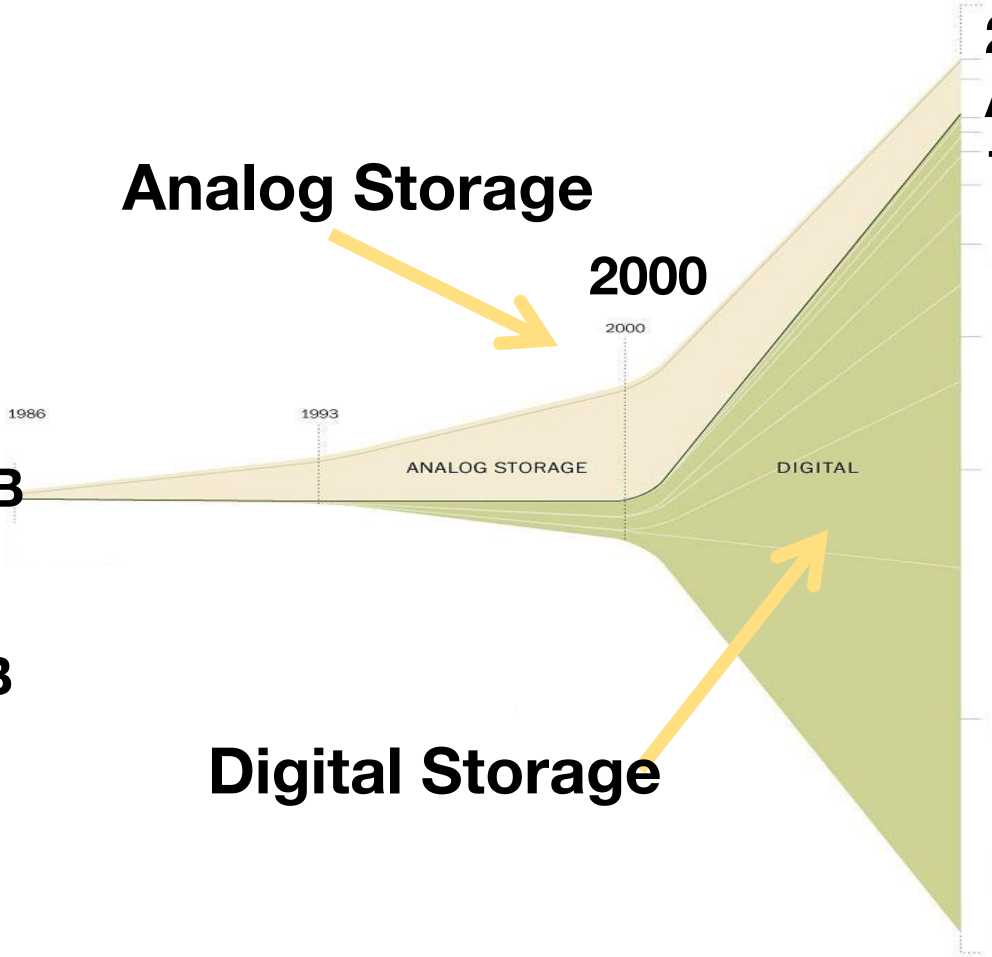
A Benchmark Proposal for Datacenter Computing

