Madrid, 2015-09-01

Stream Processing on Demand for Lambda Architectures

European Workshop on Performance Engineering (EPEW) 2015

Johannes Kroß[1], Andreas Brunnert[1], Christian Prehofer[1], Thomas A. Runkler[2], Helmut Krcmar[3]

[1] fortiss GmbH, [2] Siemens AG, [3] Technische Universität München

fortiss GmbH
An-Institut Technische Universität München

Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

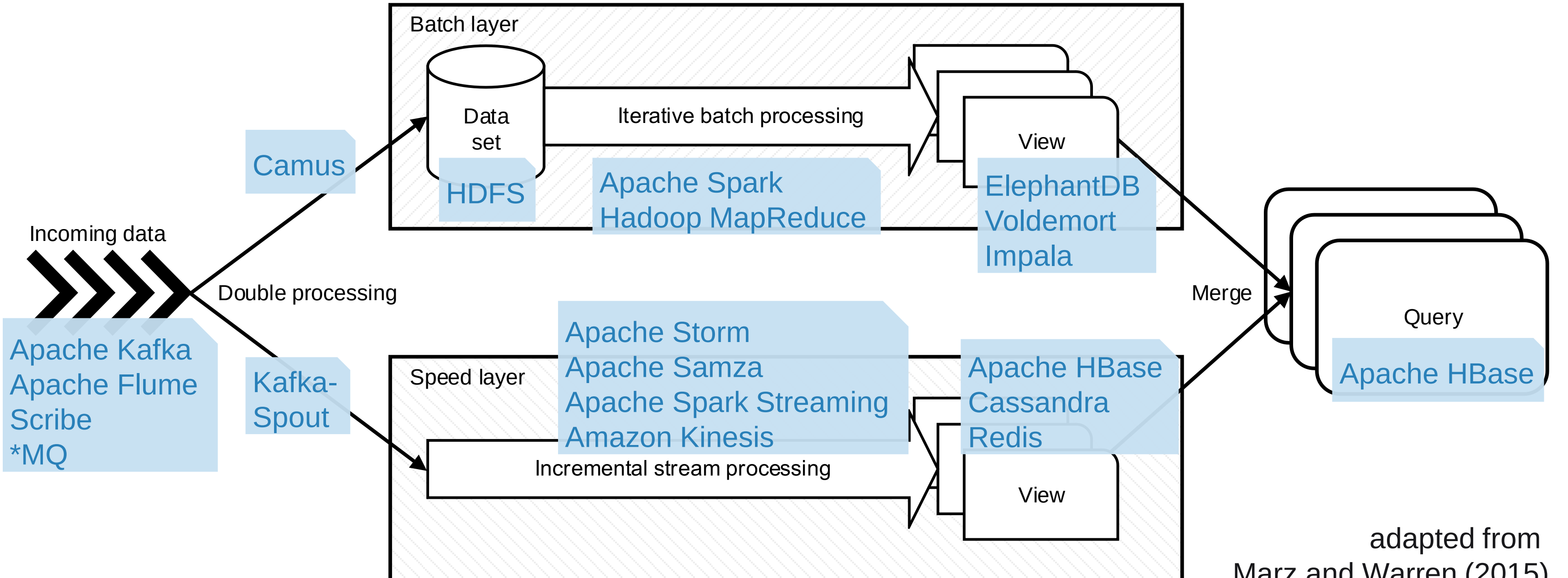
Motivation

- Various complementary big data technologies with different characteristics
- Development of complex system of systems

→ Performance issues and high resource requirements (Brunnert et al. 2014)

Motivation

Data Processing in the Lambda Architecture



adapted from
Marz and Warren (2015)

