such may occupy hope (host) constraints to be subhlation procedure. One way to check global constraints; in form each ands or in perstedicity is able to also the source of the locking for c<sup>-1</sup>, the locking due to at yield ands or , these may be a particule. This way so obtain that have been provided and the locking for c<sup>-1</sup>, and the locking of the locking source of the locking for c<sup>-1</sup>, and the locking source of the locking of the models to mannata hoge senses (at a family and the locking for models not encounted in the participant of the locking for the models not a statistication of the locking for the models one in more parameters for the locking relation provided and the locking for the model of the locking of the locking for the models one in more parameters of the locking relation provided and the locking for the model of the locking of the locking for the locking for the locking of the locking for the locking set of the locking for the locking for the locking for the locking set of the locking for the locking for the locking of the locking set of the locking the locking for the locking for the locking set of the locking the locking for the locking for the locking set of the locking the locking for the locking for the locking set of the locking the locking for the locking for the locking set of the locking the locking for the locking for the locking of the locking set of the locking the locking for the locking for the locking for the locking set of the locking the locking for the locking set of the locking the locking for the

As unintoint to the second range Association of the proton on incorrect view of the Chool range. Association protocol and the is being used to locate is cryptographically authenticated, this is a threat to availability of data anther fram to authenticity. The same approach used above to detect partitions could help victims realized. that they are not seeing a globally consistent view of the Chord

An attacker could mayet a particular data item by inserting a node into the Chord ring with an ID immediately following the item? way, and having the node return more when atted to retrieve the data. Requiring (nod beeking) that nodes use IDs derived from the SHA-1 bank of their IP addresses makes that attack hards:

These  $\log N$  messages per looking may be too many for some splications of Costan, quescidity if each manage must be sust in a madem informed box. Instead of placing in fingers at dismost, a start of the splication of Costan at the power of the splication of the sp

A different approach to improving leaking alterncy might be to us server selection. Each funger twide earny could point to the first k nodes in that entry's introval on the ID ring, and a node could measure the asserved delay to each of the k andes: the k index are generally equivalent for purposes of leaking, so a node could be mast effective with recurrice (hand leaking, in which the node measuring the delay's in alto the node forwarding the lockup.

### S. Conclusion

Many distributed peer-to-peer applications need to determine the node that stress a state intern. The Cheel protocol relevant the challenging geoblem in discrimination dimanne. It effects a power charge the period of the stress of the stre

Amounts futures of Cheet include its singleicy, provide correctors, and provide performance even in the fibe of concoursest node arrowin and department. It containes to futation consetly, allet a degraded performance, where a node's information, and sparmental neuritic confirm that Cheet arrow only information, and experimental neuritic confirm that Cheet arrow well with the number of nodes, recovers from large matchers of simultaneous node finance and joins, and answers must londong correctly even sharing recovmental neuritic contern must londong correctly even sharing recoving joins, and answers must londong correctly even sharing recov-

We believe that Chord will be a valuable component for peerto-peer, large-scale distributed applications such as cooperative fidedaring, time-dured straight strategy system, duringted induced for document and service discovery, and large-scale distributed computing platforms.

#### Acknowledgments

We thank Frank Dabek for the measurements of the Chord prototype described in Section 6.6, and David Andersen for setting up the testbed used in those measurements.

### References

 ANDERED, D. Rechest overlay network. Master's thesis, Department of EECS, MIT, May 2001. http://nms.los.mit.edu/projects/com/.

[2] BARRIER, A., AMADE, E., BALLINTIN, G., KUI, I., VEREARE, F., VAN DER WUR, I., VAN STEEN, M., AND TANENBAUM, A.



Lecture 1: Introduction September 13, 2012 42/49

# On the paper: Conclusion.

N In to He have been

Figure 13: Lookup latency on the Internet prototype, as a function of the total number of nodes. Each of the ten physical sites runt multiple independent cupiet of the Chord node toftware.

copies of the Chood software at each site. This is different from running (Nice N) virtual nodes at each size to provide good load tation scales even though we do not have more than a small number

For each number of nodes shown in Figure 13, each physical site issues 16 Chord lookups for madonaly chosen keys one-byone. The graph plots the median, the 5th, and the 95th percentil depending on number of nodes. For the case of 180 nodes, a type cal lookup involves five two-way message exchanges, four for the Chord lookup, and a final message to the successor node. Typical round-trip delays between sites are 60 milliseconds (as measure by ping). This the expected lookup time for 180 nodes is about 300 milliocends which is clear to the reserved median of 255 The low 5th percentile latencies are caused by lookups for keys close (in ID space) to the querying node and by query hops that reby lookups whose hops follow high delay paths.