Chen Zheng

Institute of Computing Technology, CAS

Data in the World



1986
Analog
2.62 billion GB

Digital
0.02 billion GB

Analog Storage

2000

Digital Storage

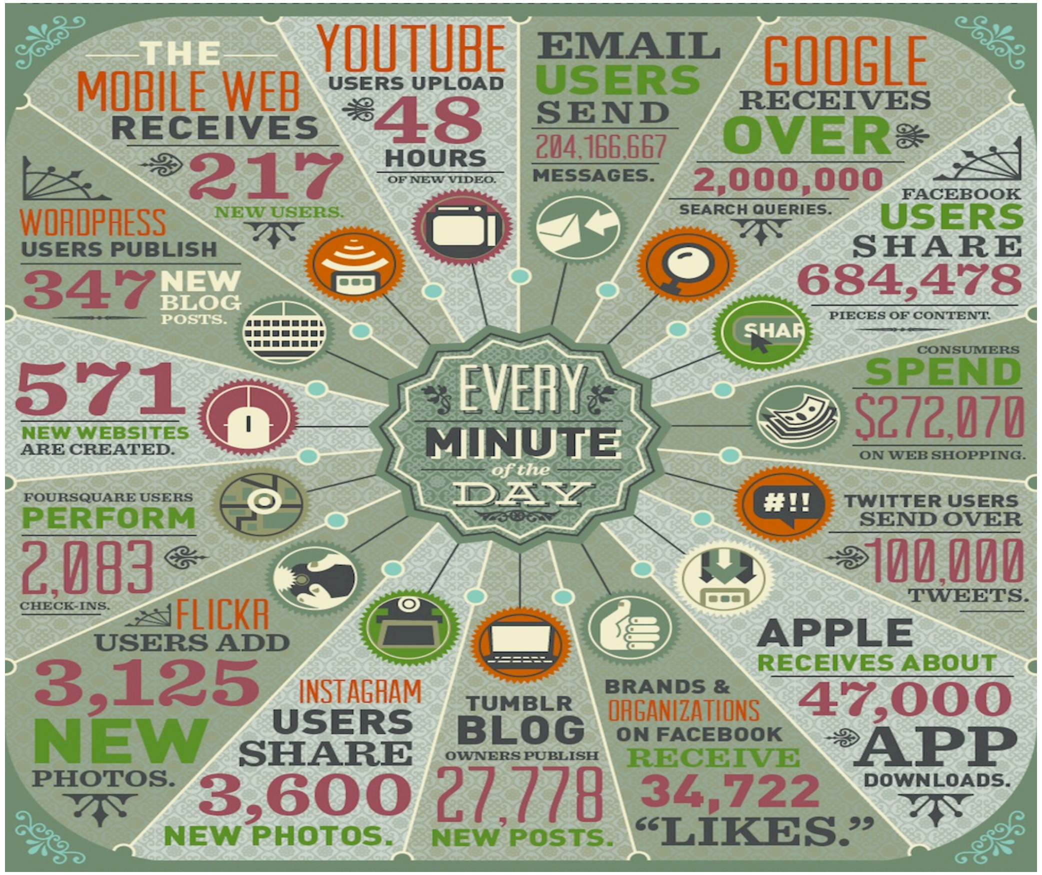
2007
Analog
18.86 billion GB

Digital
276.12 billion GB

(Washington posts 2011) http://www.washingtonpost.com/wp-dyn/content/graphic/2011/02/11/GR2011021100614.html

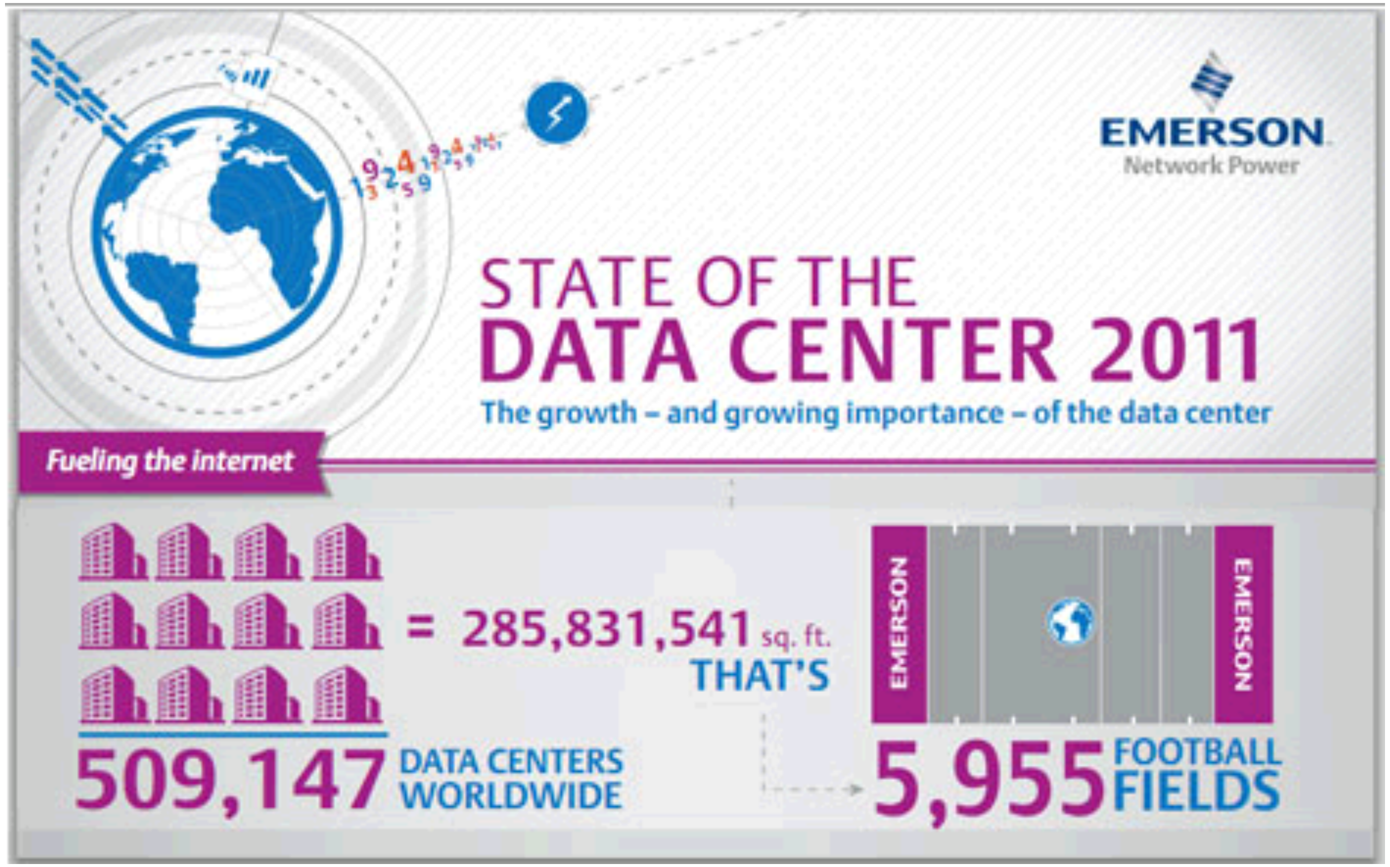
Data Never Sleeps

Data Is Created Every Minute!



http://www.domo.com/blog/2012/06/how-much-data-is-created-every-minute/?dkw=socf3

Data Centers in the World

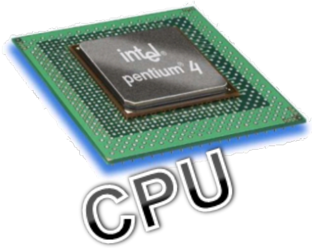


Emerson December 2011

http://www.emersonnetworkpower.com/en-US/About/NewsRoom/Pages/2011DataCenterState.aspx