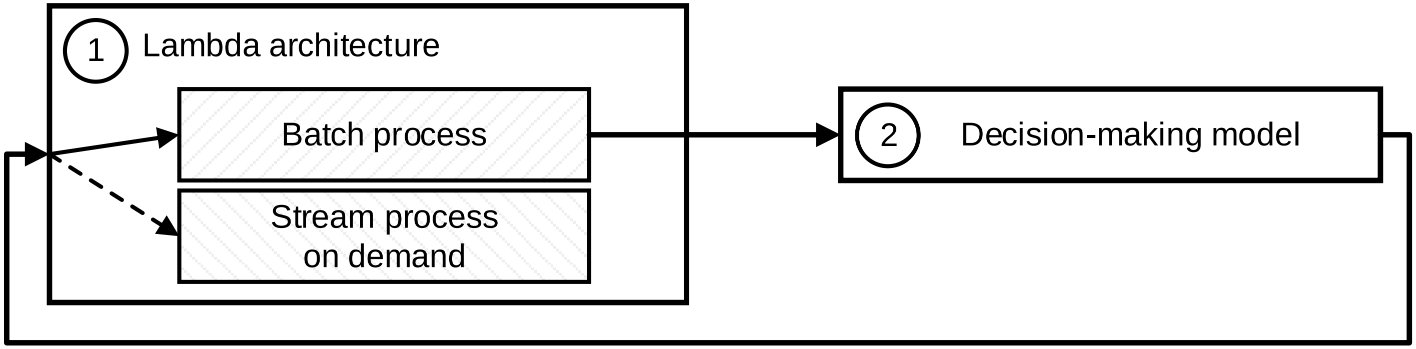
- Enable real-time queries on big data
- Design principles:
  - \+ Data immutability
  - \+ Recomputation
  - \+ Fault-tolerance
  - \- Resource requirements

Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

Stream Processing On Demand

A Novel Approach



Iterative Procedure:

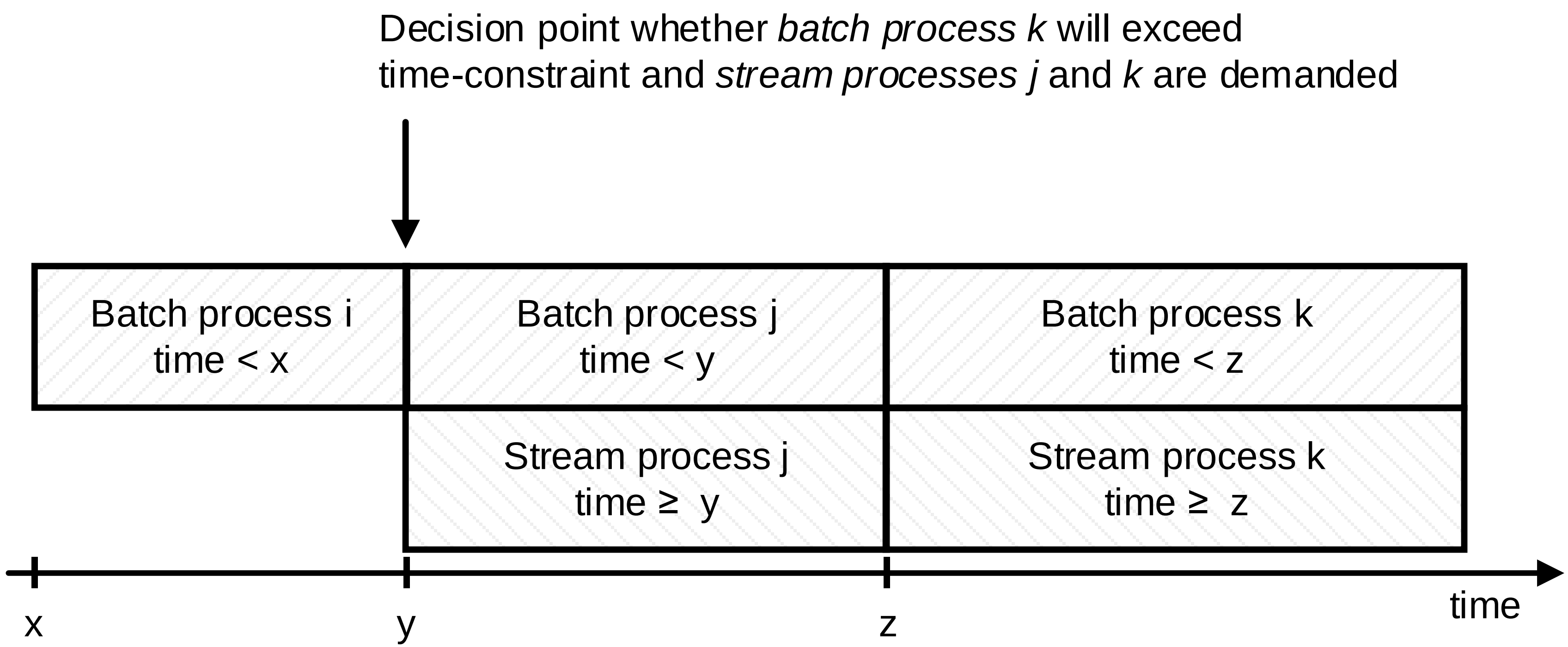
1) Regular batch iteration (in parallel with stream process)

