
Graduate Committee
- Prof. Anura P. Jayasumana (Advisor)
- Prof. V. Chandrasekar
- Prof. Daniel F. Massey
- Prof. Indrajit Ray

Dilum Bandara
dilumb@engr.colostate.edu
Contributions

Propose a peer-to-peer based approach to enable the collaboration of group of heterogeneous, dynamic, & distributed resources in a scalable & efficient manner

Developed resource discovery, caching, & distributed data fusion solutions that are more suitable for collaborative P2P systems by characterizing real-world resource, query, & user behavior
Outline

• Motivation
• Problem statement
• Multi-attribute resource & query characteristics
• Resource & query aware resource discovery
• Multi-attribute resource & range query generation
• Community-based caching
• NDN for DCAS
• Summary & future work
Collaborative Peer-to-Peer Systems

- Advances in Web 2.0, high-speed networks, cloud computing, & social networks
- P2P systems will play an even greater role in distributed resource collaboration
- Diverse peers bring in unique resources & capabilities to a virtual community to accomplish something big
- Scalable alternative to Distributed Collaborative Adaptive Sensing (DCAS), Internet of Things, cloud & opportunistic computing, etc.
Collaborative Adaptive Sensing of the Atmosphere (CASA)

- Distributed Collaborative Adaptive Sensing (DCAS) system
- CASA aggregates groups of resources as & when needed
  - 10,000 radars to cover U.S.
  - High data rate, real-time, heterogeneous, multi-attribute, dynamic, & distributed
  - Dedicated & reliable resources
Global Environment for Network Innovations (GENI)

- Collaborative & exploratory platform for innovation
- Aggregating groups of resources across multiple administrative domains
  - Dedicated & reliable resources

- Sensors
  - Cameras
  - Sensors mounted on busses
  - Micro weather stations
  - Radars

- Processing & storage
  - Amazon EC2
  - Amazon S3

- Networks
  - Internet2
  - Emulab
  - BEN dark fibers
Community (P2P) Cloud Computing

• Resource aggregation within datacenters
  – Data intensive cloud computing
  – Encryption, business logic, & scientific algorithms
  – Storage, GPUs, FPGAs
  – Virtual networks in/out & within cloud
    • Sensors can’t be inside a datacenter

• Community as a datacenter
  – Resourceful peers & home servers
  – Aggregation of bandwidth at edge
  – Users govern themselves & hold data
  – Monetary & non-monetary benefits
  – Voluntary & unreliable resources
Community (P2P) Cloud Computing

- Resource aggregation within datacenters
  - Data intensive cloud computing
  - Encryption, business logic, & scientific algorithms
  - Storage, GPUs, FPGAs
  - Virtual networks in/out & within cloud
    - Sensors can’t be inside a datacenter

- Community as a datacenter
  - Resourceful peers & home servers
  - Aggregation of bandwidth at edge
  - Users govern themselves & hold data
  - Monetary & non-monetary benefits
  - Voluntary & unreliable resources
Problem Statement

• Motivation
  – CASA, GENI, & cloud computing need to aggregate heterogeneous, multi-attribute, & dynamic groups of resources that are distributed
  – Very little is known about their characteristics
  – Existing solutions rely on many simplifying assumptions
    • Few attributes, i.i.d. attributes, attribute values ~uniform/Zipf’s, large domains, very specific queries, ignore dynamic attributes

• Goal
  – Develop better resource discovery & distributed data fusion solutions & necessary tools, while characterizing real-world resources, queries, & user behavior
  – Empower peers to engage in greater tasks beyond capabilities of individual peers
    • Enhanced performance, efficiency, scalability, & resource utilization
Outcomes

1. Detailed analysis of real-world resource, query, & user characteristics, & their impact on P2P-based resource discovery – CCNC ‘12 [6], AICCSA ‘11 [7], [4], [12]


3. Tool to generate large synthetic traces of multi-attribute resources & range queries – GLOBECOM ’11 [8], [13]


5. Demonstrated applicability of Named Data Networking (NDN) for Distributed Collaborative Adaptive Sensing (DCAS) systems such as CASA – [10]
Resources & Queries