State-of-Practice Benchmark Suites

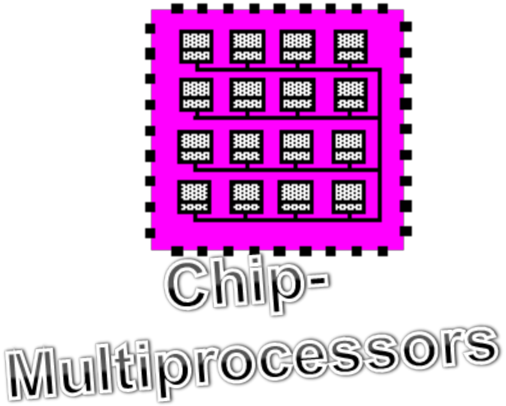
SPEC CPU



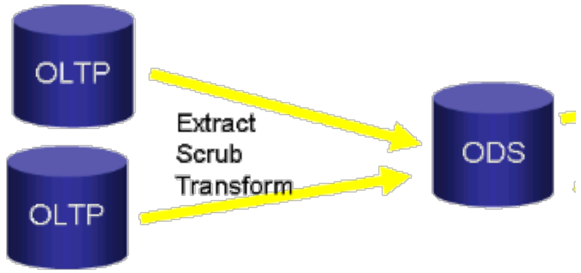
SPEC Web

HPCC

PARSEC



TPCC



Gridmix

YCSB

Why a New Benchmark Suite

- No benchmark suite covers diversity of data center workloads
  - Fast changing
  - The DataCenter has huge software stacks

- State-of-art: BigDataBench
  - includes applications according to its popularity

What we have done?

- BigDataBench
- DC Benchmark
- OS Benchmark

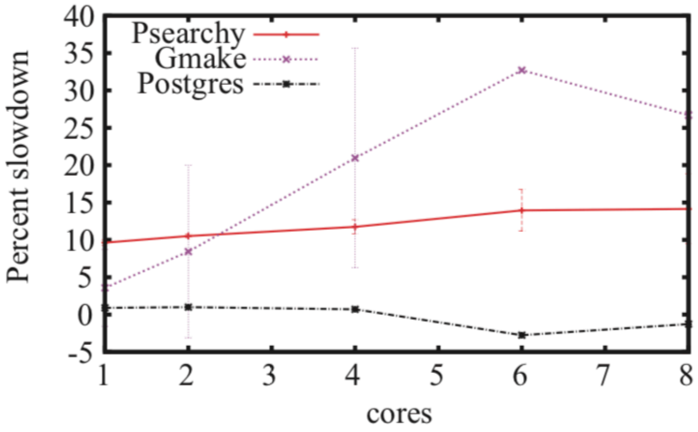
OS Benchmarking

Time	Benchmark	Target	Benchmark workloads
1983	unixbench [15]	Performance	string handling; scientific applications; exec; file copy; pipe throughput; process creation; shell; system call; graphical tests
1996	lmbench [13]	Peroformance	Bandwidths: cache; pipe; tcp; Latencies: context switch; network; file system; process creation; signal handling, syscall overheads; memory latency; Miscellanious: Processor clock rate calculation
1997	hbench [14]	Performance	cache behavior; memroy bandwidth; process creation; file access
2004	BenchIT [70]	Performance	memory (bandwidth & latencies); file access; MPI communication; database (MySQL); sting operations; sort algorithms; binary search; numerical algorithms; applications (CGV; iRODS; MGV; reflection)
2010	mosbench [11]	Scalability	Exim; memcached; Apache; PostgreSQL; gmake; Psearchy; MapReduce (Metis)

Year	System	Micro benchmark	Application-level	workloads
1995	Hive [30]	4	3	*SPLASH-2(RayTrace; ocean)*; *Pmake*; file read; file write; open file; page fault; fault injection (Raytrace or Pmake)
1997	Disco [34]	0	4	*TPC-D on Informix Database*; *Pmake*; *SPLASH-2 (RayTrace)*; *SPEC WEB96 (Apache)*
1999	Tornado [52]	7	0	Memory allocation; object miss handling; object garbage collection; procedure calling; thread Creation; in-core Page Fault; file stat
1999	Celluar Disco [29]	0	4	*TPC-D (Informix)*; *parallel make (Pmake)*; *SPLASH-2(RayTrace)*; *SpecWEB96 (Apache)*; ***TPC-D + RayTrace***
2003	Xen [28]	5	2	SPEC INT2000; *OSDB (PostgreSQL)*; dbench(file system); *SPEC WEB99*; lmbench suite; ***OSDB + SPEC WEB99*** **+ dd + fork bomb**
2005	K42 [73, 74]	0	2	*SPEC SDET*; ***SPEC SDET + streaming applications***
2007	Linux Containers [75]	5	3	lmbench suite; iperf; dd; *dbench*; Postmark; CPU-intensive; *kernel compile*; *OSDB (PostgreSQL)*; ***OSDB*** **+ dd**
2008	Corey [26]	5	2	memclone; mempass; a simple TCP service; object operations (global & local share); file duplication; *Metis*; *webd (filesum)*
2009	HeliOS [23]	6	1	message passing(SingBench); netstack; PostMark; **SAT solver + a disk indexer**; scheduling stress test; *Mail server*
2009	Barrelfish [21]	7	3	message passing; NPB (CG, FT, IS); SPLASH-2 (Barnes-Hut, radiosity); ipbench; *httperf* ; *lighthttpd*; *SQLite*
2010	fos [22, 76]	4	1	system call (local & remote); pings; process creation; file access; *ApacheBench*
2010	Linux [11] (evaluation)	0	7	*Exim*; *memcached*; *Apache*; *PostgreSQL*; *gmake*; *Psearchy*; *MapReduce (Metis)*
2011	Cerburus [77]	5	4	signal handling; process fork & clone; inter-VM message passing; file reading; network; *histogram*; *dbench*; *Apache*; *Memcached*
2012	Dune [35]	11	3	getpid; page fault; page walk; ptrace; trace; appel1; appel2; SPEC2000; *Lighttpd*; *Wedge*; GCBench; Linked List; Hash Map; *XML parser*
2013	Tessellation [33]	1	2	NAS parallel benchmarks (EP); *a video player*; ***video player + dropbox***
2014	K2 [25]	4	0	DMA transfer; ext2fs accessing; UDP loopback; memory allocation

Why we need hybrid OS benchmark?