2) Decision-making model

   → Decide if stream processing is additionally required in the next batch iteration

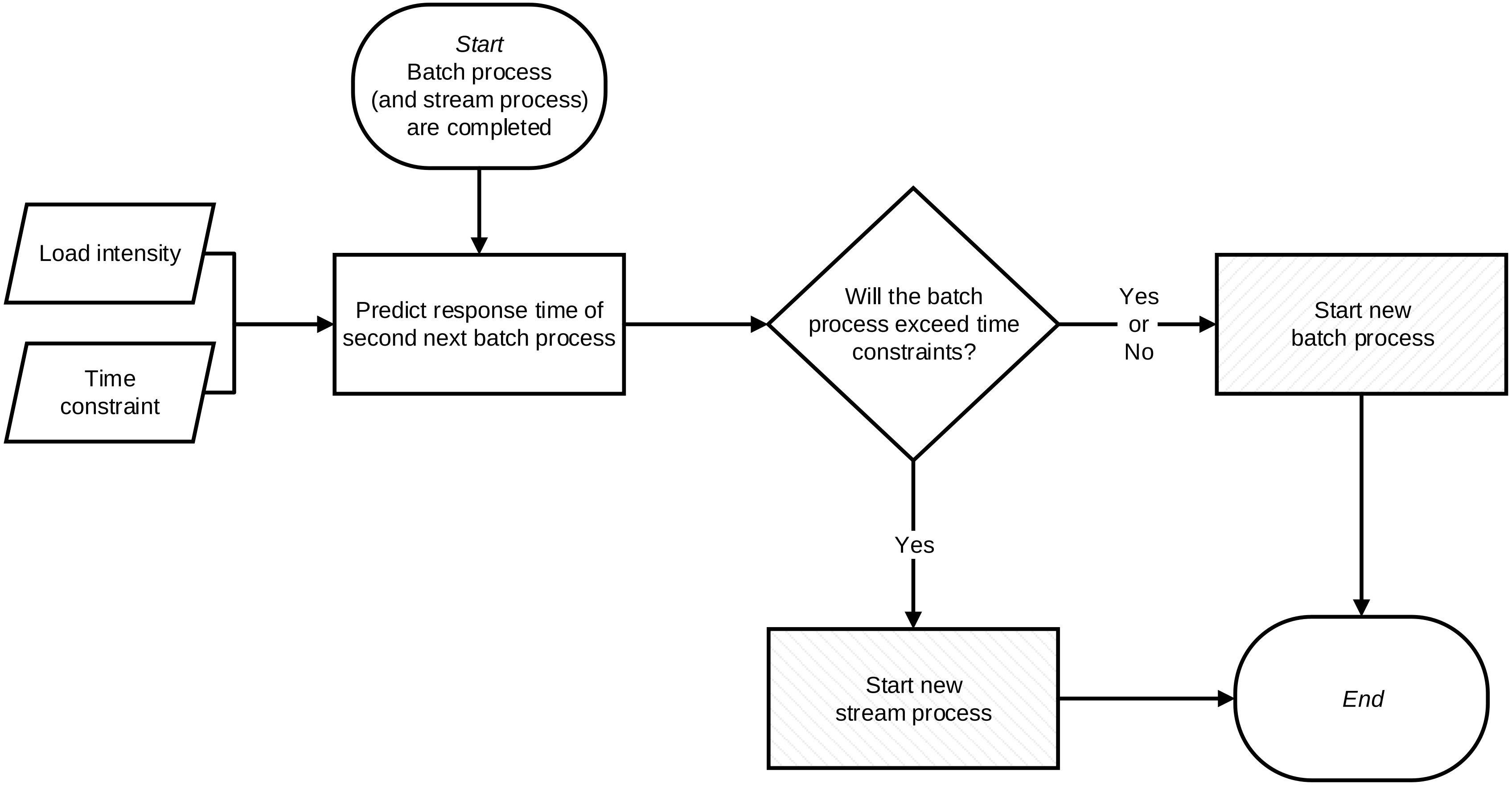
Stream Processing On Demand

Chronological Sequence of Batch and Stream Processes



Stream Processing On Demand

Decision-making Model

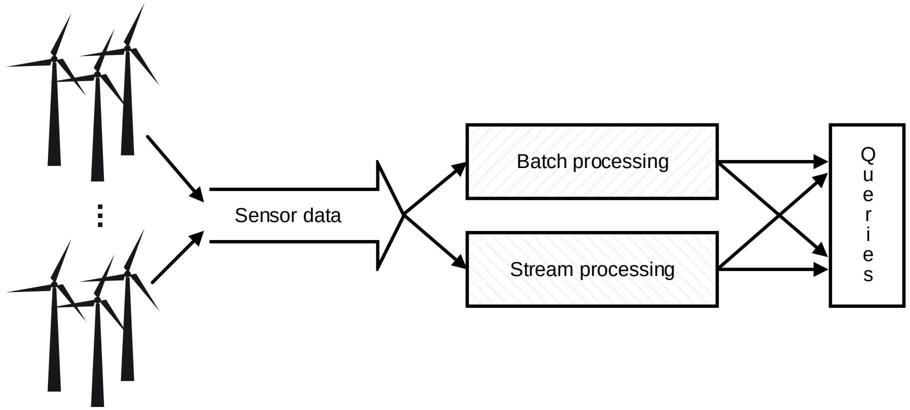


Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

Experimental Validation

Smart Energy Use Case



- Background:
  - Wind turbines (WT) measure several thousand parameters
  - Wind turbine availabilities (WTA) lie between 67.4% and 99% (Faulstich et al. 2011)
- Assumption:
  - Dependent on the WTA, WT either produce a constant set of data or none
    → variable amount of monitoring data
- Example use case:
  - Data are analyzed for bargaining power at the European power exchange
    → time-constraint of 15-minutes for the continuous intraday spot market
  - Development of a data generator to produce random WT monitoring data:

```
id,  timestamp,                  power,  param1,  ...  paramN
12,  2015-04-01 08:23:04.125,    12.67,  value1,  ...  value1
15,  2015-04-01 08:23:03.973,    13.49,  value2,  ...  value2
13,  2015-04-01 08:23:04.096,    12.59,  value3,  ...  value3
...
```

Experimental Validation

Implementation of the Batch Layer

- Apache Hadoop
  - Hadoop Distributed File System (HDFS) to store data sets
  - Hadoop MapReduce for batch processing

  (single node cluster in pseudo-distributed mode)

- Sample analytic batch process:
  - Simple moving average algorithm for a MapReduce job:
    - Map function pseudo code:

```
map ( Object key1 , String value1 ):
    // key1 : file name
    // value1 : measurements of wind turbines of one farm
    for each line l in value :
        kv = parse (l)
        emit ({ kv.id , kv. timestamp }, {kv. timestamp , kv. power })
```

    - Reduce function pseudo code:

