Various Faces of Data Centric Networking and Systems

Eiko Yoneki

University of Cambridge Computer Laboratory

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

2

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)
2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

Shift to Content Based Networking

- Original Internet
  - 70s technology, conversational pipes, end-to-end
- Now, Internet use (>90%):
  - Content retrieval & Service access
  - Request & Delivery of *named data* - access content
- Shift to a content-centric view:
  - Content-awareness and massive storage
  - Existing approach – e.g. Publish/Subscribe overlay

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Multi-Point Communication

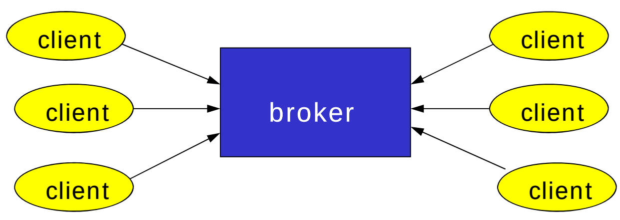
- Application level multicast
  - IP multicast is not supported well over wide area networks
  - Use DHT (Distributed Hashing Table)
  - Use tree routing in order to get logarithmic scaling
  - Bayeux/Tapestry and CAN
  - Service model of multicast is less powerful than content-based messaging system
- Research prototypes of messaging systems
  - Scribe (Topic-based system using DHT over Pastry)
  - SIENA (Content-based distributed event service)
  - JEDI (Content-based messaging system)
  - Gryphon (Topic/content-based message brokering system)

5

UNIVERSITY OF CAMBRIDGE Computer Laboratory

CBN: Content Based Networking

- Publish/Subscribe Paradigm
- Subscription model:
  - Topic-based (Channel)
    - Topics can be in hierarchies but not with several super topics
  - Content-based
    - Express interests as a query over the contents of data
    - How to turn subscriptions into routing mechanism in decentralised environments?



Publish data Subscribe data

6

CDN: *Content Distribution Networks*

- Cache of data at various points in a network
- Content served closer to client→Edge Caching
  - Less latency, better performance
- Load spread over multiple distributed systems
  - Robust (to ISP failure)
  - Handle flashes better (load spread)
- Limitation
  - No mechanism with dynamic/personalized content, while more content is becoming dynamic
  - Difficult to manage content lifetimes and cache performance, dynamic cache invalidation
- CDN Providers
  - Coral Content Distribution Network
  - Akamai
  - BitTorrent
  - …

7

CCN: Content Centric Networking

- Content-Centric Networking (CCN), Named Data Networking (NDN)
- To networking that enables networks to self-organize and push relevant content where needed

  → From CDNs to native *Content Networks*

8

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Goals of CCN

- Network delivers content from closest location
- Integrates a variety of transport mechanisms
- Integrated caching (short-term memory)
- Search for related information
- Verify authenticity and control access

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Existing Related Projects

- Next generation Internet proposals:
  - LNA, TRIAD, NIRA, ROFL, i3, DONA
- Van Jacobsen's CCN and NDN
- PSIRP (Publish/Subscribe Internet Routing Paradigm)
- 4WARD - Architecture and Design for the Future Internet
  - NetInf

...and...

- Traditional Publish/Subscribe Systems, P2P and sensor networks

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Why Big Data?

- Increase of Storage Capacity
- Increase of Processing Capacity
- Availability of Data
- Hardware and software technologies can manage ocean of data

Big Data: Technologies

- Distributed systems
  - Cloud (e.g. Infrastructure as a service)
- Storage
  - Distributed storage (e.g. Amazon S3)
- Data model/indexing
  - High-performance schema-free database (e.g. NoSQL DB)
- Programming Model
  - Distributed processing (e.g. MapReduce)
- Operations on big data
  - Analytics – Realtime Analytics

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Distributed Infrastructure

- Computing + Storage transparently
  - Cloud computing, Web 2.0
  - Scalability and fault tolerance
- Distributed servers
  - Amazon EC2, Google App Engine, Elastic, Azure
  - E.g. EC2 - key decisions for provisioning instances:
    - Pricing? Reserved, on-demand, spot, geography
    - System? OS, customisations
    - Sizing? RAM/CPU based on tiered model
    - Storage? Quantity, type
    - Networking / security
- Distributed storage
  - Amazon S3
  - Hadoop Distributed File System (HDFS)
  - Google File System (GFS), BigTable
  - Hbase

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Challenges

- When you process big data, you need to scale very far and need to build on distribution and combine theoretically unlimited amount of machines to one single distributed storage

15

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Challenges

- Distribute and shard parts over machines
  - Still fast traversal and read to keep related data together
  - Scale out instead scale up
- Avoid naïve hashing for sharding
  - Do not depend of the number of node
  - But difficult add/remove nodes
  - Trade off – data locality, consistency, availability, read/write/search speed, latency etc.
- Analytics requires both real time and post fact analytics – and incremental operation

16

Data Model/Indexing

- Support large data
- Fast and flexible
- Operate on distributed infrastructure

- Is SQL Database sufficient?