- Mimic the industry computing scenario
- Isolated performance, scalability may be impacted by hybrid deployment

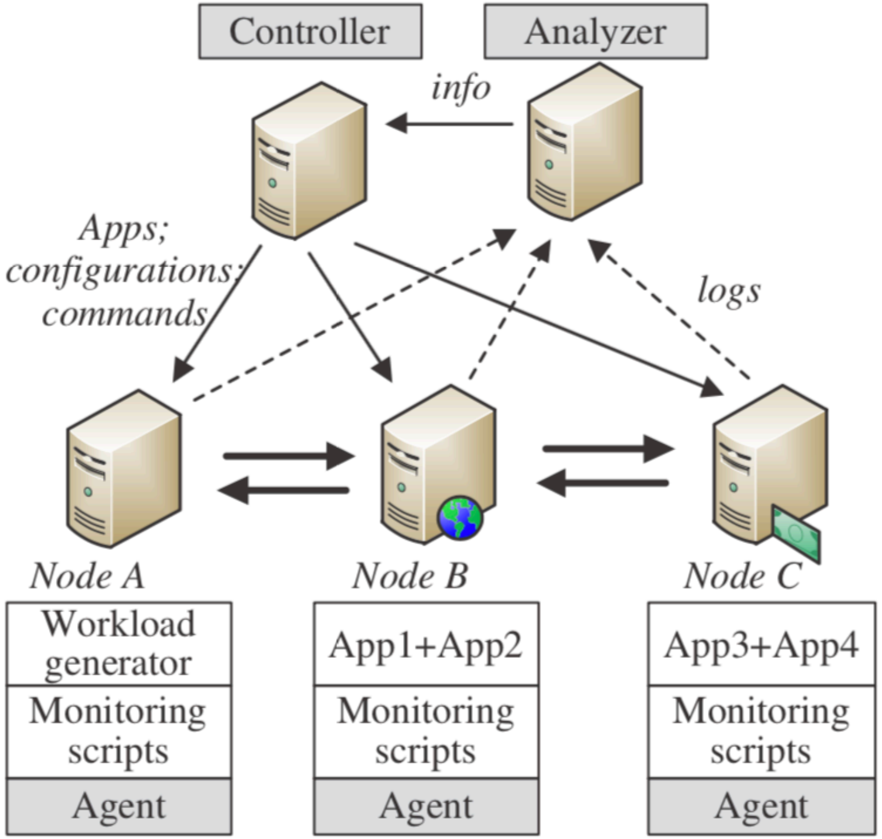


- Tail latency may change

MBench

Type	Workloads	Resource target	Workload type	Metrics
Micro benchmarks	SPEC CPU (bzip2, sphix3)	CPU	-	execution time
	cachebench	cache, memory	-	bandwidth
	IOzone	filesystem (disk)	-	bandwidth
	netperf	network (ethernet)	-	bandwidth
	Will-It-Scale [68]	kernel functionalities	-	throughput, tail latency
Application benchmarks	PARSEC [72] (bodytrack, streamcluster)	CPU, memory, file system	offline batch	execution time
	memcached	memory, CPU, network	service	throughput, tail latency
	Spark (kmeans, pagerank)	memory, CPU, network	analytics	execution time
	PostgreSQL	file system, memory, CPU	service	throughput, tail latency
	Hadoop (sort, grep)	file system, memory, CPU	analytics	execution time
	Search (tomcat; nutch)	network, memory, CPU; CPU, memory, network	service	throughput, tail latency

MController



What we have done?

- BigDataBench

- DC Benchmark

- OS Benchmark

What next – Three Use Cases

- E-Commerce (Alibaba)
- Social Network
- Search Engine

E-Commerce (Alibaba)

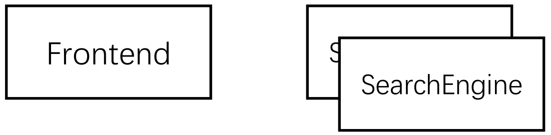
- A benchmark for Alibaba Tmall
  - Recommend product to users
  - Each next search, next page of content need to be recommended in real time

- Workload feature
  - The picture of the whole system
  - User behaviors, e.g., request spike
  - Data distribution, Hot/Cold access pattern

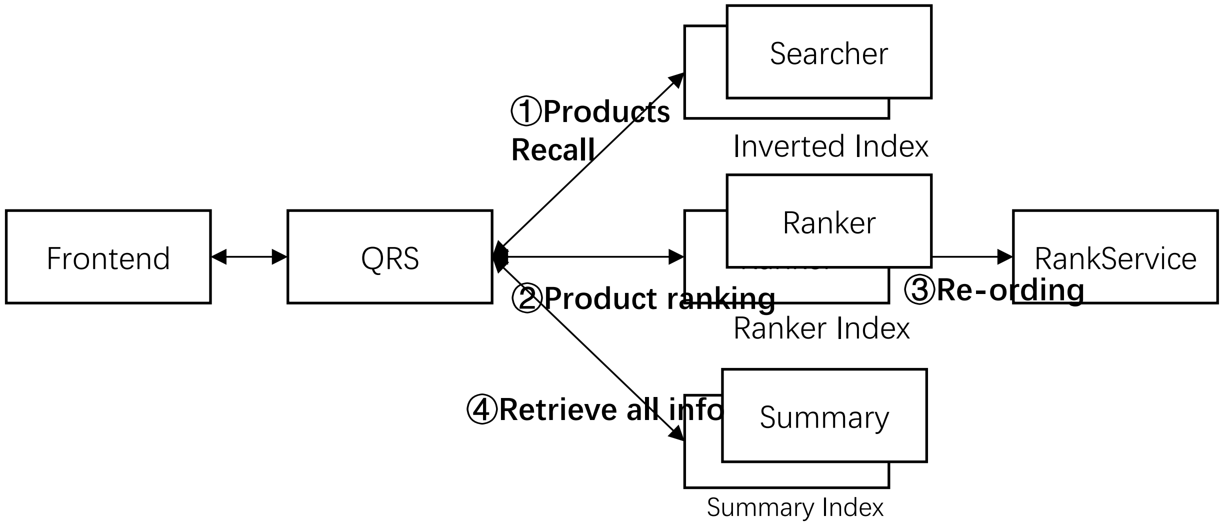
- Data Model
  - Real Data Set
  - Not just statistical generated