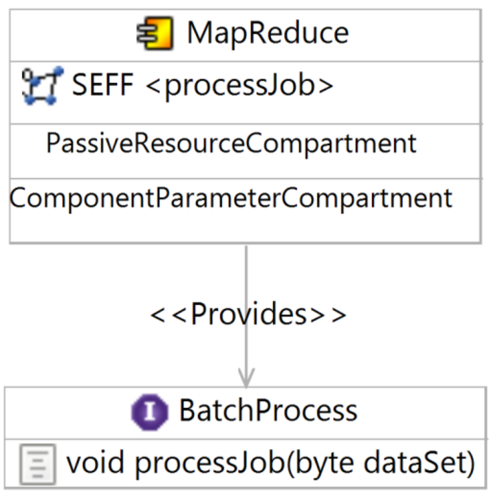
```
reduce ( Object key , Iterator < object > values ):
    // key: an object containing id and timestamp
    // values : power values ordered by timestamp
    result = simpleMovingAverage ( values )
    emit (id , result )
```

Experimental Validation

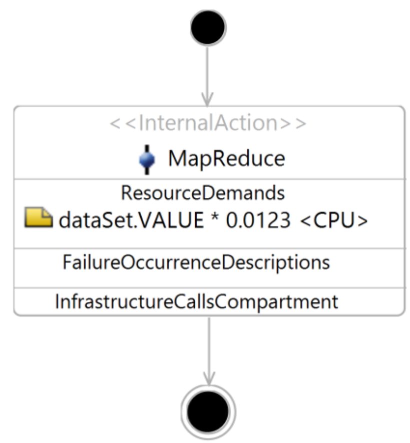
Decision-Making Model & Performance Model Prototype

- Palladio Component Model (PCM) to predict the response time of batch processes
- Integrated measurements for CPU resource demands

Repository model



Service effect specification (SEFF) <processJob>



Experimental Validation

Controlled Experiment

- Generation of monitoring data for 10 wind farms with 100 WT each
- Configuration of a sliding window of 24 hours for the MapReduce job
- Naïve forecast to forecast the workload of the next batch iteration
- Assumption of different WTA to simulate variable load intensities

Experimental Results

Scenario	WTA	Fluctuation	PRT	MRT	RE
1	85 %	± 0 %	12.78 minutes	12.17 minutes	5.01 %
	90 %	± 0 %	13.53 minutes	13.60 minutes	0.51 %
	95 %	± 0 %	14.28 minutes	15.47 minutes	7.69 %
2	85 %	+ 5 %	12.78 minutes	13.82 minutes	7.53 %
	90 %	+ 5 %	13.53 minutes	15.03 minutes	9.98 %
3	90 %	- 5 %	13.53 minutes	12.58 minutes	7.55 %
	95 %	- 5 %	14.28 minutes	13.17 minutes	8.43 %

Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

Related Work

- Lambda architecture
  - Martinez-Prieto et al. (2015) adapt the lambda architecture for semantic data
  - Casado and Younas (2015) give an extensive review about related technologies
  - Aniello et al. (2013) & Rychlý et al. (2015) focus on scheduling stream processes
  - Alrokayan et al. (2015) concentrate on scheduling batch processes

- Several approaches exist to simplify implementing the lambda architecture
  - storm-yarn[1] and Nabi et al. (2014) integrate different stream processing technologies in the Apache Hadoop environment
  - Summingbird[2] is an open source library to write algorithms that can be used for batch as well as stream processing

- Prediction of batch processes
  - Barbierato et al. (2014), Verma et al. (2011), Vianna et al. (2013) provide modeling approaches to predict response times of single MapReduce jobs
  - Castiglione et al. (2014) model the performance for big data applications in cloud infrastructures

[1] https://github.com/yahoo/storm-yarn
[2] https://github.com/twitter/summingbird

Agenda

- Motivation
- Stream Processing On Demand
- Experimental Validation
- Related Work
- Conclusion and Future Work

Conclusion and Future Work

- We introduced a novel approach to ...

  … use resources more efficiently and

  … reduce hardware costs

  when implementing the lambda architecture

- We plan to ...

  … integrate stream processing

  … fully automate our approach

  … apply different workload forecasting techniques

  … conduct a more comprehensive evaluation

References

- Alrokayan, M., