

Scalability

non-functional influences



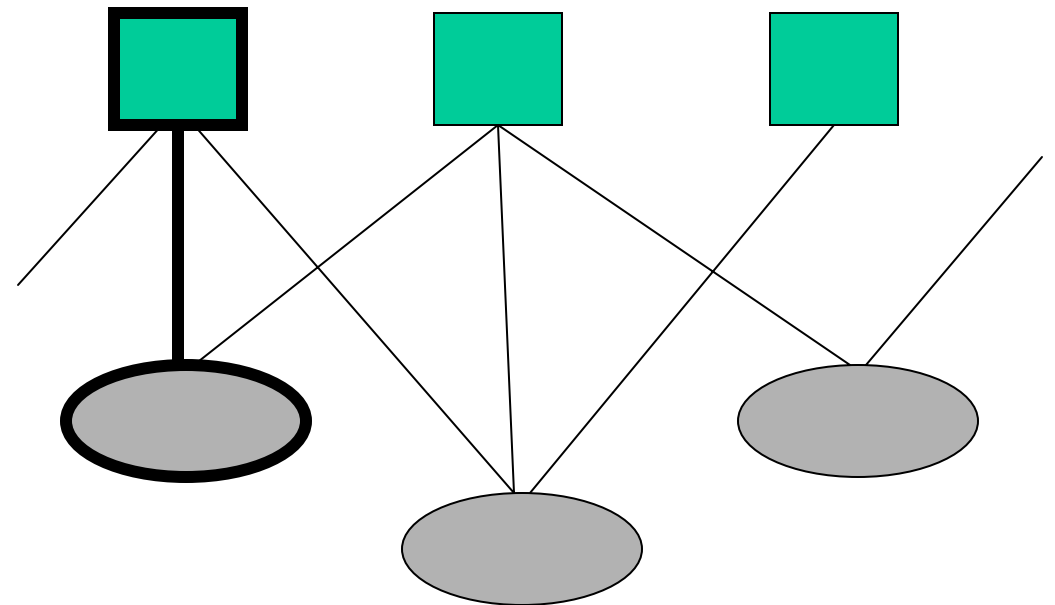
Reto Krummenacher, Elena Simperl (UIBK)

02/10/07



Assumptions

- Openness
- Dynamics
- Distributedness
- Decentralization
- Scale



- out(data): functionality to publish
- rd(query): functionality to retrieve

- Query Complexity
 - Triple pattern matching
 - Query resolution (reasoning)
- Explicit vs. Inferred Data
 - Pure data – no reasoning
 - Local reasoning
 - Distributed reasoning
- Transactions

Non-Functionals



- Security of the whole TS – retrieve,publish
 - Fault-tolerance – retrieve,publish
 - Availability – retrieve,publish
 - Reliability – retrieve,publish
 - Completeness – retrieve
 - Correctness – retrieve
 - Consistency – retrieve
 - Durability – publish
 - Response time – retrieve,publish
 - Scalability – retrieve,publish
- Heavily TS related
- Heavily data related

- Redundancy:
 - duplication of processes (n-out-of-m, $n < m$)
- Recovery:
 - reinstallation of previous state
- Load balancing:
 - decentralization of processes
 - By use of replication
 - By use of distribution of data (e.g. partitioning, clustering)
- Replication:
 - cloning and distributing
- Distribution of data:

- **Fault-Tolerance**
 - Redundancy [prevention, 2-out-of-2]
 - Recovery [in case of failure]
- **Availability**
 - Load balancing [1-out-of-N]
- **Reliability**
 - Redundancy
- **Completeness**
 - (No distribution of data)

- Consistency [No system-caused inconsistency]
 - (No replication)
- Correctness [No false-positives, a query engine issue]
- Durability
 - Recovery [persistency given by storage]
- Response Time
 - **Inter-Kernel** Load balancing
- Scalability
 - *Trade-off management*

- Correctness \leftrightarrow Completeness
- Completeness \leftrightarrow Availability
Load balancing by distribution
- Consistency \leftrightarrow Availability
Load balancing by replication
- Consistency \leftrightarrow Fault Tolerance
Replication might be the hidden tool
- Reliability \leftrightarrow Response Time
- Completeness \leftrightarrow Response Time
- Consistency \leftrightarrow Response Time
- Durability \leftrightarrow Response Time
Non local writing results in latency on out

Scalability Levels: Open Questions



- TS access via SpaceURL only
 - DNS-like discovery of spaces
 - DNS scales, issue?

Scalability Levels: Open Questions



- TS access without SpaceURL a la Google (search)
 - ok: no guarantees, not primarily data-driven

- TS access without SpaceURL a la Web
 - With discovery via TS Ontology-enabled „Routing Tables“??
 - With discovery via s-hashing??
 - Where to end the routing: Isolation of kernels or finite search path → No completeness
 - Distributed reasoning or large scale (local) reasoning

- **Issues with multiread/multiout**