Survey of RDF Storage Managers

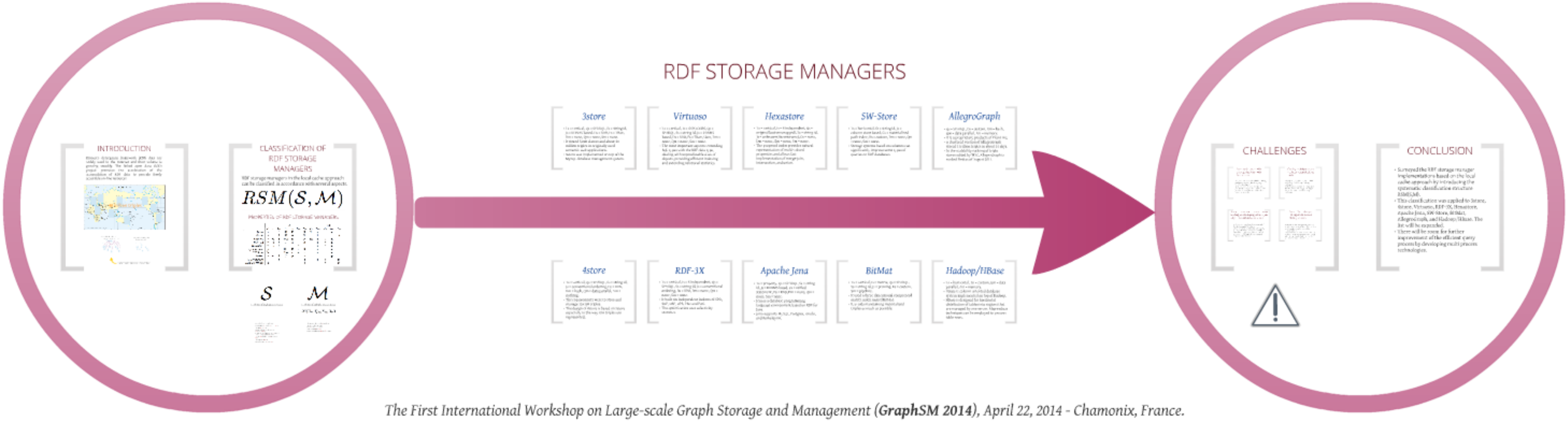
Kiyoshi Nitta
Yahoo JAPAN Research
Tokyo, Japan
knitta@yahoo-corp.jp

Iztok Savnik
University of Primorska &
Institute Jozef Stefan, Slovenia
iztok.savnik@upr.si

Subject

Predicate

Object



*The First International Workshop on Large-scale Graph Storage and Management (**GraphSM 2014**), April 22, 2014 - Chamonix, France.*

Survey of RDF Storage Managers

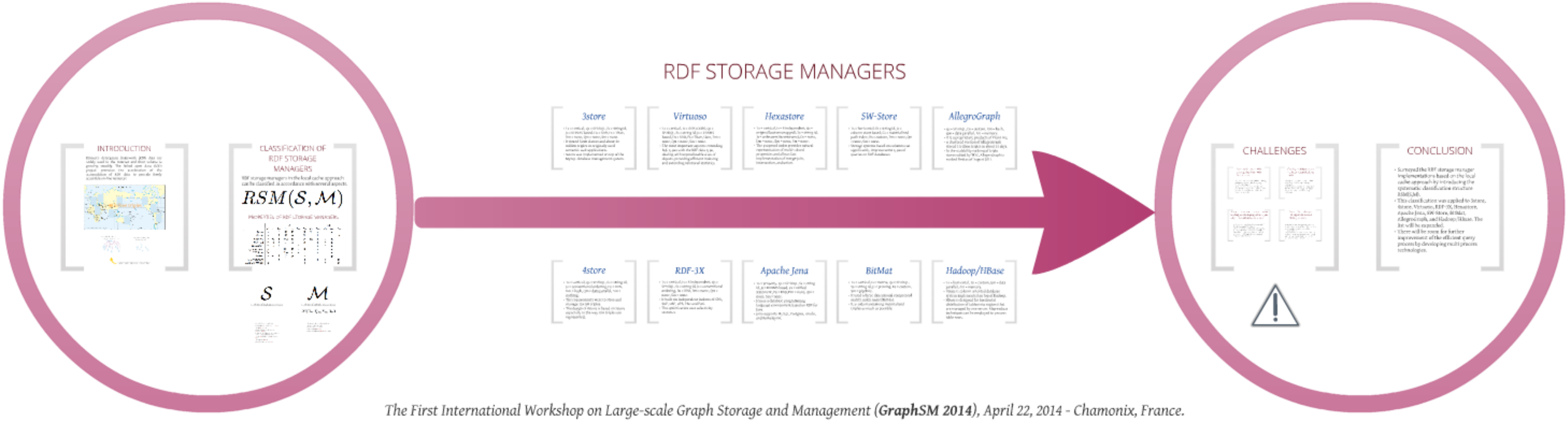
Kiyoshi Nitta
Yahoo JAPAN Research
Tokyo, Japan
knitta@yahoo-corp.jp

Iztok Savnik
University of Primorska &
Institute Jozef Stefan, Slovenia
iztok.savnik@upr.si

Subject

Predicate

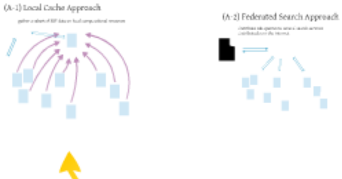
Object



*The First International Workshop on Large-scale Graph Storage and Management (**GraphSM 2014**), April 22, 2014 - Chamonix, France.*

INTRODUCTION

Resource description framework (RDF) data are widely used in the Internet and their volume is growing steadily. The linked open data (LOD) project promotes the acceleration of the accumulation of RDF data to provide freely accessible on-line resources


(A-1) Local Cache Approach
gather a subset of RDF data in local computational resources
(A-2) Federated Search Approach

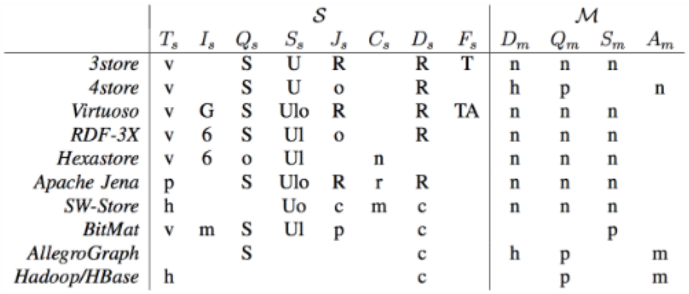
plays an important role for query process efficiency

CLASSIFICATION OF RDF STORAGE MANAGERS

RDF storage managers in the local cache approach can be classified in accordance with several aspects.

$$RSM(\mathcal{S},\mathcal{M})$$

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

$$\mathcal{S}$$

attributes of single process issues

$$\mathcal{S}(T_s, I_s, Q_s, S_s, J_s, C_s, D_s, F_s)$$

$$\mathcal{M}$$

attributes of multiple process issues

$$\mathcal{M}(D_m, Q_m, S_m, A_m)$$

RDF STORAGE MANAGERS

3store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox, Dm = none, Qm = none, Sm = none.
- It stored 5,000 classes and about 20 million triples in originally used semantic web applications.
- 3store was implemented on top of the MySQL database management system.