• Multi-attribute resources
  – Computing, storage, network, sensors, etc.
  – Static – CPU speed, no of CPU cores, Doppler radar, sensor range
  – Dynamic – Free CPU, memory, bandwidth, sensing frequency

\[
r = \left( a_1 = v_1, a_2 = v_2, \ldots, a_i = v_i \right)
\]

\[
r = \begin{cases} 
  CPUSpeed = 2.4 \text{ GHz}, \text{NumCores} = 2, \text{CPUArchi} = "x86", \text{CPUFree} = 53\% \\
  MemoryFree = 1071\text{MB}, \text{OS} = "Linux_2.6.31", \text{Application} = "NS 3.01"
\end{cases}
\]

• Multi-attribute range queries

\[
q = \left( n, a_1 \in [l_1, u_1], a_2 \in [l_2, u_2], \ldots, a_i \in [l_i, u_i] \right)
\]

\[
q = \begin{cases} 
  n = 5, \text{CPUSpeed} \in [2.0\text{GHz}, \text{MAX}], \\
  \text{MemoryFree} \in [256\text{MB}, 512\text{MB}], \text{OS} = "Linux_2.6.32"
\end{cases}
\]
Multi-Attribute Resource & Query Characteristics [7, 12]

- Datasets – PlanetLab, SETI@home, GCO grid, & CSU
- Real-world resource & query characteristics diverge substantially from conventional assumptions
  - Few attributes → Many attributes
  - i.i.d. → Complex correlation patterns
  - Uniform/Zipf’s → Different marginal distributions, highly skewed
  - Large domains → Small domains for some attributes
  - Ignore dynamic attributes → Most popular, change rapidly
  - Very specific queries → Less specific queries
How These Characteristics Will Affect Resource Discovery?

- Evaluate fundamental design choices for resource discovery
- Used node & query traces from PlanetLab

Centralized $O(1)$

Unstructured P2P $O(hops_{\text{max}})$

Structured P2P – Distributed Hash Table (DHT) $O(\log N)$

Superpeer $O(hops_{\text{max}})$
How These Characteristics Will Affect Resource Discovery?

- Evaluate fundamental design choices for resource discovery
- Used node & query traces from PlanetLab

Clock speed
Memory
Bandwidth
Design Choices for P2P-Based Resource Discovery – Performance Analysis [6, 12]

- Real-world queries are relatively easier to resolve
- Ring-based designs
  - Advertising & query cost – $O(A_{\text{Dynamic}})$ & $O(N)$
  - Load balancing problem
Outline

• Motivation
• Problem statement
• Multi-attribute resource & query characteristics
• Resource & query aware resource discovery
• Multi-attribute resource & range query generation
• Community-based caching
• NDN for DCAS
• Summary & future work
Resource & Query Aware Resource Discovery [3]

- Ring-based resource discovery solutions
  - Pros – Scalable & performance guarantees
  - Cons – High query ($O(N)$) & advertising cost, & unbalanced load
    - Conventional solutions assume $D_i \gg N$
    - Add more nodes to balance load $\sim R/N & \sim Q/N$
- Domain of some attributes is small $D_i \ll N$
  - E.g., CPU cores, CPU architecture, & OS
- Less specific queries
  - Not useful to advertise even attributes with large $D_i$ at high resolutions
    - E.g., Free CPU 40-100%, Free Disk 5-1000 GB
  - Effectively, $D_i \ll N$
- $N = \max(D_i)$
  - How to reduce $N$ while balancing load?

\[
C_{\text{Query}}^q = \sum_{i \in \mathcal{Q}} \left( h_i^q + \left\lfloor \frac{(u_i - l_i)}{D_i} N \right\rfloor - 1 \right)
\]
Heuristic 1 – Prune Nodes With Lower Contribution

Nodes have fixed resource index & query capacity

a) Remove $c \rightarrow$ Reduce query cost $Q^c_{Out} = 0$
   - Can $b$ or $d$ accept any resources indexed at $c$?
   - $d$ is preferred as no changes are required to overlay network

b) Remove $a$, $b$, or $d \rightarrow$ Reduce query cost $Q^i_{Out} < Q^i_{Thr}$
   - Can neighbors accept resource index & query load?
Heuristics 2 & 3 – Key Transfer

- Nodes are already contributing & overloaded

- Heuristic 2
  - $i$ is overloaded
  - Move keys/resources to successor or predecessor – If it can accept
  - Successor is preferred
  - Minor changes to overlay

- Heuristic 3
  - $i$ is overloaded & successor & predecessor not willing to accept load
  - Add new successor or predecessor – Load must not exceed capacity of a node
  - Successor is preferred
  - Some changes to overlay
Heuristics 4 & 5 – Replication & Fragmentation