NoSQL (Schema Free) Database

- NoSQL database
  - Operate on distributed infrastructure (e.g. Hadoop)
  - Based on key-value pairs (no predefined schema)
  - Fast and flexible

- Pros: Scalable and fast
- Cons: Fewer consistency/concurrency guarantees and weaker queries support

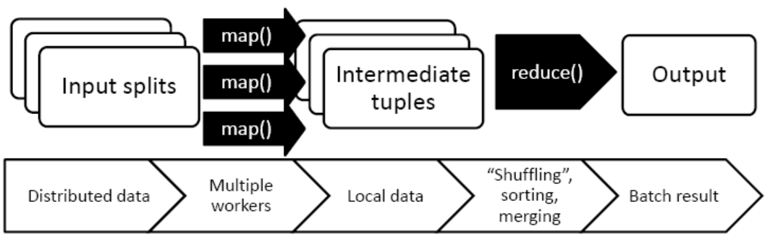
- Implementations
  - MongoDB
  - CouchDB
  - Cassandra
  - Redis
  - BigTable
  - Hibase
  - Hypertable
  - ...

Distributed Processing

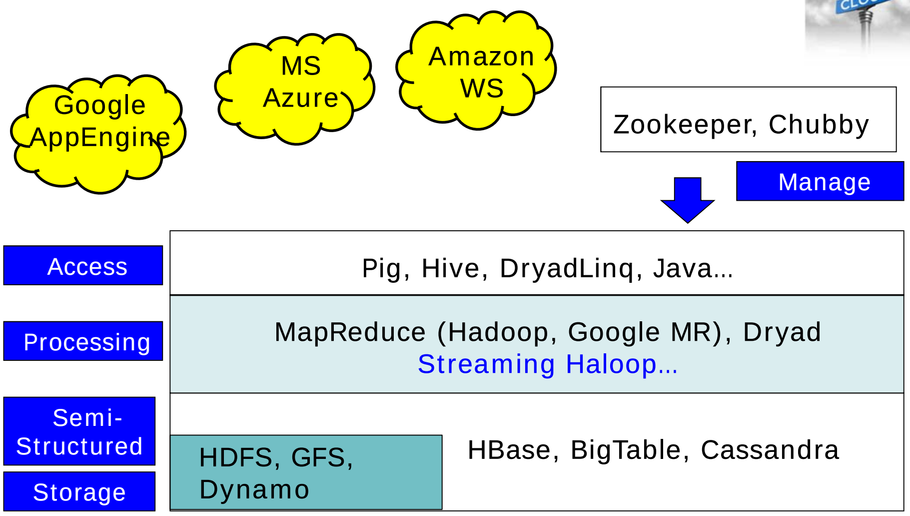
- Non standard programming models
  - Use of cluster computing
  - No traditional parallel programming models (e.g. MPI)
  - E.g. MapReduce

MapReduce

- Target problem needs to be parallelisable
- Split into a set of smaller code (map)
- Next small piece of code executed in parallel
- Finally a set of results from map operation get synthesised into a result of the original problem (reduce)



Distributed Infrastructure



5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

   Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Programming in Data Centric Environment

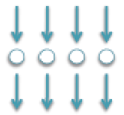
- Data Centre and Cloud environments
  - Applications = a service
  - Platform = a service (e.g. Google AppEngine)
  - Infrastructure = a Service (e.g. Amazon EC2)
  - Challenges:
    - Programming Model (exposure of concurrency, parallelism) and its implementation
    - Physical architecture (new communication protocols, structures)
    - High volume (e.g. billions of entities and terabytes of data) of data management in cloud infrastructure → Data oriented perspective
- Network/System meets Programming

23

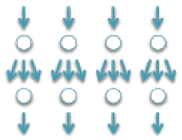
UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Cloud Programming Model

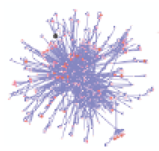
Batch computing: Condor, Grid Engine, Amazon SQS
- Programming Model: Relatively easy, but restricted
- Challenges: Scheduling, Load Balancing, Fault Tolerance
- Resources: Sufficient local memory & cores, fast file system