Virtuoso

- Ts = vertical, Is = GSPO OGPS, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox,ABox, Dm = none, Qm = none, Sm = none.
- The most important aspects: extending SQL types with the RDF data type, dealing with unpredictable sizes of objects, providing efficient indexing and extending relational statistics.

Hexastore

- Ts = vertical, Is = 6 independent, Qs = original(customwrapped), Ss = string id, Js = unknown(itseemsnone), Cs = none, Dm = none, Qm = none, Sm = none.
- The proposed index provides natural representation of multi-valued properties and allows fast implementation of merge-join, intersection, and union.

SW-Store

- Ts = horizontal, Ss = string id, Js = column store based, Cs = materialized path index, Ds = custom, Dm = none, Qm = none, Sm = none.
- Storage systems based on columns can significantly improve some types of queries on RDF databases.

AllegroGraph

- Qs = SPARQL, Ds = custom, Dm = hash, Qm = data parallel, Am = memory.
- It is a proprietary products of Franz Inc.
- A clustered version of AllegroGraph stored 1 trillion triples in about 14 days.
- In the scalability ranking of triple stores edited by W3C, AllegroGraph is ranked first as of August 2011.

4store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = hash, Qm = data parallel, Am = nothing.
- The requirements were to store and manage 15x109 triples.
- The design of 4store is based on 3store especially in the way RDF triples are represented.

RDF-3X

- Ts = vertical, Is = 6 independent, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = none, Qm = none, Sm = none.
- It built six independent indexes of SPO, SOP, OSP, OPS, PSO and POS.
- The optimization uses selectivity statistics.

Apache Jena

- Ts = property, Qs = SPARQL, Ss = string id, Js = RDBMS based, Cs = reified statement, Ds = RDB, Dm = none, Qm = none, Sm = none.
- Jena is a database programming language environment based on RDF for Java
- Jena supports MySQL, Postgres, Oracle, and BerkeleyDB.

BitMat

- Ts = vertical, Is = matrix, Qs = SPARQL, Ss = string id, Js = pruning, Ds = custom, Sm = pipeline.
- It used a three dimensional compressed matrix index named BitMat.
- It avoids maintaining materialized triples as much as possible.

Hadoop/HBase

- Ts = horizontal, Ds = custom, Qm = data parallel, Am = memory.
- HBase is column-oriented database system implemented on top of Hadoop.
- HBase is designed for horizontal distribution of tables into regions that are managed by one server. Map-reduce techniques can be employed to process table rows.

CHALLENGES

More varied values with S properties than with M attributes

- Researches so far have succeeded in achieving good performances by developing single process technologies.
- While practical semantic web applications tend to process large-scale data sets, solutions based on data distribution parallelism have become more popular.

Caching techniques have not been researched that much

- Only Apache Jena and SW-Store reported confirming the efficiency of caching techniques.
- Technologies for automatic investigation and classification of processing queries might become important to utilize caching technologies.

Many researches have been working on developing efficient join algorithms with index structures

- This area has a long history in the research of database management systems.
- While the accumulated RDF data-set is rapidly growing and SPARQL queries are basically constructed from joins of triple patterns, join operations will be applied more strongly in semantic web applications.

Most RDF storage managers can accept SPARQL queries

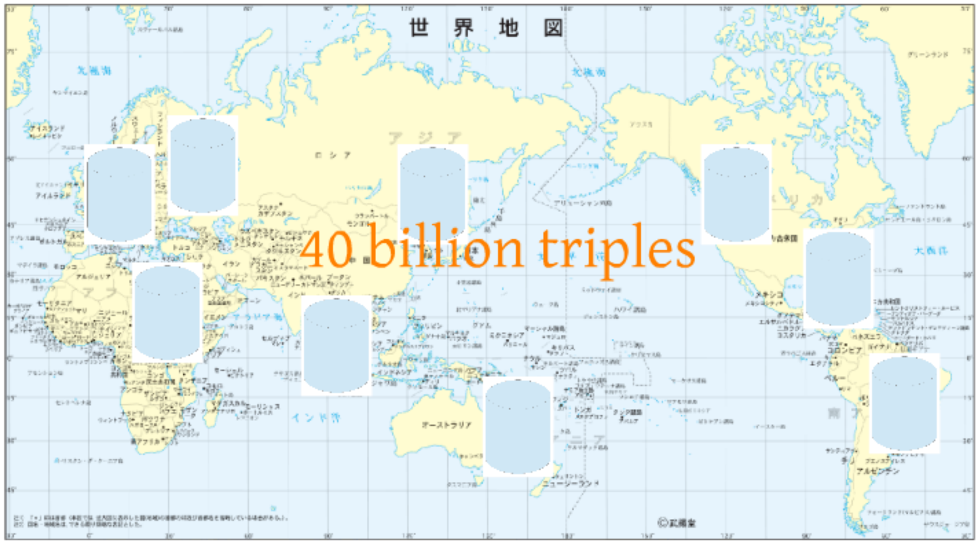
- SPARQL-based RDF storage managers rarely cause semantic mismatch due to the existence of RDF algebras described in the W3C recommendation.
- While OPTIONAL operator was introduced to make the query language convenient enough, efficient processing of such queries will be one of the most crucial challenges.

CONCLUSION

- Surveyed the RDF storage manager implementations based on the local cache approach by introducing the systematic classification structure RSM(S,M).
- This classification was applied to 3store, 4store, Virtuoso, RDF-3X, Hexastore, Apache Jena, SW-Store, BitMat, AllegroGraph, and Hadoop/HBase. The list will be expanded.
- There will be room for further improvement of the efficient query process by developing multi process technologies.

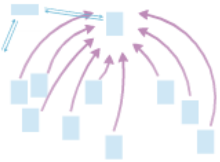
INTRODUCTION

Resource description framework (RDF) data are widely used in the Internet and their volume is growing steadily. The linked open data (LOD) project promotes the acceleration of the accumulation of RDF data to provide freely accessible on-line resources