The lesson from Figure 13 is that lookup latency grows slowly with the total number of nodes, confirming the simulation results

#### 7 Future Work

Based on our experience with the prototype mentioned in Section 6.6, we would like to improve the Chord design in the follow-

Cheed currently has no specific mechanism to heal partit rings; such rings could appear locally consistent to the stabilizatiprocedure. One way to check global consistency is for each node to periodically ask other nodes to do a Chord lookun for ----the lookup doet not yield node a, there may be a partition. That will only detect partitions whose nodes know of each other. One way to obtain this knowledge is for every node to know of the same small set of initial nodes. Another approach might be for nodes to maintain long-term memory of a random set of nodes they have encountered in the past; if a partition forms, the random sets in one

A malicious or burry set of Chord unticipants could present an incorrect view of the Chord ring. Assuming that the data Chord is being used to locate is cryptographically suthenticated, this is a approach used above to detect partitions could help victims realize that they are not seeing a globally consistent view of the Chord

An attacker could target a particular data item by inserting a node into the Chord ring with an ID immediately following the item's key, and having the node return errors when asked to retrieve the data. Requiring (and checking) that nodes use IDs derived from the SRA-1 hash of their IP addresses makes this attack harder. Even log N messages per lookup may be too many for some

applications of Chord, especially if each message must be sent to a random internet host. Instead of placing its fingers at distances that are all powers of 2. Chord could easily be channed to place its enery to 1/(1+d) of the original distance, meaning that here. hops would suffice. However, the number of fingers needed would

A different approach to improving lookup latency might be to use server selection. Each finger table entry could point to the first measure the network delay to each of the k nodes. The k nodes are penerally emphased for purposes of lookun, so a node could forward lookups to the one with lowest delay. This approach would be most effective with recurrine Chord lookups, in which the node

#### Conclusion

Many distributed peer-to-peer applications need to determine the node that stores a data item. The Chord restocol solves this challenging problem in decentralized manner. It offers a power fal primitive: given a key, it determines the node responsible for storing the key's value, and does to efficiently. In the steady state, for only about (7/10x N) other nodes, and resolves all lookups via mation for nodes leaving and joining require only  $O(\log^T N)$  met-

Attractive features of Chord include its simplicity, provable correctaest, and provable performance even in the face of concurren partially correct. Our theoretical analysis, simulations, and exper mental results confirm that Chord scales well with the number of nodes, recovers from large numbers of simultaneous node failures and joins, and answers most lookups correctly even during recov-

. We believe that Chord will be a valuable component for peerto-peer, large-scale distributed applications such as cooperative file sharing, time-shared available storage systems, distributed indices for document and service discovery, and large-scale distributed computing platforms.

#### Acknowledgments

We thank Frank Dabek for the measurements of the Chard prototype described in Section 6.6, and David Andersen for setting up the testbed used in those measurement

### 9 References

- ANDERSEN, D. Randoms overlay networks. Master's thesis, Department of EECS, MIT, May 2001.
- 121 BARRER, A., AMALS, E., BALLINTIN, G., KUZ, I., VERSAIL,

Conclusion is the section for the final words about the work done. Here you almost repeat what was mentioned in Abstract and Introduction

Naturally, if your paper reveals something new and important. You will try to put it as close as possible to the beginning of the paper: into Abstract and Introduction

Here, however, you just list all those main findings of your paper and probably the impact of it onto the corresponding field.

> Lecture 1: Introduction September 13, 2012

43/49



## On the paper: Acknowledgement.

N In to He have been

Figure 13: Lookup latency on the Internet prototype, as a function of the total number of nodes. Each of the ten physical sizes runs multiple indexendent coules of the Chord ande software.

copies of the Chool software at each site. This is different from manning  $O(\log N)$  without nodes at each site to provide good land balance, rather, this instention is to measure heaved low implementation scales even though we do not have more than a small number of deployed modes.

The end muscle of tasket down in Figure 11, each physical masses 11 CoLC and tasket down the prior addy. The control of the strength of the models of the strength of collection physics. The media is lease, ranger than 110 is 23 min. The strength of the strength of the strength of the strength of holding beings the strength of the large strength of the large strength of the stren

The lesson from Figure 13 is that lookup latency gows slowly with the total number of nodes, confirming the simulation results that demonstrate Chard's scalability.