• Heuristics 2 & 3 will fail if load is too much for a single node

• Heuristic 4
  – Query load is too high
  – Add new node & replicate index
  – Don’t increase query cost
  – More changes to overlay

• Heuristic 5
  – Resource index is too large
  – Add new node & fragment index
  – Rarely increase query cost
  – More changes to overlay

In practice, nodes can index many resources & answer many queries/second → Cliques are not large
Resource & Query Aware Resource Discovery – Performance Analysis

- Each heuristic addresses a specific problem
- More efficient & load balanced solution when all 5 heuristics are combined
  - Work with both single & multi-attribute resources
Outline

- Motivation
- Problem statement
- Multi-attribute resource & query characteristics
- Resource & query aware resource discovery
- Multi-attribute resource & range query generation
- Community-based caching
- NDN for DCAS
- Summary & future work
ResQue – Resource & Query Generator [8, 13]

- Large-scale performance studies need large datasets
  - Neither practical nor economical to capture large datasets at high resolution

- Generate large synthetic traces using information gathered from small real-world traces
  - Resources
    - Large no of resources, many attributes, & temporal changes
    - Vectors of static attributes – Empirical copula
    - Time series of dynamic attributes – Library of time series segments
      - Detect structural changes in time series
  - Multi-attribute range queries
    - Probabilistic finite state machine
  - Preserve statistical characteristics, dependencies, & temporal patterns
  - Dataset neutral
ResQue – Multi-Attribute Resource Generation

- Satisfy KS-test with a significance level of 0.05
- Available [www.engr.colostate.edu/cnrl/Projects/CP2P/](www.engr.colostate.edu/cnrl/Projects/CP2P/)
ResQue – Multi-Attribute Range Query Generation

\[ Q_1 = \{\text{CPUSpeed}\} \times 1 \]
\[ Q_2 = \{\text{MemFree}, 1\text{MinLoad}\} \times 2 \]
\[ Q_3 = \{\text{MemFree}, \text{CPUSpeed}, \text{TxRate}\} \times 1 \]

\[ q_1 = \{\text{CPUSpeed}\} \quad 1/8 \]
\[ q_2 = \{\text{CPUSpeed}, \text{TxRate}\} \quad 1/8 \]
\[ q_3 = \{\text{MemFree}, 1\text{MinLoad}\} \quad 1/2 \]
\[ q_4 = \{\text{MemFree}, \text{CPUSpeed}\} \quad 1/8 \]
\[ q_5 = \{\text{MemFree}, \text{CPUSpeed}, \text{TxRate}\} \quad 1/8 \]

- Probabilistic Finite State Machine (PFSM)
- No of attributes, popularities, & occurrences of attribute pairs are similar
- Satisfy KS-test with a significance level of 0.05
Outline

• Motivation
• Problem statement
• Multi-attribute resource & query characteristics
• Resource & query aware resource discovery
• Multi-attribute resource & range query generation
• Community-based caching
• NDN for DCAS
• Summary & future work
Small communities are emerging within large P2P systems
- Based on semantic, geographic, & organizational interests
  - BitTorrent communities
- Objective – Gain better performance while being in a large system

Ways to improve query/lookup performance
1. Satisfy only the most dominant queries
2. Form clusters of communities

<table>
<thead>
<tr>
<th>Community</th>
<th>EX</th>
<th>FE</th>
<th>SP</th>
<th>TB</th>
<th>TS</th>
<th>TE</th>
<th>TR</th>
</tr>
</thead>
<tbody>
<tr>
<td>fenopy.com</td>
<td>0.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seedpeer.com</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>torrentbit.net</td>
<td>0.40</td>
<td>0.29</td>
<td>0.00</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>torrentscan.com</td>
<td>0.48</td>
<td>0.33</td>
<td>0.00</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>torrentsection.com</td>
<td>0.53</td>
<td>0.23</td>
<td>0.00</td>
<td>0.31</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>torrentreactor.net</td>
<td>0.10</td>
<td>0.08</td>
<td>0.00</td>
<td>0.06</td>
<td>0.09</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>youbittorrent.com</td>
<td>0.36</td>
<td>0.35</td>
<td>0.00</td>
<td>0.29</td>
<td>0.42</td>
<td>0.20</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Cosine similarity among search clouds of communities
Community-Based Caching (Cont.)