Call graph in Taobao Search

- The Search engine is split --- Scalability
  - Searcher
  - Ranker
  - Summary
- Personalized recommendation -- RankServices
  - Add ads
  - Re-ording
  - Not only for search scenarios



Normal Search Engine



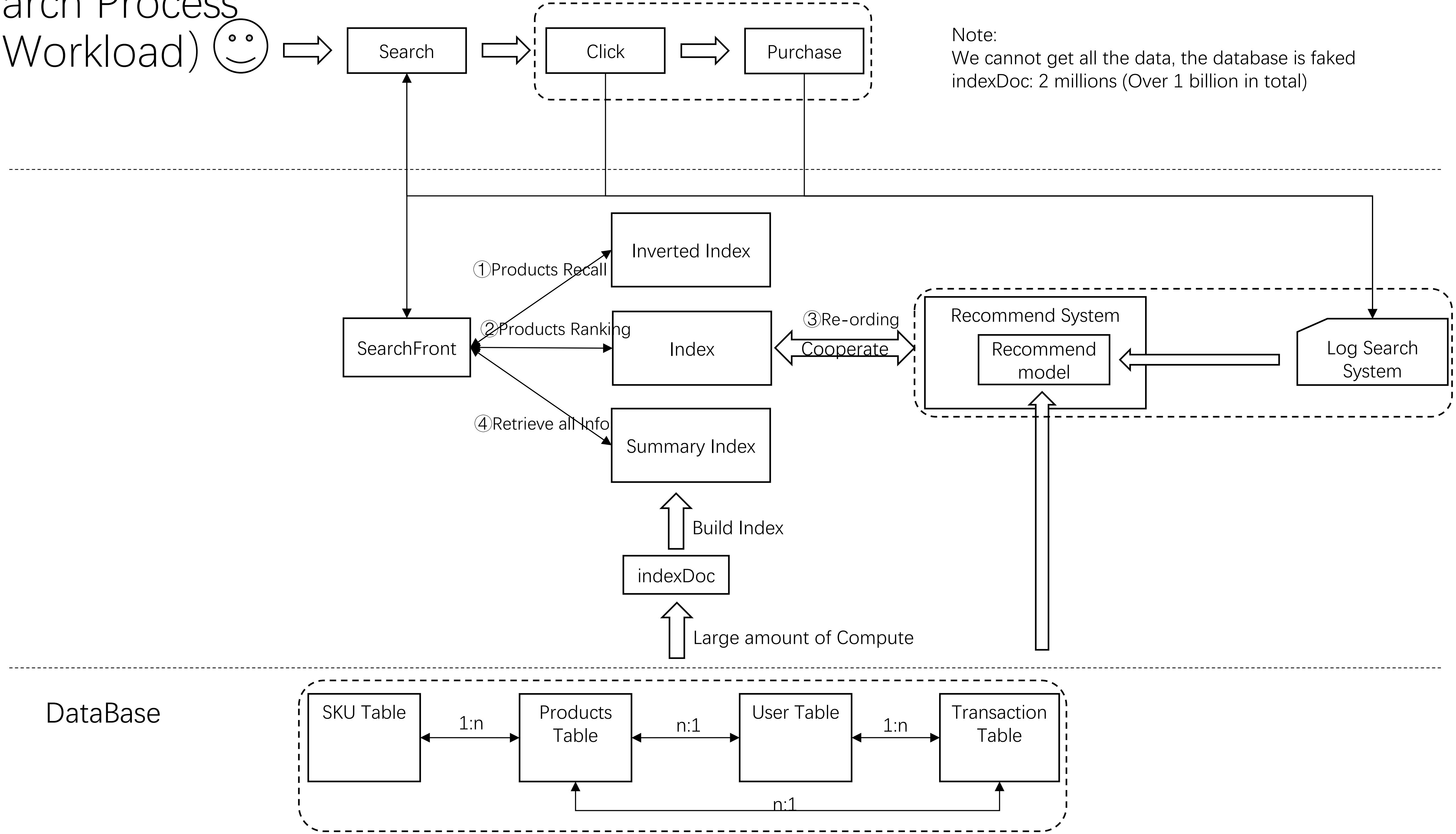
E-commerce Search in Taobao

Comparing with Other Benchmarks

- TPC, SPECWeb
  - Outdated, not updating
- Technology evolve:
  - WebService implementation: SOAP → REST
  - Data Exchange Format: XML → Json/Protubuf
  - HTTP/1.1 → HTTP/2, SPDY, IPV6
- Business model evolve:
  - No personalized recommendation process, no AI support
  - Ingle datastore → co-processing of multiple datastores
  - Multiple datastores make latency a big challenge
  - Only Scale up, not Scale out
- We only include the search part in e-commerce
  - TPC-W, SPECWeb include the entire process
  - They have more diverse interaction

Search Process (Workload)

Note:
We cannot get all the data, the database is faked
indexDoc: 2 millions (Over 1 billion in total)