Loosely coupled: Hadoop, Dryad, Amazon EMR
- Programming Model: More complicated, more expressive
- Challenges: Parallel Communication
- Resources: 4+ cores, 1 TB / core disk, 4 GB / core RAM
  - Cloudera Recommendations: http://bit.ly/bi2lec



Tightly coupled: MPI, Pregel, Hadoop
- Programming Model: Most complicated, most expressive
- Challenges: Parallel Algorithms
- Resources: High Bandwidth, low latency interconnects
  - Amazon Cloud Compute Instance Type



24

Data Flow Programming

- Data parallel programming (e.g. MapReduce, Dryad/LINQ, Skywriting)
- Declarative networking
  - Declarative language: “ask for what you want, not how to implement it”
  - Declarative specifications of networks, compiled to distributed dataflows
  - Runtime engine to execute distributed dataflows
  - Adopting a data centric approach to system design and by employing declarative programming languages → simplify distributed programming

25

CIEL: Dynamic Task Graphs

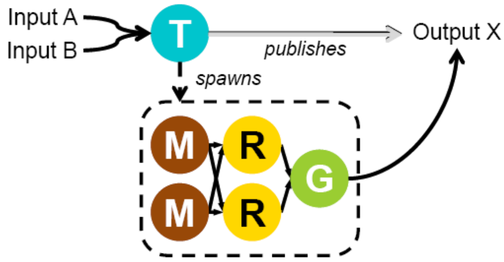
- MapReduce prescribes a **task graph** that can be adapted to many problems
- Later execution engines such as Dryad allow more flexibility, for example to combine the results of multiple separate computations
- CIEL takes this a step further by allowing the task graph to be specified at run time – for example:

```
while (!converged) spawn(tasks);
```

26

Dynamic Task Graph

- Skywriting: Allow tasks to spawn other tasks
- Data-dependent control flow



- CIEL: Execution engine for dynamic task graphs (D. Murray et al. CIEL: a universal execution engine for distributed data-flow computing, NSDI 2011)

27

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

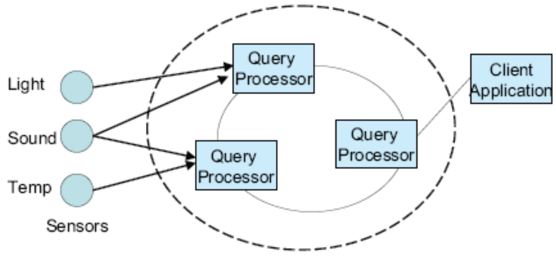
28

Stream Data Processing

- Stream Data Processing and Data/Query Model
  - Stream: infinite sequence of {tuple, timestamp} pairs
  - Continuous query is result of a continuous query is an unbounded stream, not a finite relation
- Data stream processing emerged from the database community (90's)
- Database systems and Data stream systems
  - Database
    - Mostly static data, ad-hoc one-time queries
    - Store and query
  - Data stream
    - Mostly transient data, continuous queries
- Stream data processing is analogue to Complex Event Processing

Sensor Networks and Data Query

- Sensor networks macro-programming
  - State-space, EnviroTrack, Hood, Abstract region
  - Declarative/query: TinyDB
- Data collection: streaming to distributed DB
- Continuous query: Allocation of operators



Real-Time Data

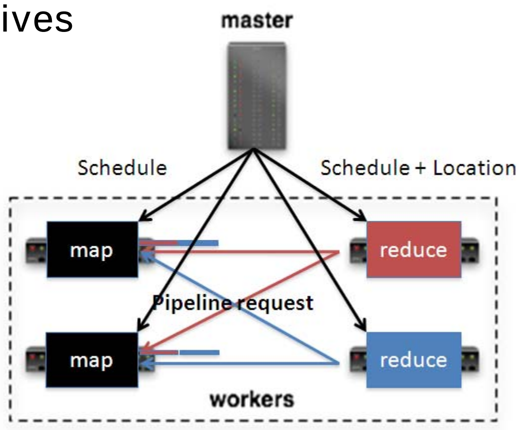
- Departure from traditional static web pages
- New time-sensitive data is generated continuously
- Rich connections between entities

- Challenges:
  - High rate of updates
  - Continuous data mining - Incremental data processing
  - Data consistency

31

Data-Flow in Hadoop Online

- Pipelining within and between MapReduce jobs - Extended to take a series of job
- Support MR jobs continuously: analyse data as it arrives



32

Big Data: Techniques for Analysis

- Applying these techniques: larger and more diverse datasets can be used to generate more numerous and insightful results than smaller, less diverse ones

- Classification
- Cluster analysis
- Crowd sourcing
- Data fusion/integration
- Data mining
- Ensemble learning
- Genetic algorithms
- Machine learning
- NLP
- Neural networks
- Network analysis
- Optimisation
- Pattern recognition
- Predictive modelling
- Regression
- Sentiment analysis
- Signal processing
- Spatial analysis
- Statistics
- Supervised learning
- Simulation
- Time series analysis
- Unsupervised learning
- Visualisation

33

Do we need new Algorithms?