- Reduce mixing among communities while in same overlay
  1. Sub-overlay among community members
     - Nodes indicate their communities using Community IDs
     - Find community members by sampling routing tables of nodes pointed by fingers
     - Maintain fingers to those community members
     - Overlay properties are preserved
  2. Cache contents based on community interest
     - “What is important to me is also important to other community members”
     - Local Knowledge-based Distributed Caching (LKDC)

- Path length $O(\log N)$
- By probing $i$-th finger & its successor
  $2(i + 2 \log_2 N - b)$ - 1 nodes can be found
  - $1 \leq i \leq b$
- Community of size $M$ has $M/2^b - i + 1$ peers within the range of $i$-th finger
Distributed Local Caching

• Each overlay node
  – Independently decides what keys to cache based on the queries it forwards
  – Tries to minimize average query cost
    \[
    h_{ave}^n = \frac{\text{Total hops}}{\text{Total queries}} = \frac{\sum_{k \in K} \lambda_k^n h_k^n (1 - x_k^n)}{\sum_{k \in K} \lambda_k^n}
    \]
  – Maximize hop count reduction while satisfying its cache capacity \( C_n \)
    \[
    \text{maximize } R = \sum_{k \in K} \lambda_k^n h_k^n - \sum_{k \in K} \lambda_k^n h_k^n (1 - x_k^n) = \sum_{k \in K} \lambda_k^n h_k^n x_k^n
    \]
    subject to \( \sum_{k \in K} S_k x_k^n \leq C_n \)
  – NP complete
  – For improving lookup performance ok to assume \( S_k = 1 \) → Cache keys with largest

\( \lambda_k^n \), \( h_k^n \), \( x_k^n \), \( C_n \), \( S_k \), \( k \), \( n \), \( K \), \( \lambda_k^n \)
### Distributed Local Caching (Cont.)

**Where to place cache entries?**
- At nodes that forward most number of messages
- $6, (4, 5), (0, 1, 2, 3), \ldots$
- Hops reduce $16, 8, 8, 4, 4, 4, 4, 2, \ldots$
- Hops reduce by placing $c_k$ entries

**How many entries to create?**
- Problem
  \[
  \text{minimize } H_{\text{ave}} = h_{\text{ave}} - \frac{1}{N} \sum_{k \in K} f_k g(c_k)
  \]
  subject to $\sum_{k \in K} c_k = B$, $c_k \leq N - 1$ $\forall k \in K$
- Solution
  - Allocate in proportion to popularity
  \[
  c_k = \begin{cases} 
  \frac{N - 1}{f_k(B - l(N - 1))} & \text{if } k \leq l \\
  \frac{P(l, K, \alpha)}{f_k} & \text{else}
  \end{cases}
  \]
Community-Based Caching – Performance Analysis

- Model provides a lower bound & more accurate than previous approaches
- 40% reduction in average path length
  - Most popular communities – 48-53%
  - Least popular community – 23% (7% with caching)
- Quickly adapt to rapid changes in popularity
Outline

- Motivation
- Problem statement
- Multi-attribute resource & query characteristics
- Resource & query aware resource discovery
- Multi-attribute resource & range query generation
- Community-based caching
- NDN for DCAS
- Summary & future work
NDN for Data Fusion in DCAS Systems [10]

- Internet
  - Designed to share resources → End-to-end
  - Users value ability to access contents → End irrelevant
  - Traffic aggregation, location dependence, & security
- Named Data Networking (NDN/CCN)
  - Access & route contents based on application layer names
  - In-network caching, duplicate message suppression, on demand data generation, better security, & incremental deployment
- Distributed Collaborative Adaptive Sensing (DCAS) systems
  - Multiple redundant sensors, multi-application, & multi-user systems
  - Data pull – Users’ information needs determine how system is used
  - Sensor specific data names
    - “Reflectivity data from CSU CHILL”
  - Users are concerned about a particular event(s) occurring within an Area Of Interest (AOI)
    - “Reflectivity over Fort Collins” or “Wind speed in southwestern Oklahoma”
## Why NDN for CASA?