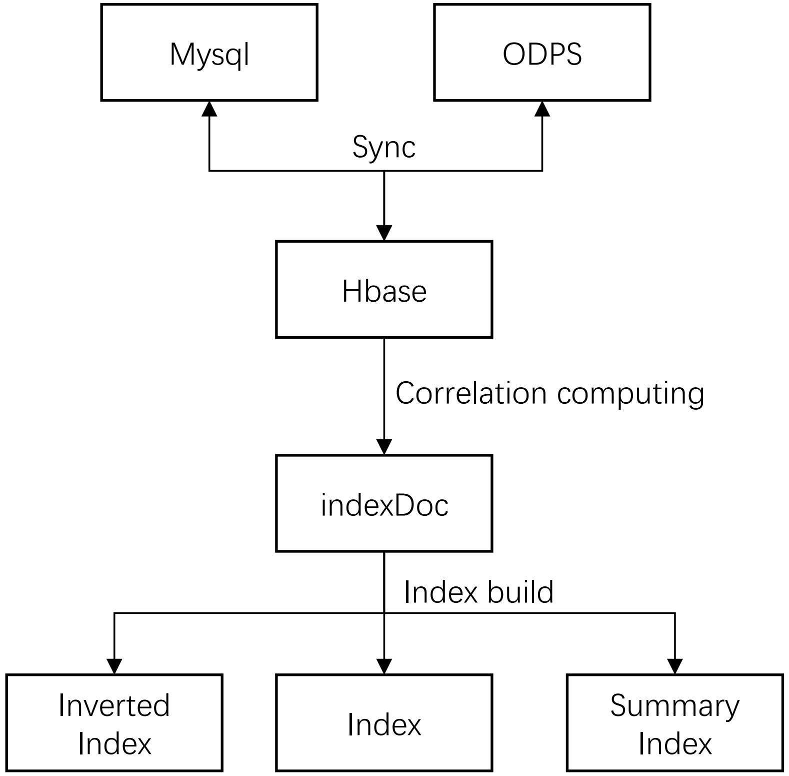
Building Index

- IndexDoc is the input data for indexing

- Hbase synchronize multiple data sources

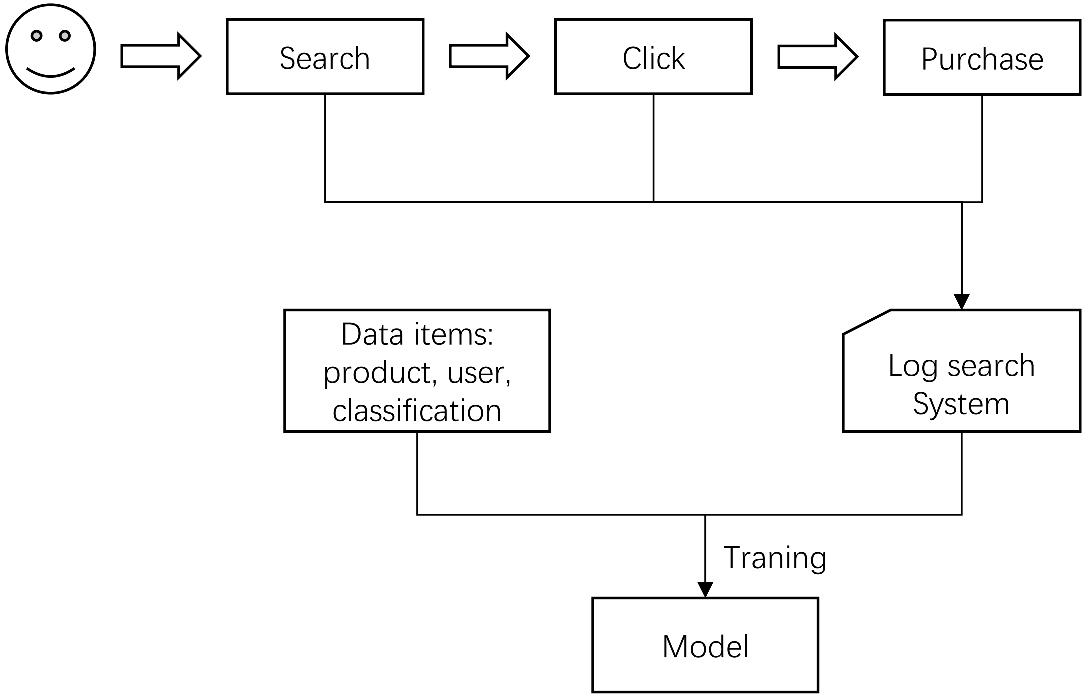
- IndexDox is generated by correlation compute

- Index
  - Full Index built with T+1 updates (Offline)
  - Incremental index built for live updates
  - Incremental index building: using message queues



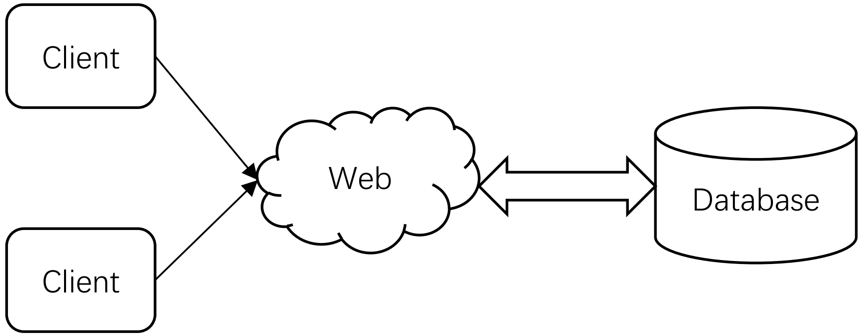
Recommendation System

- Predict user purchase intent
  - User behavior logs, Model, Predict
  - Recommend search words for next search
- Insert ads by changing the ranking list
- Model update:
  - T+1 update (offline)
  - Ncremental model for real-time update (online)
- Latency requirements:
  - behavior generation, log collection, filter cleaning, modeling, updating to online system
  - Entire processed in seconds
- Taobao uses iGraph as storage system

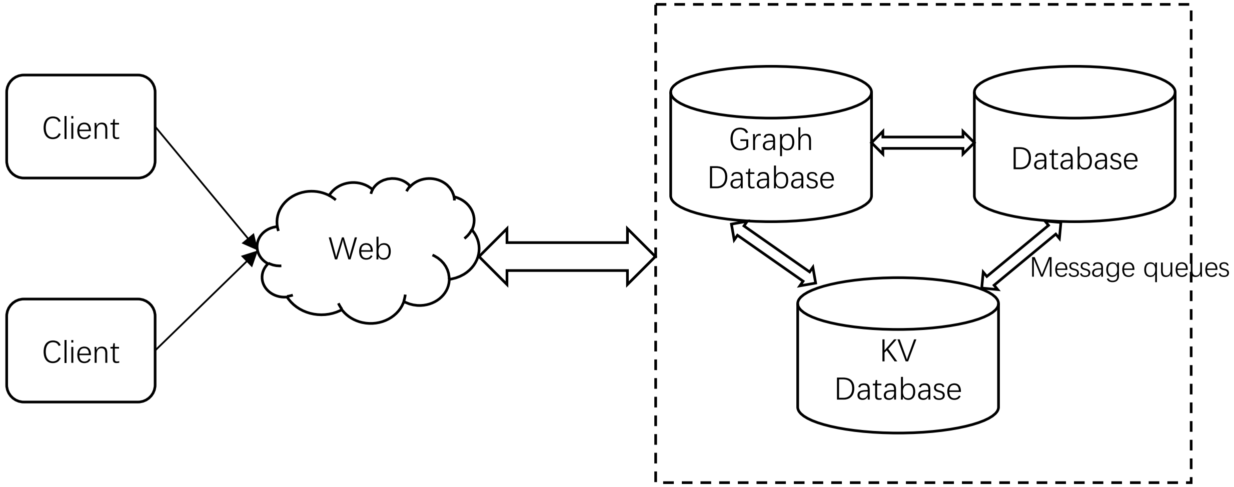


Data management feature

- Traditional online Service
  - Single DBMS，most are relational database（DBMS）
  - Strong consistency



- Online services in E-commerce
  - SQL & NoSQL DBMS
  - Personalized recommend: cooperate in multiple databases
  - Weak consistency
  - Scalability