(A-1) Local Cache Approach

gather a subset of RDF data on local computational resources

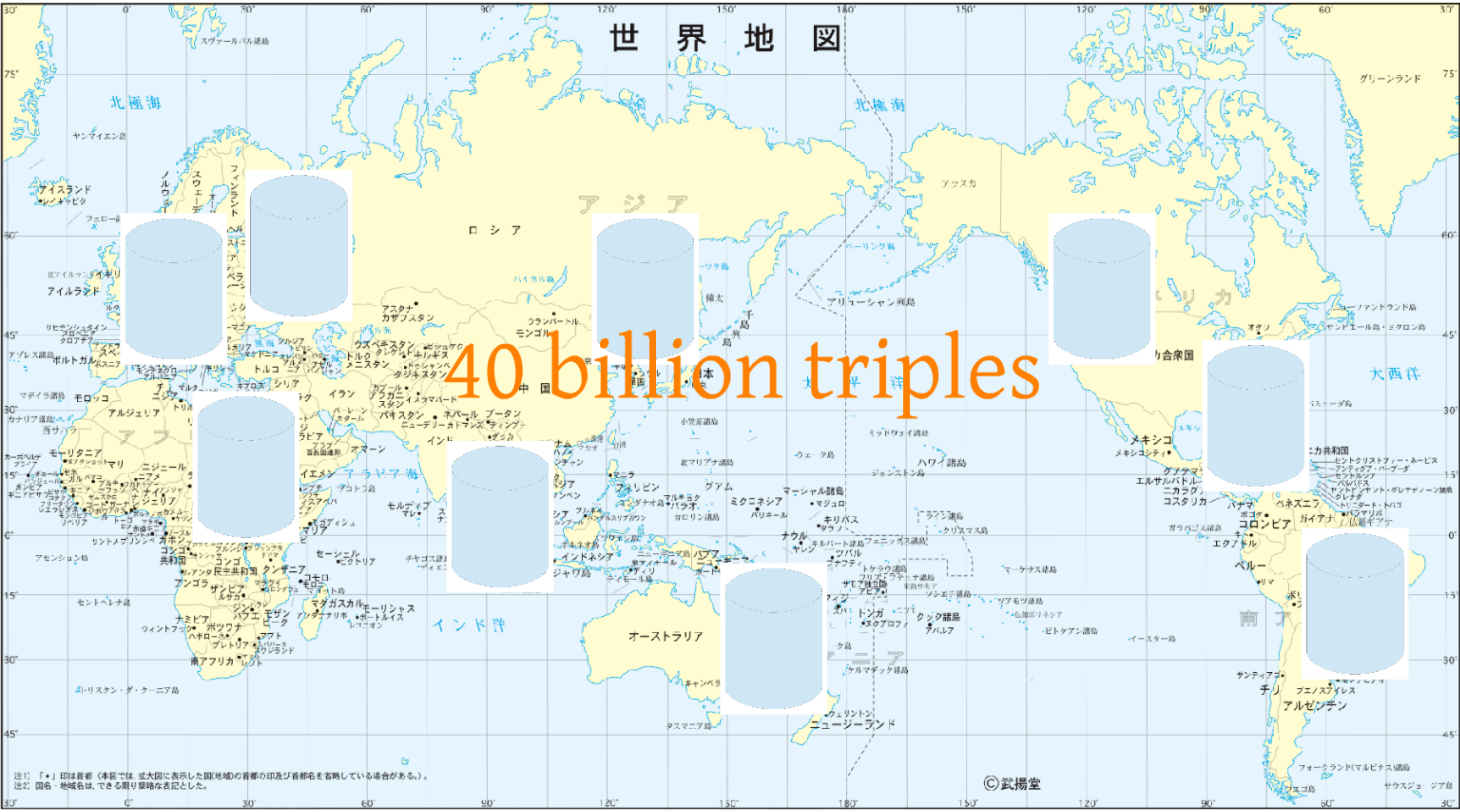


(A-2) Federated Search Approach

distribute sub-queries to several search services distributed over the Internet

plays an important role for query process efficiency

accessible on-line resources



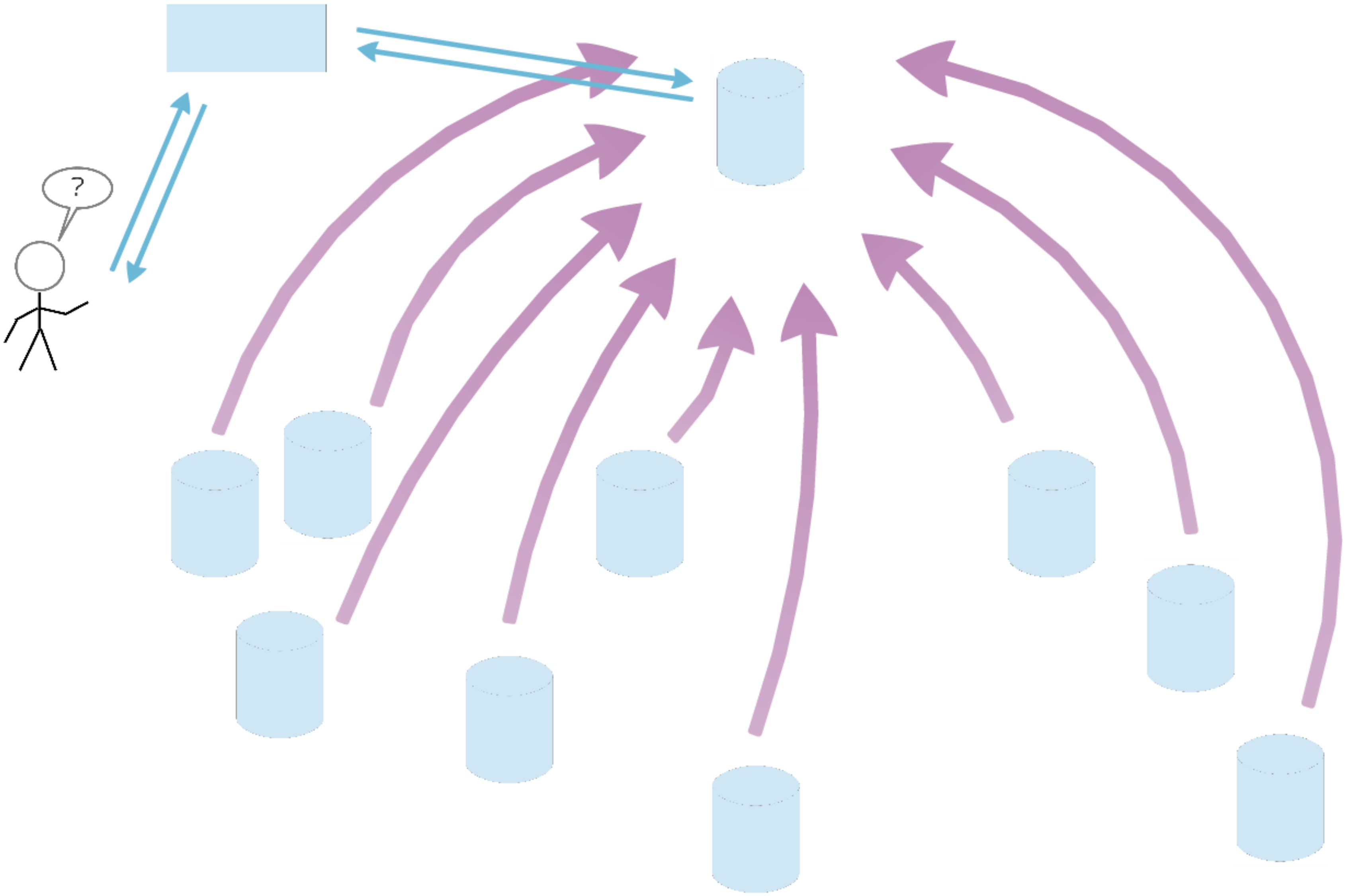
40 billion triples

(A-1) Local Cache Approach

(A-2) Federated Search Approach

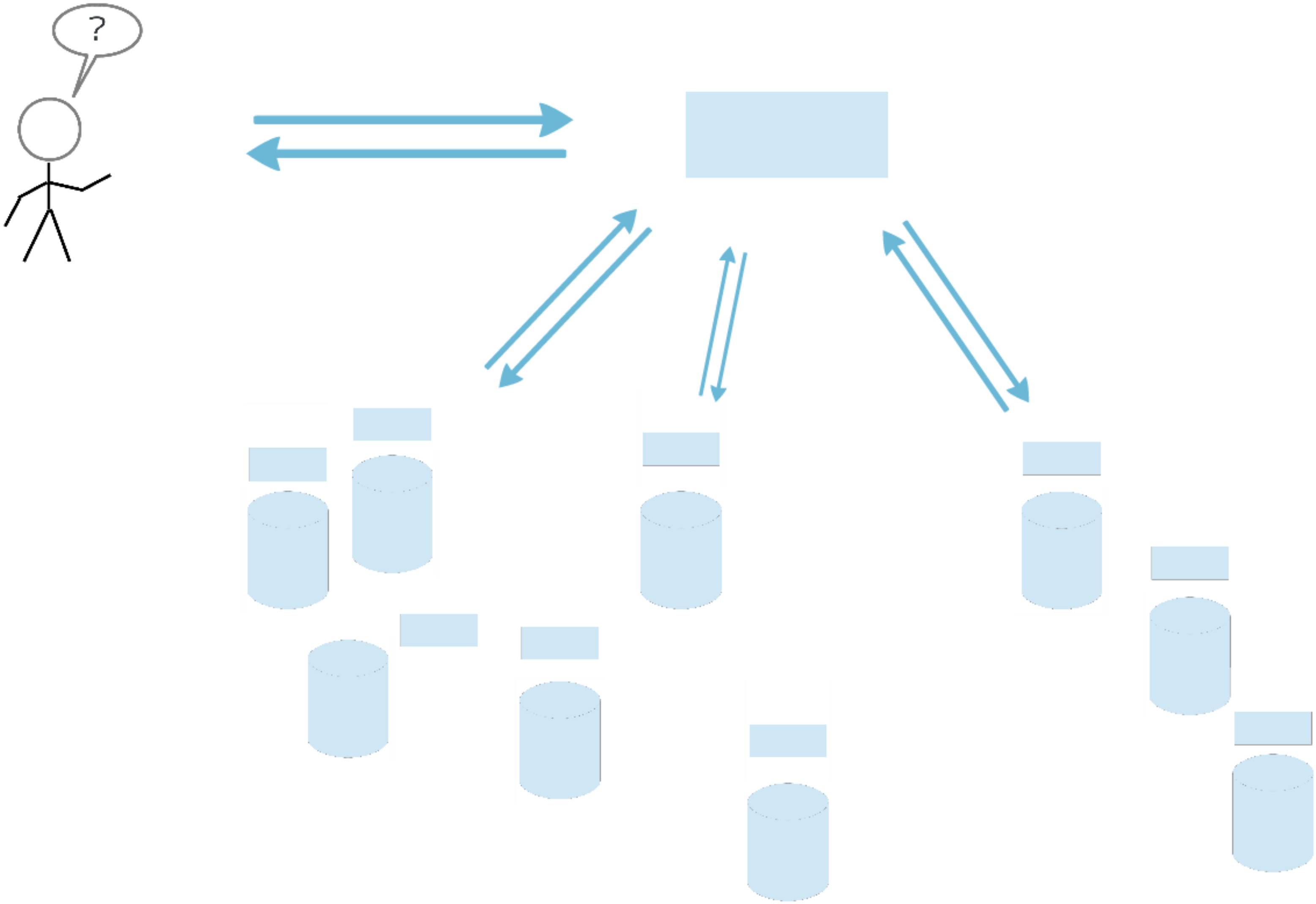
(A-1) Local Cache Approach

gather a subset of RDF data on local computational resources



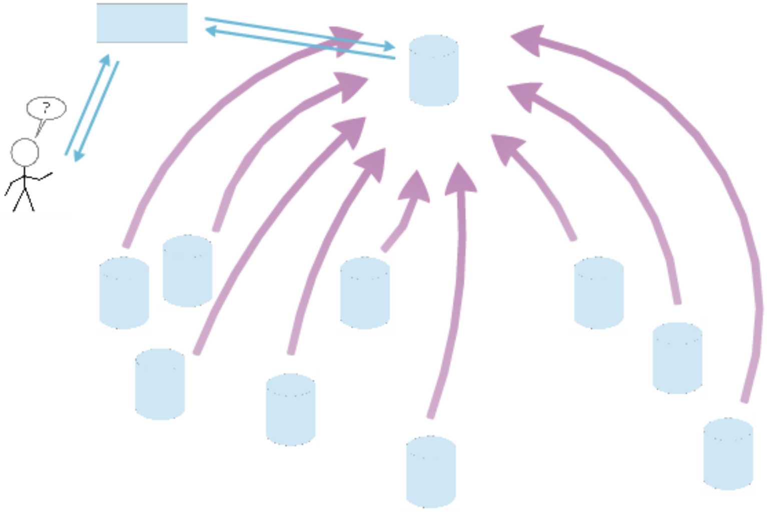
(A-2) Federated Search Approach

distribute sub-queries to several search services distributed over the Internet



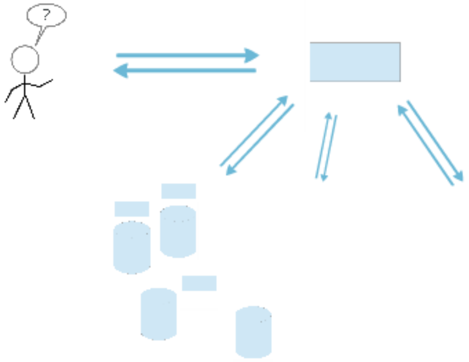