<table>
<thead>
<tr>
<th>Geographic location &amp; weather event specific names</th>
<th>Content dependent names</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Queries &amp; data</td>
<td>• 2 packet types – Interests &amp; data</td>
</tr>
<tr>
<td>• Aliases for same data</td>
<td>• /FortCollins/Reflectivity/13:32/</td>
</tr>
<tr>
<td></td>
<td>• Multiple names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decouple data, security, &amp; access from sensor</th>
<th>Decouple identity, security, &amp; access from end point</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Use any available sensor</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>High temporal &amp; spatial locality</th>
<th>Exploit temporal &amp; spatial locality</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pull based</th>
<th>Receiver driven communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>• End-user information needs determine what &amp; how resources are used</td>
<td>• On demand data generation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlay routing</th>
<th>Multiple routing schemes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Load balancing, resilience, &amp; security</th>
<th>Better reliability &amp; security</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multi-path routing &amp; mobility</td>
<td>• Multi-path routing &amp; mobility</td>
</tr>
</tbody>
</table>
NDN for DCAS – Naming Data

- End users specify an AOI, application, & time
  - /AOI/application/time/
  - Interest packet is looking for an application near AOI
    - Process data close to source → Save bandwidth
  - AOI is typically expressed as a rectangular area
    - /x₁/y₁/x₂/y₂/application/time
    - Larger AOIs are broken into smaller ones

- Application needs to subscribe to radars
  - CASA radars negotiate among themselves on how to provide data
    - /x₁/y₁/x₂/y₂/radar/time/subscription/n/dataType
    - PIT is modified to accept up to n data packets per tile

- Application pull data from selected radars
  - /xᵣ/yᵣ/xᵣ/yᵣ/radar/time/x₁/y₁/x₂/y₂/bitmap/dataType
NDN for DCAS – Overlay Construction & Query Resolution

• Overlay routing – Content Addressable Network (CAN)
  – Maps to 2D space while preserving locality
  – No local minimas as in other greedy routing solutions
• End users connect to overlay using a set of proxies
• In network caching & duplicate interest suppression
NDN for DCAS – Simulation Setup

- Parameters from CASA IP1 test bed
  - 121 radars placed on a 300 km x 300 km area, 30 km apart, 40 km range
  - 30 PPI scans, unsynchronized radars
  - 4 bytes per data type per tile (tile 100 m x 100 m)
  - 5 proxies, 16 x 2 reflectivity & velocity, & 4 x 3 NBRR, nowcasting, & QPE
  - 1 Gbps links

- Reflectivity data from a large-scale weather event over Oklahoma
  - 05/23/2011 10:00pm to 05/24/2011 2:00am
  - AOI – Active weather if reflectivity ≥ 25 dBz
  - End users – 2 NWS, 30 EMs, 8 researches, & 20 media
NDN for DCAS – Performance Analysis

- Bandwidth requirements are reduced
  - Subscription scheme – 61%, Oldest First Caching (OFC) – 87%
  - Better load distribution
- Better quality data – Waiting time & staleness is reduced
  - Waiting time – 88%, Staleness – 69%
Summary

• Proposed a P2P-based approach to enable collaboration of a group of heterogeneous resources

• Achieved goal of enabling integration of groups of resources & data fusion
  – Real-world datasets exhibit several noteworthy features that affect performance of resource aggregation
  – Resource & query aware P2P-based resource discovery solution
  – Tool to generate synthetic resource & query traces
  – Community-based caching for large P2P systems
  – Demonstrated applicability of NDN for DCAS

Collective power of P2P communities & their resources →
Globally distributed virtual clouds for many applications
Future Work

• Support all key phases of resource aggregation [4, 12]
  – Extend resource & query aware resource discovery solution
  – Hybrid between DHT & superpeer
    • Superpeers – Good for resource matching & binding

• Identify semantic-based P2P communities within overlay [1]
  – Compare with cluster-based solutions, alternative routing, & churn

• Aggregate data from heterogeneous sensors in NDN
  – Integrate other sensors & enhance event-specific queries
  – Reference implementation based on CCNx

• Supporting incentives, trust, security, & privacy [4]
  – Determine ultimate success
  – With right tools & incentives in place, it will be more efficient & rewarding to accomplish a greater task through collaboration
Publications


Acknowledgments

• Prof. Anura Jayasumana
• Prof. V. Chandrasekar, Prof. Daniel Massey, & Prof. Indrajit Ray
• CASA & NSF (award number 0313747)
• Dr. Michael Zink, Veeresh Rudrappa, & Sudharshan Varadarajan
• Dr. Panho Lee, Dr. Sanghun Lim, Vidarshana, Saket, Dulanjalie, Pritam, Yi, Negar, & many other colleagues at CNRL
• Parents, wife, & son

Thank you!