#### 7. Future Work

Based on our experience with the prototype mentioned in Section 6.6, we would like to improve the Chard design in the follow-

Core controls has no specific mechanism is built participant ingr, such migr code happe is colif constraints in the industruprovedness. One way to check global constraints; is for each nucler to periodicity and other nodes to do a Chall lokay for c: if the locking does not yield nade ..., there may be a partice. This way is obtain that laceshade is if or every node to know of dens differ until or of cinetal nodes. Another approach to fails not of data and mill or of cinetal nodes. Another approach to fails node to support of the nodes of the start of the node of the node of the node encountered in the part of a particular form, the readom with a net partition no cilled to the lock of the opport of the nodes of the node encountered in the part of a partition form, the readom with a net partition no cilled to the lock of the opport of the net of the nodes.

A malicinum or barger set of Cool participant could present an incorrect view of the Chord ring. A maning that the data Chord is being used to locate is cryptographically autoheticated, this is a threat to resultability of data rather than is anthenticated, this is a spareock used show to detect participants could help within steakle that they are not seeing a globally consistent view of the Chord ring.

As attacker could target a particular data item by insetting a node into the Cherd ring with an ID immediately following the item? ite key, and having the node return more when aided to retrieve the data. Requiring (and checking) that nodes use IDs derived from the SBA-1 bath of their IP addresses makes that partick hards:

The top of a line of the set of

A different approach to improving lookup littlessy might be to use server relection. Each flagger thick earny could point to the flast k modes in flast entry's interval on the D ring, and a node could measure the averació delay to each of the k modes. The k nodes are generally equivalent for purposes of lookup, so a node could be most effective with measure the directive charger beneral to lookup.

#### 8. Conclusion

Many diministral per-to-per applications need to determine the node that stress a data issue. The Cherd protocol selects this challenging goolsam is descentihilated names. In efferts a power laparativer general rady of determinist the sole responsible for a stress of the sole stress of the sole stress of the sole rady and trivial stress, each node maintains returning information of only obset (17% s) of the nodes, and needers all lockings the Object N messages to other nodes. Updates the iterating information of the node heating and pointing require and or (12% N) metmory for node heating and pointing require and (17% N) metsages and the nodes of the node

Amories fentures of Chord include in singleizy, growth cortextens, and provide performance even in the face of concurrent node arrowin and department. It contains to fraction consetly, a best of departed performance, where a node's information, used sparsing performance there are node's information, and sparsmental neutrin contribution for distantiations, and sparsmental neutrin contribution will with the number of nodes, recovers from large mundlers of simultaneous node failures and joins, and answers much loboling correctly even hanging recovmental neutrino correctly even hanging recov-

cr. We believe that Chord will be a valuable component for peer to spen. Luges-calls distributed applications such as cooperative file during, same-durand vanishe burge systems, durintered indices for document and service discovery, and large-cale durinbuted computing platforms.

#### Acknowledgment

We thank Frank Dabek for the measurements of the Chord prototype described in Section 6.6, and David Andersen for setting up the testbed used in those measurements.

### 9. References

- ANDERSEN, D. Realisest overlay networks. Master's thesis, Department of EECS, MIT, May 2001.
- [2] BAREER, A., AMADE, E., BALLINTUN, G., KUZ, I., VERLAIE, F. MANDER WITE 1. VAN STREEM, AND TANDOLATING A.

### Acknowledgements section is for the mentioning people and grants, which helped you with preporation of the paper.

You can mention here proof readers, people who helped you and sometimes guide.

You don't need (and have no obligation to put all those people into author list, though sometimes people mistakenly put them there) to put everyone as your co-author. This is the right place.

Aalto University School of Science and Technology

### Lecture 1: Introduction September 13, 2012 44/49

## On the paper: References.

N States of Local division

Figure 13: Lookup latency on the Internet prototype, as a function of the total number of nodes. Each of the ten physical sizes runs multiple indexendent coules of the Chord ande software.

copies of the Chool software at each site. This is different from manning  $O(\log N)$  without nodes at each site to provide good land balance, rather, this instention is to measure heaved low implementation scales even though we do not have more than a small number of deployed modes.

The end muscle of tasket down in Figure 11, each physical masses 11 CoLC and tasket down the prior addy. The control of the strength of the models of the strength of collection physics. The media is lease, ranger than 110 is 23 min. The strength of the strength of the strength of the strength of holding beings the strength of the large strength of the large strength of the stren

The lesson from Figure 13 is that lookup latency goves slowly with the total number of nodes, confirming the simulation results that demonstrate Chard's scalability.