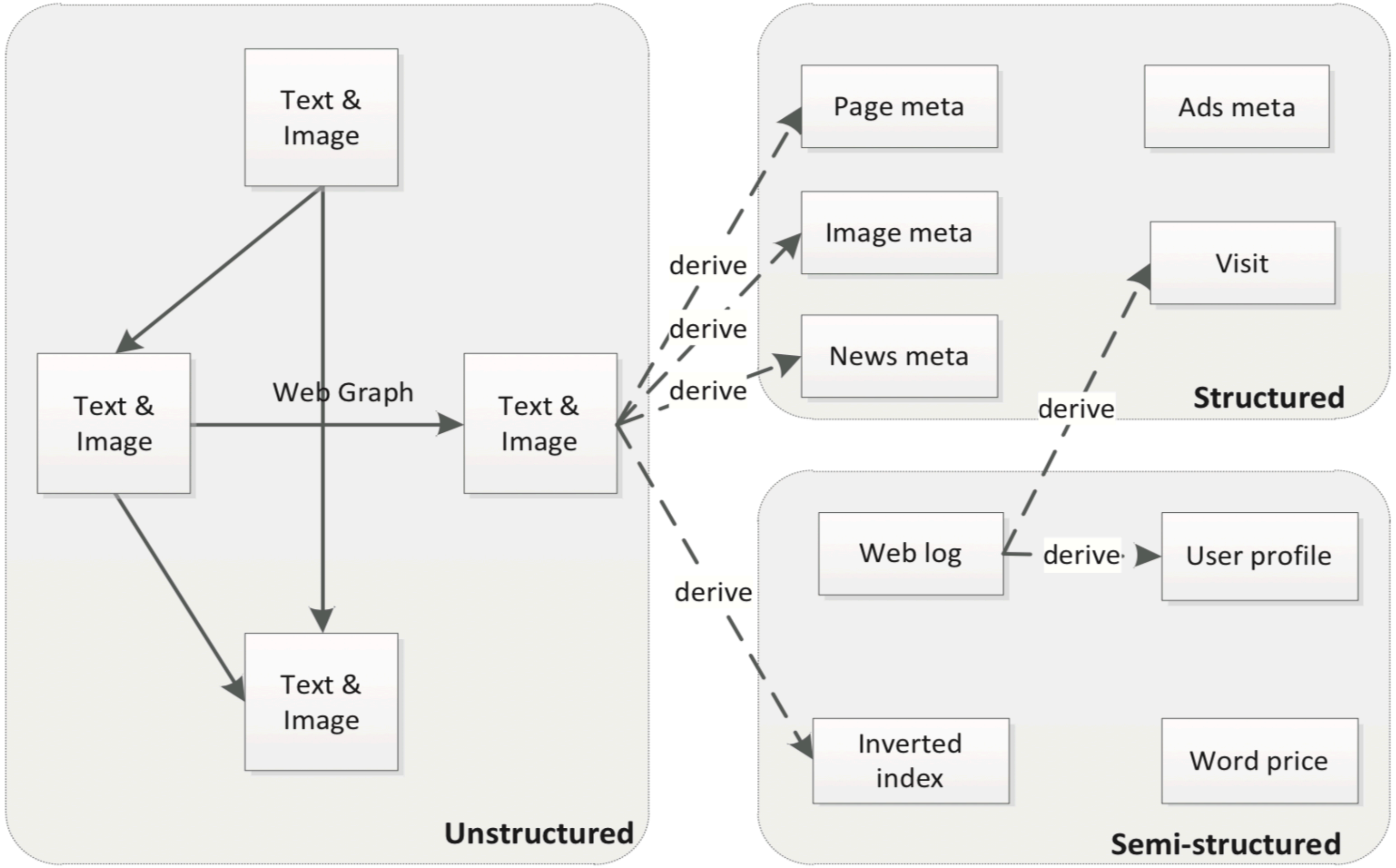
Data Access Pattern

- NoSQL
  - Low read/write latency
  - High throughput for read/write
  - Most transaction not meet ACID
- Personalized recommendation
  - The backend is not supported with a single DBMS like traditional buiness
  - Mine info from multiple databases
  - Real-time recommendation is processed in seconds.

Build the benchmark

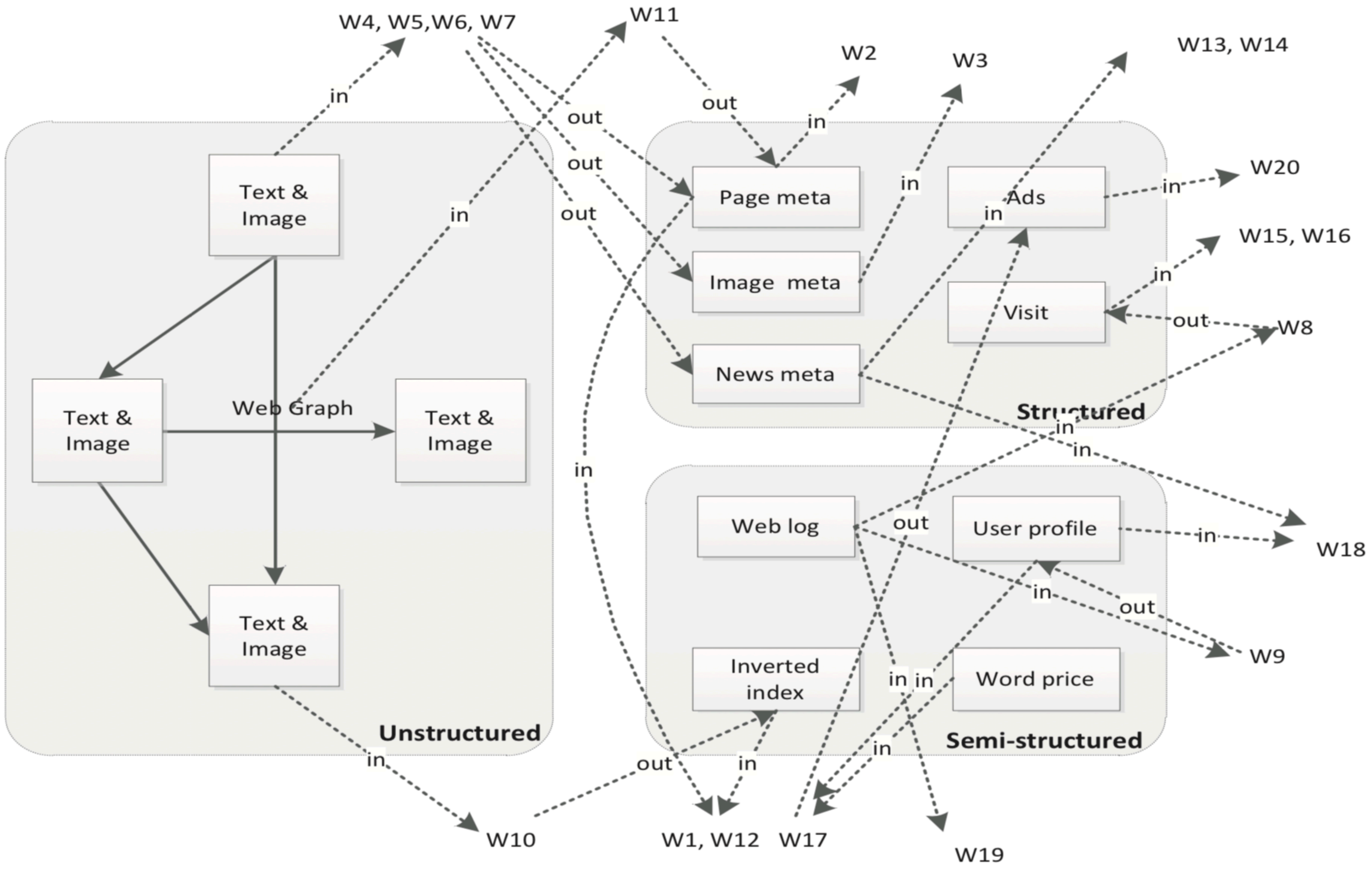
- With same flow, but limited data/ generated data
- Data motif way?

Search Engine: Data Model



Search Engine: Workload

- W1: General Search
- W2: Snapshot
- W3: Image search
- W4: Extraction meta data of web page
- W5: Extraction meta data of news
- W6: Extraction meta data of image
- W7: Abstract extraction
- W8: Extraction meta data of search query
- W9: User profiling
- W10: Indexing
- W11: Web rank computing
- W12: News search
- W13: News classification
- W14: Hot news
- W15: Hot search word
- W16: Similar query mining
- W17: Targeted advertising
- W18: News recommendation
- W19: Anomaly visit detection
- W20: Revenue reporting



Search Engine: Data Generator

- Generate the Internet

- Generate the query

http://prof.ict.ac.cn

Thank You!