(A-1) Local Cache Approach

gather a subset of RDF data on local computational resources



(A-2) Federated Search Approach

distribute sub-queries to several search services
distributed over the Internet



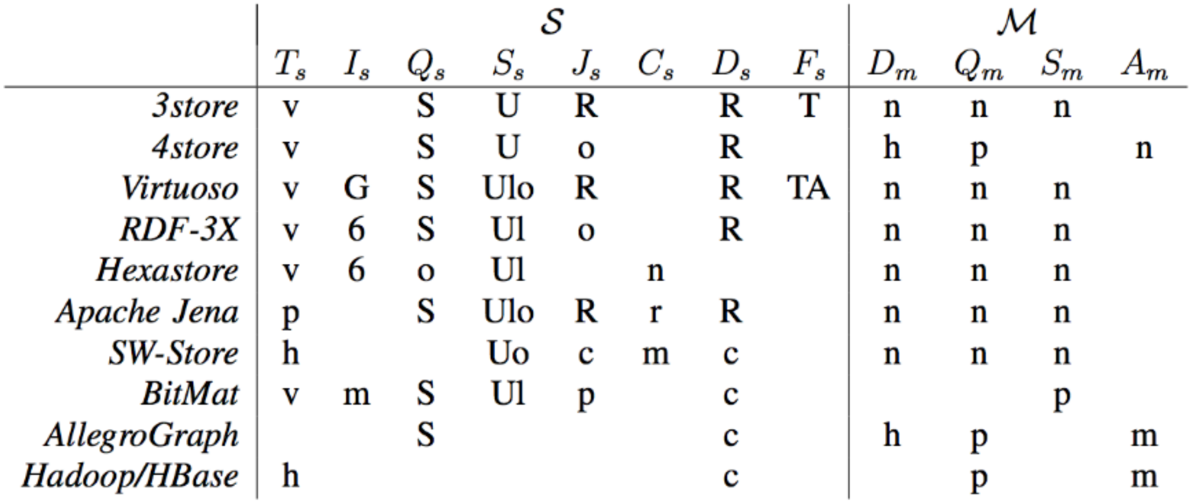
plays an important role for query process efficiency

CLASSIFICATION OF RDF STORAGE MANAGERS

RDF storage managers in the local cache approach can be classified in accordance with several aspects.

$$RSM(\mathcal{S}, \mathcal{M})$$

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

$$\mathcal{S}$$

attributes of single process issues

$$\mathcal{S}(T_s, I_s, Q_s, S_s, J_s, C_s, D_s, F_s)$$

T_s triple table type (vertical | property | horizontal)

I_s index structure type (6-independent | GSPO- OGPS | matrix)