#### 7. Future Work

Based on our experience with the prototype mentioned in Section 6.6, we would like to improve the Chord design in the followmentation

A malicious of unique your forced or other products and a second second

that they are not seeing a globally consistent view of the Chord ring.

An attacker could target a particular data item by inserting a node into the Chord mag with an ID immediately following the tem is hey, and having the node return more when added to retive the data. Requiring (and checking) that nodes use IDs derived from the SIA-1 bash of their IP addresses makes this strack harder break log?, messages per looking may be too many for some

The top of the starting of th

A different approach to improving looking alterncy might be to us server relevant. Each farger twice earny could possible to the first k nodes in flat entry's interval on the ID ring, and a node could measure the neuronic dicty to each of the k nodes: In the k nodes are generally equivalent for purposes of looking, so a node could forward lookings to the case with lower tellery. This approach would be mast effective with recurring CAM looking, in which the node meaning the doubly in allot the node forwarding the looking.

#### 8. Conclusion

Many dimbined per-to-per applications need to determine the node that stress and the ion-fluctuation as a data users. The Cherd protocol science of hypothyper-periods for a log-color dimbined parameter gives a log-ly of determination the node respectively for a log-color dimbined parameter gives a log-ly of determination of the log-color dimbined parameter gives a strength science of the log-log-color dimbined parameter gives and log-log-color dimbined parameter gives a strength science of the log-log-color dimbined parameter and log-ly (h) restriction of the log-log-color dimbined parameter and log-ly (h) and log-color dimbined parameters and log-color dimbined parameters and log-ly (h) and log-color dimbined parameters and log-color dimbined para

Amories fentures of Chord include in simplicity, provide contretens, and provide performance even in the face of concurrent node arrowin and departures. It containes to fraction consetly, a lot a degraded performance, when a node's information, and eparmental neutrin confirm that Chard relations, and equimental neutrin confirm that Chard relation will with the number of nuclei, recovers from large matchers of simultaneous node finance and joins, and aureem num biologic correctly even sharing recornel joins, and aureem num biologic correctly even sharing recor-

We believe that Chord will be a valuable component for peerto-peer, large-scale distributed applications such as cooperative the during time-duried straight transpectyteme, distributed indices for document and service discovery, and large-scale distributed computing platforms.

### Acknowledgments

We thank Frank Dabek for the measurements of the Chord prototype described in Section 6.6, and David Andersen for setting up the testbed used in these measurements.

#### 9. References



Reference section is for all relevant work you mention in the text. Most of the references goes from the Related work and Introduction section.

Never, include as references papers which you do not mention at all, or mention "not-enough". Some reviewers use reference list to understand how good you studied your own field. Try to be up-to-date here also.



Lecture 1: Introduction September 13, 2012 45/49

# Time usage

Try to optimize your time. You paper should be multi-sided, address different aspects, different methods. Find time for every side, even for self-criticism.

## Remember:

- Conferences happens all the time, but try to aim at some conference with fixed date.
- Put as much deadlines for yourself as possible. This help you to organize your time.
- All documented small outcomes that are collected in one work produce big work.
- Having all pieces on the paper put your ideas in order and allow not to lose results.
- Find time for reading (and studying), find time for writing.



# **Tools to use**

Researcher has a various set of tools to use for all the Research methods. They are not emerging as often as new papers, but still remember to keep track of the current tools. They are created to help you.

There are a lot of tools for Data communication researcher:

- Simulators (NS-2, NS-3, OverSim, OMNeT++).
- Development tools (Eclipse, Visual Studio, EMacs).
- Document preparation systems (WYSIWYG) (T<sub>E</sub>X, MS Word, OpenOffice).
- Networking tools (BRITE, Wirshark).
- Data Analysis tools (R, GnuPlot).
- Mathematical labs (MatLab, Mathematica).
- Networking labs (PlanetLab, OneLab).



# Other methods to increase own understanding

There is a set of other methods to increase the quality of a research:

- attend Computer Science courses and border Science courses.
- attend public presentations, especially with presenters from other groups and Universities.
- take part in the review process.

What else?

Non-formal collaboration!!!



Lecture 1: Introduction September 13, 2012 48/49

# **Questions and Comments?**

Thank you.



Lecture 1: Introduction September 13, 2012 49/49