- Can't always store all data
  - Online/streaming algorithms
- Memory vs. disk becomes critical
  - Algorithms with limited passes
- N^2 is impossible
  - Approximate algorithms

34

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Typical Operation with Big Data

- Smart sampling of data
  - Reducing original data with maintaining statistical properties
- Find similar items → efficient multidimensional indexing
- Incremental updating of models → support streaming
- Distributed linear algebra → dealing with large sparse matrices
- Plus usual data mining, machine learning and statistics
  - Supervised (e.g. classification, regression)
  - Non-supervised (e.g. clustering..)

35

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Easy Cases

- Sorting
  - Google 1 trillion items (1PB) sorted in 6 Hours
- Searching
  - Hashing and distributed search

→ Random split of data to feed M/R operation

- Not all algorithms are parallelisable

36

More Complex Case: Stream Data

- Have we seen x before?
- Rolling average of previous K items
  - Sliding window of traffic volume
- Hot list – most frequent items seen so far
  - Probability start tracking new item
- Querying data streams
  - Continuous Query

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

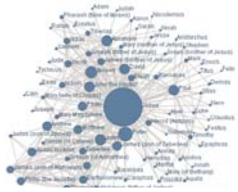
Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

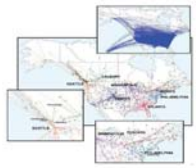
UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Big Graph Data

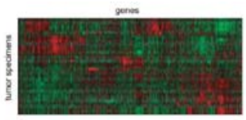
Bipartite graph of appearing phrases in documents



Social Networks

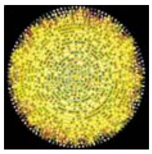


Airline Graph



Gene expression data

Internet Map [lumeta.com]



Protein Interactions [genomebiology.com]

39

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

How to Process Big Graph Data?

- Data-Parallel (MapReduce, DryadLINQ)
  - Generalisation of NoSQL can be found in commodity architecture: Large datasets are partitioned across several machines and replicated
  - No efficient random access to data
  - Graph algorithms are not fully parallelisable
- Parallel DB
  - Tabular format providing ACID properties
  - Allow data to be partitioned and processed in parallel
  - Graph does not map well to tabular format
- Moden NoSQL
  - Allow flexible structure (e.g. graph)
  - Trinity, Neo4J
  - In-memory graph store for improving latency (e.g. Redis, Scalable Hyperlink Store (SHS)) → Expensive for petabyte scale workload

40

Different Algorithms for Graph

- Different Algorithms perform differently

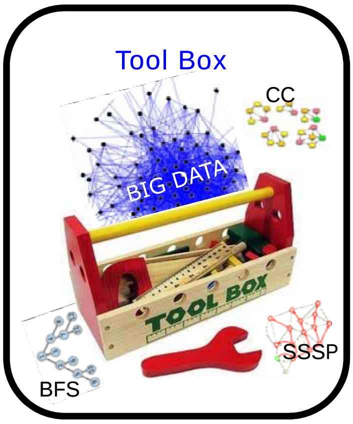
  BFS
  DFS
  CC
  SCC
  SSSP
  ASP
  MIS
  A*
  Community
  Centrality
  Diameter
  Page Rank
  ...

Algorithm	SQL Server PDW	DryadLINQ	SHS
PageRank	8,970	4,513	90,942
SALSA	2,034	439	163
SCC	475	446	214,858/1,073
WCC	4,207	3,844	1,976
ASP	30,379	17,089	246,944

Running time in seconds processing the graph with 50million English web pages with 16 servers (from Najork et al WSDM 2012)

Big Graph Data Processing

- MapReduce is ill-suited for graph processing
  - Many iterations are needed for parallel graph processing
  - Intermediate results at every MapReduce iteration harm performance
- Graph specific data parallel
  - Multiple iterations needed to explore entire graph
  - Iterative algorithms common in Machine Learning, graph analysis



UNIVERSITY OF CAMBRIDGE

Data Parallel with Graph is Hard

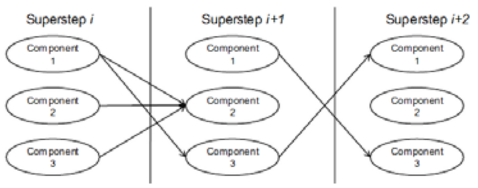
- Designing Efficient Parallel Algorithms
  - Avoid Deadlocks on Access to Data
  - Prevent Parallel Memory Bottlenecks
  - Requires Efficient Algorithms for Data Parallel
- High Level Abstraction Helps → MapReduce
  - But processing millions of data with interdependent computation, difficult to deploy
- Data Dependency and Iterative Operation is Key
  - CIEL
  - GraphLab
  - Naiad
- Graph Specific Data Parallel
  - Use of Bulk Synchronous Parallel Model
  - BSP enables peers to communicate only necessary data while data preserve locality

43

UNIVERSITY OF CAMBRIDGE

Bulk Synchronous Parallel Model

- Computation is sequence of iterations
- Each iteration is called a super-step
- Computation at each vertex in parallel



- Google Pregel: Vertex-based graph processing; defining a model based on computing locally at each vertex and communicating via message passing over vertex's available edges
  - BSP-based: Giraph, HAMA, GoldenORB

44

BSP Example

- Finding the largest value in a strongly connected graph

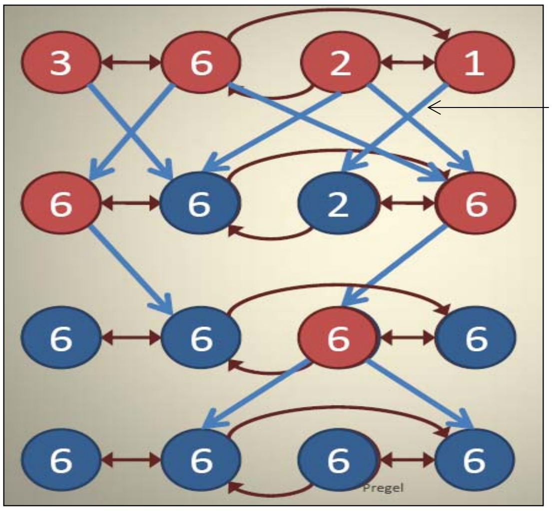
Local Computation

Communication

Local Computation

Communication

...



Message

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

Delay Tolerant Networks

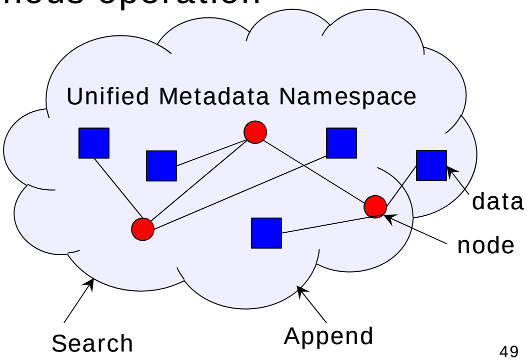
- Delay Tolerant Networks (DTN)
  - Network holds data
  - Path existing over time
  - Store and forward paradigm
- Weak and episodic connectivity - Eventual connectivity
- Non-Internet-like networks
  - Stochastic mobility
  - Periodic/predictable mobility
  - Exotic links
    - Deep space [40+ min RTT; episodic connectivity]
    - Underwater [acoustics: low capacity, high error rates & latencies]

Prototypes: Architecture

- Providing Connectivity to Developing Countries: DakNet
- Vehicular Communications: DriveThru, DieselNet
- Wildlife Tracking: ZebraNet
- Haggle: Pocket Switched Networks, Social Networking
- DTNRG and the Bundle Protocol (RFC 5050)
  - Mostly an engineering approach to implement the InterPlaNetary Internet
- DTN and ICN: both now have content centric view

Haggle Node Architecture

- Each node maintains a data store: its current view of global namespace
  - Persistence of search: delay tolerance and opportunism
- Semantics of publish/subscribe and an event-driven + asynchronous operation
- Multi-platform
  (written in C++ and C)
  - Windows mobile
  - Mac OS X, iPhone
  - Linux
  - Android



49

5 Faces in DCN

1. Content-Centric Networking (CCN) and Content Distribution Networks (CDN)

Big Data

2. Programming in Data Centric Environment
3. Stream Data Processing and Data/Query Model
4. Graph Structured Data: Network, Storage, and Query Processing
5. Network holds Data in Delay Tolerant Networks (DTN)

See You Next Week !

50