Q_s query type (SPARQL | original)

S_s translation method type of IRI and literal strings (URI | literal | long | none)

J_s join optimization method type (RDBMS-based | column-store-based | conventional-ordering | pruning | none)

C_s cache type (materialized-path-index | reified-statement (r) | none)

D_s dabase engine type (RDB | custom)

F_s inference feature type (TBox, ABox, and no)

$$\mathcal{M}$$

attributes of multiple process issues

$$\mathcal{M}(D_m, Q_m, S_m, A_m)$$

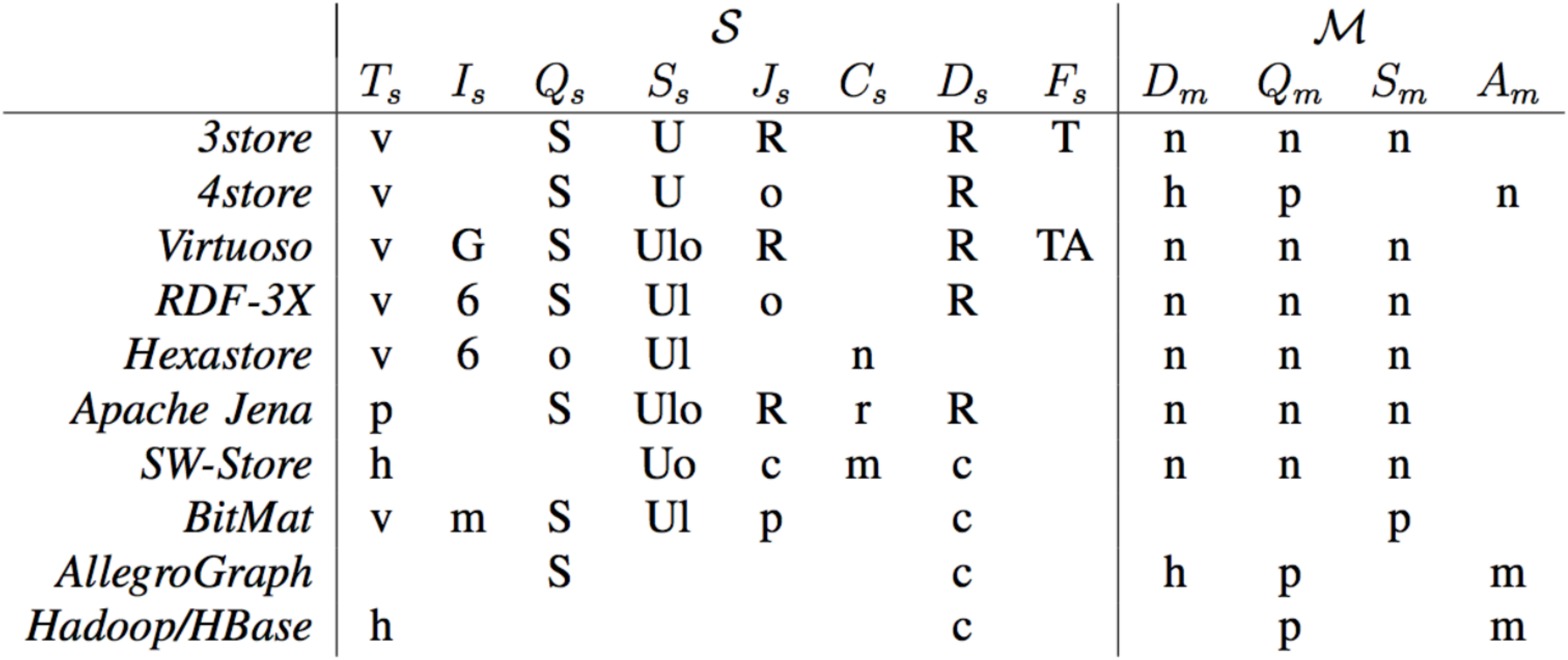
D_m data distribution method type (hash | data-source | none)

Q_m query process distribution method type
(data-parallel | data-replication | none)

S_m stream process type (pipeline | none)

A_m resource sharing architecture type (memory | disk | nothing)

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

RDF STORAGE MANAGERS

3store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox, Dm = none, Qm = none, Sm = none.
- It stored 5,000 classes and about 20 million triples in originally used semantic web applications.
- 3store was implemented on top of the MySQL database management system.

Virtuoso

- Ts = vertical, Is = GSPO OGPS, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox,ABox, Dm = none, Qm = none, Sm = none.
- The most important aspects: extending SQL types with the RDF data type, dealing with unpredictable sizes of objects, providing efficient indexing and extending relational statistics.

Hexastore

- Ts = vertical, Is = 6 independent, Qs = original(customwrapped), Ss = string id, Js = unknown(itseemsnone), Cs = none, Dm = none, Qm = none, Sm = none.
- The proposed index provides natural representation of multi-valued properties and allows fast implementation of merge-join, intersection, and union.

SW-Store

- Ts = horizontal, Ss = string id, Js = column store based, Cs = materialized path index, Ds = custom, Dm = none, Qm = none, Sm = none.
- Storage systems based on columns can significantly improve some types of queries on RDF databases.

AllegroGraph

- Qs = SPARQL, Ds = custom, Dm = hash, Qm = data parallel, Am = memory.
- It is a proprietary products of Franz Inc.
- A clustered version of AllegroGraph stored 1 trillion triples in about 14 days.
- In the scalability ranking of triple stores edited by W3C, AllegroGraph is ranked first as of August 2011.

4store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = hash, Qm = data parallel, Am = nothing.
- The requirements were to store and manage 15x109 triples.
- The design of 4store is based on 3store especially in the way RDF triples are represented.

RDF-3X

- Ts = vertical, Is = 6 independent, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = none, Qm = none, Sm = none.
- It built six independent indexes of SPO, SOP, OSP, OPS, PSO and POS.
- The optimization uses selectivity statistics.

Apache Jena

- Ts = property, Qs = SPARQL, Ss = string id, Js = RDBMS based, Cs = reified statement, Ds = RDB, Dm = none, Qm = none, Sm = none.
- Jena is a database programming language environment based on RDF for Java
- Jena supports MySQL, Postgres, Oracle, and BerkeleyDB.

BitMat

- Ts = vertical, Is = matrix, Qs = SPARQL, Ss = string id, Js = pruning, Ds = custom, Sm = pipeline.
- It used a three dimensional compressed matrix index named BitMat.
- It avoids maintaining materialized triples as much as possible.

Hadoop/HBase

- Ts = horizontal, Ds = custom, Qm = data parallel, Am = memory.
- HBase is column-oriented database system implemented on top of Hadoop.
- HBase is designed for horizontal distribution of tables into regions that are managed by one server. Map-reduce techniques can be employed to process table rows.

3store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox, Dm = none, Qm = none, Sm = none.
- It stored 5,000 classes and about 20 million triples in originally used semantic web applications.
- 3store was implemented on top of the MySQL database management system.

4store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = hash, Qm = data parallel, Am = nothing.
- The requirements were to store and manage 15x109 triples.
- The design of 4store is based on 3store especially in the way RDF triples are represented.

Virtuoso

- Ts = vertical, Is = GSPO OGPS, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox,ABox, Dm = none, Qm = none, Sm = none.
- The most important aspects: extending SQL types with the RDF data type, dealing with unpredictable sizes of objects, providing efficient indexing and extending relational statistics.

RDF-3X

- Ts = vertical, Is = 6 independent, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = none, Qm = none, Sm = none.
- It built six independent indexes of SPO, SOP, OSP, OPS, PSO and POS.
- The optimization uses selectivity statistics.

Hexastore

- Ts = vertical, Is = 6 independent, Qs = original(customwrapped), Ss = string id, Js = unknown(itseemsnone), Cs = none, Dm = none, Qm = none, Sm = none.
- The proposed index provides natural representation of multi-valued properties and allows fast implementation of merge-join, intersection, and union.

Apache Jena

- Ts = property, Qs = SPARQL, Ss = string id, Js = RDBMS based, Cs = reified statement, Ds = RDB, Dm = none, Qm = none, Sm = none.
- Jena is a database programming language environment based on RDF for Java
- Jena supports MySQL, Postgres, Oracle, and BerkeleyDB.

SW-Store

- Ts = horizontal, Ss = string id, Js = column store based, Cs = materialized path index, Ds = custom, Dm = none, Qm = none, Sm = none.
- Storage systems based on columns can significantly improve some types of queries on RDF databases.

BitMat

- Ts = vertical, Is = matrix, Qs = SPARQL, Ss = string id, Js = pruning, Ds = custom, Sm = pipeline.
- It used a three dimensional compressed matrix index named BitMat.
- It avoids maintaining materialized triples as much as possible.

AllegroGraph

- Qs = SPARQL, Ds = custom, Dm = hash, Qm = data parallel, Am = memory.
- It is a proprietary products of Franz Inc.
- A clustered version of AllegroGraph stored 1 trillion triples in about 14 days.
- In the scalability ranking of triple stores edited by W3C, AllegroGraph is ranked first as of August 2011.

Hadoop/HBase

- Ts = horizontal, Ds = custom, Qm = data parallel, Am = memory.
- HBase is column-oriented database system implemented on top of Hadoop.
- HBase is designed for horizontal distribution of tables into regions that are managed by one server. Map-reduce techniques can be employed to process table rows.

RDF STORAGE MANAGERS

3store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox, Dm = none, Qm = none, Sm = none.
- It stored 5,000 classes and about 20 million triples in originally used semantic web applications.
- 3store was implemented on top of the MySQL database management system.

Virtuoso

- Ts = vertical, Is = GSPO OGPS, Qs = SPARQL, Ss = string id, Js = RDBMS based, Ds = RDB, Fs = TBox,ABox, Dm = none, Qm = none, Sm = none.
- The most important aspects: extending SQL types with the RDF data type, dealing with unpredictable sizes of objects, providing efficient indexing and extending relational statistics.

Hexastore

- Ts = vertical, Is = 6 independent, Qs = original(customwrapped), Ss = string id, Js = unknown(itseemsnone), Cs = none, Dm = none, Qm = none, Sm = none.
- The proposed index provides natural representation of multi-valued properties and allows fast implementation of merge-join, intersection, and union.

SW-Store

- Ts = horizontal, Ss = string id, Js = column store based, Cs = materialized path index, Ds = custom, Dm = none, Qm = none, Sm = none.
- Storage systems based on columns can significantly improve some types of queries on RDF databases.

AllegroGraph

- Qs = SPARQL, Ds = custom, Dm = hash, Qm = data parallel, Am = memory.
- It is a proprietary products of Franz Inc.
- A clustered version of AllegroGraph stored 1 trillion triples in about 14 days.
- In the scalability ranking of triple stores edited by W3C, AllegroGraph is ranked first as of August 2011.

4store

- Ts = vertical, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = hash, Qm = data parallel, Am = nothing.
- The requirements were to store and manage 15x109 triples.
- The design of 4store is based on 3store especially in the way RDF triples are represented.

RDF-3X

- Ts = vertical, Is = 6 independent, Qs = SPARQL, Ss = string id, Js = conventional ordering, Ds = RDB, Dm = none, Qm = none, Sm = none.
- It built six independent indexes of SPO, SOP, OSP, OPS, PSO and POS.
- The optimization uses selectivity statistics.

Apache Jena

- Ts = property, Qs = SPARQL, Ss = string id, Js = RDBMS based, Cs = reified statement, Ds = RDB, Dm = none, Qm = none, Sm = none.
- Jena is a database programming language environment based on RDF for Java
- Jena supports MySQL, Postgres, Oracle, and BerkeleyDB.

BitMat

- Ts = vertical, Is = matrix, Qs = SPARQL, Ss = string id, Js = pruning, Ds = custom, Sm = pipeline.
- It used a three dimensional compressed matrix index named BitMat.
- It avoids maintaining materialized triples as much as possible.

Hadoop/HBase

- Ts = horizontal, Ds = custom, Qm = data parallel, Am = memory.
- HBase is column-oriented database system implemented on top of Hadoop.
- HBase is designed for horizontal distribution of tables into regions that are managed by one server. Map-reduce techniques can be employed to process table rows.

CHALLENGES

More varied values with S properties than with M attributes

- Researches so far have succeeded in achieving good performances by developing single process technologies.
- While practical semantic web applications tend to process large-scale data sets, solutions based on data distribution parallelism have become more popular.

Caching techniques have not been researched that much

- Only Apache Jena and SW-Store reported confirming the efficiency of caching techniques.
- Technologies for automatic investigation and classification of processing queries might become important to utilize caching technologies.

Many researches have been working on developing efficient join algorithms with index structures

- This area has a long history in the research of database management systems.
- While the accumulated RDF data-set is rapidly growing and SPARQL queries are basically constructed from joins of triple patterns, join operations will be applied more strongly in semantic web applications.

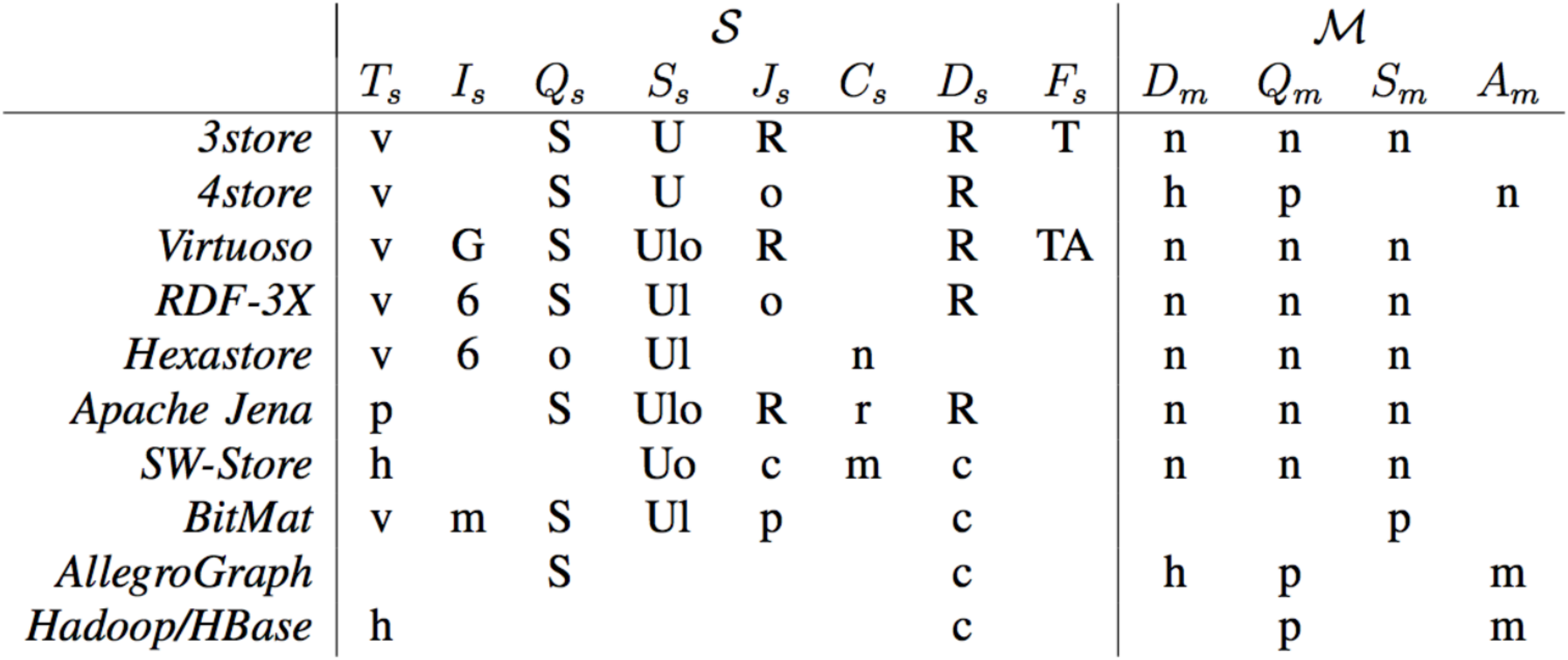
Most RDF storage managers can accept SPARQL queries

- SPARQL-based RDF storage managers rarely cause semantic mismatch due to the existence of RDF algebras described in the W3C recommendation.
- While OPTIONAL operator was introduced to make the query language convenient enough, efficient processing of such queries will be one of the most crucial challenges.

More varied values with S properties than with M attributes

- Researches so far have succeeded in achieving good performances by developing single process technologies.
- While practical semantic web applications tend to process large-scale data sets, solutions based on data distribution parallelism have become more popular.

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

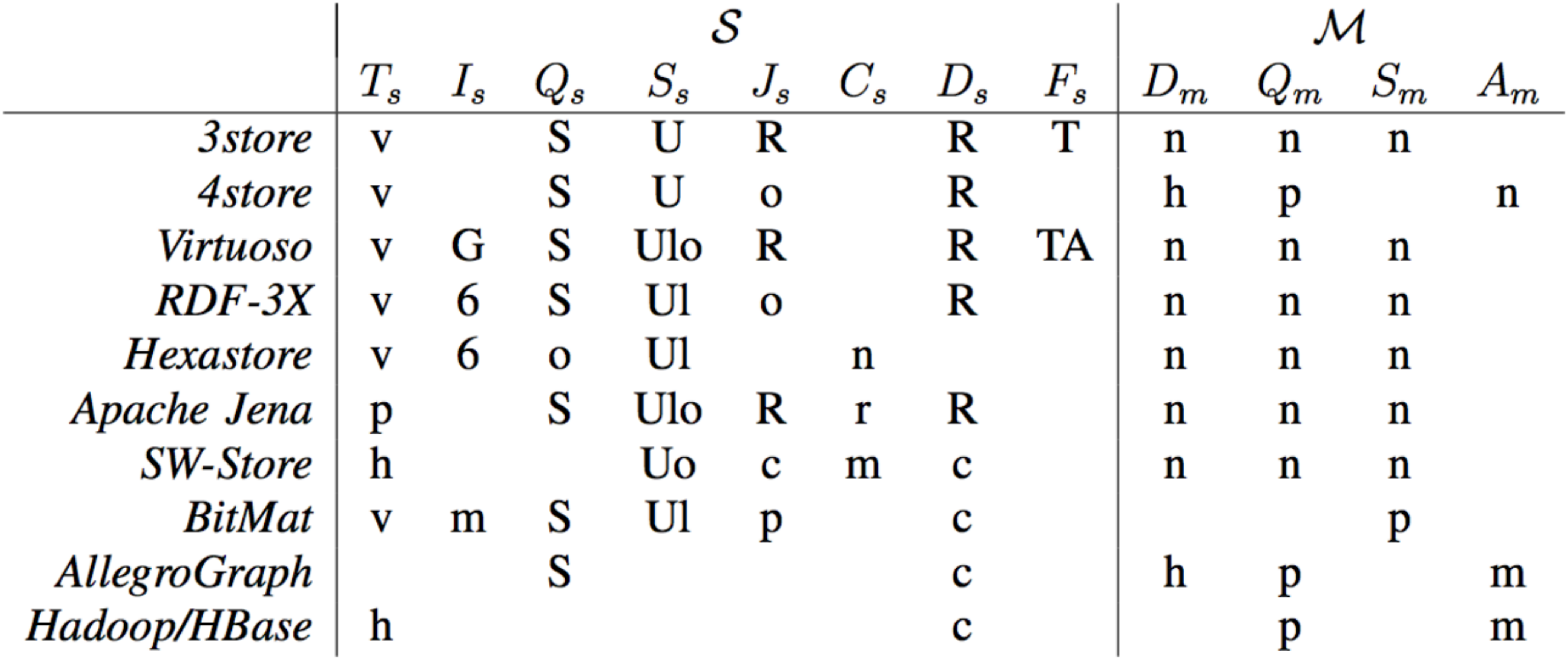
More varied values with S properties than with M attributes

- Researches so far have succeeded in achieving good performances by developing single process technologies.
- While practical semantic web applications tend to process large-scale data sets, solutions based on data distribution parallelism have become more popular.

Caching techniques have not been researched that much

- Only Apache Jena and SW-Store reported confirming the efficiency of caching techniques.
- Technologies for automatic investigation and classification of processing queries might become important to utilize caching technologies.

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

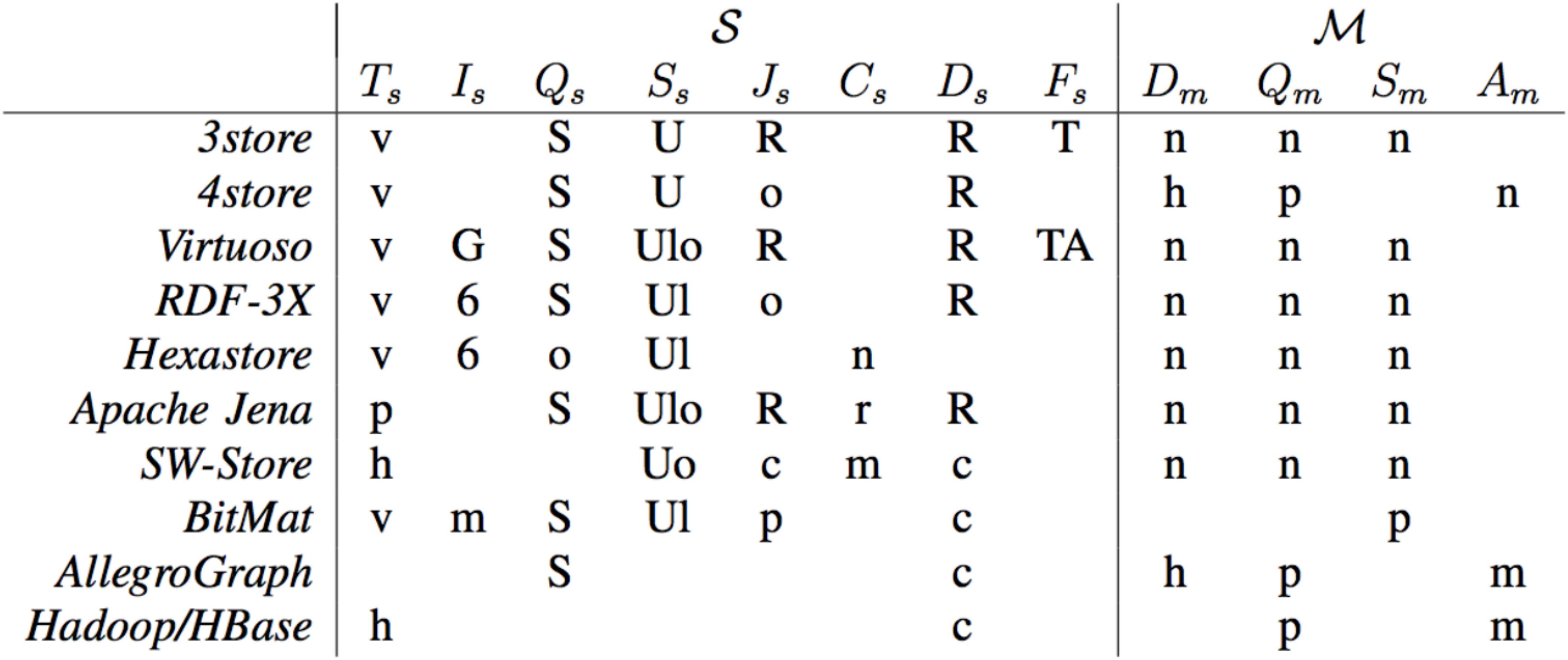
Caching techniques have not been researched that much

- Only Apache Jena and SW-Store reported confirming the efficiency of caching techniques.
- Technologies for automatic investigation and classification of processing queries might become important to utilize caching technologies.

Many researches have been working on developing efficient join algorithms with index structures

- This area has a long history in the research of database management systems.
- While the accumulated RDF data-set is rapidly growing and SPARQL queries are basically constructed from joins of triple patterns, join operations will be applied more strongly in semantic web applications.

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

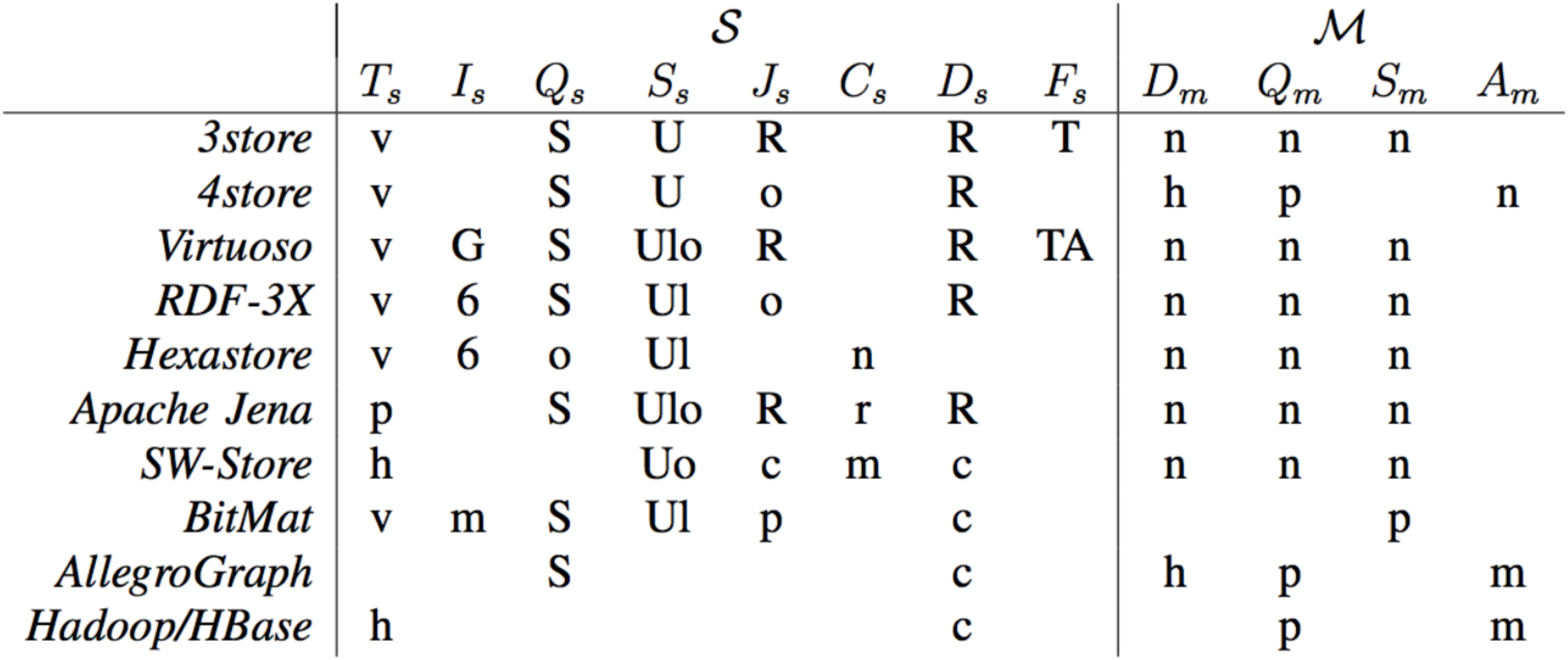
Many researches have been working on developing efficient join algorithms with index structures

- This area has a long history in the research of database management systems.
- While the accumulated RDF data-set is rapidly growing and SPARQL queries are basically constructed from joins of triple patterns, join operations will be applied more strongly in semantic web applications.

Most RDF storage managers can accept SPARQL queries

- SPARQL-based RDF storage managers rarely cause semantic mismatch due to the existence of RDF algebras described in the W3C recommendation.
- While OPTIONAL operator was introduced to make the query language convenient enough, efficient processing of such queries will be one of the most crucial challenges.

PROPERTIES OF RDF STORAGE MANAGERS



	$\mathcal{S}$								$\mathcal{M}$			
	T_s	I_s	Q_s	S_s	J_s	C_s	D_s	F_s	D_m	Q_m	S_m	A_m
3store	v		S	U	R		R	T	n	n	n	
4store	v		S	U	o		R		h	p		n
Virtuoso	v	G	S	Ulo	R		R	TA	n	n	n	
RDF-3X	v	6	S	Ul	o		R		n	n	n	
Hexastore	v	6	o	Ul		n			n	n	n	
Apache Jena	p		S	Ulo	R	r	R		n	n	n	
SW-Store	h			Uo	c	m	c		n	n	n	
BitMat	v	m	S	Ul	p		c				p	
AllegroGraph			S				c		h	p		m
Hadoop/HBase	h						c			p		m

Most RDF storage managers can accept SPARQL queries

- SPARQL-based RDF storage managers rarely cause semantic mismatch due to the existence of RDF algebras described in the W3C recommendation.
- While OPTIONAL operator was introduced to make the query language convenient enough, efficient processing of such queries will be one of the most crucial challenges.

CONCLUSION

- Surveyed the RDF storage manager implementations based on the local cache approach by introducing the systematic classification structure RSM(S,M).
- This classification was applied to 3store, 4store, Virtuoso, RDF-3X, Hexastore, Apache Jena, SW-Store, BitMat, AllegroGraph, and Hadoop/HBase. The list will be expanded.
- There will be room for further improvement of the efficient query process by developing multi process technologies.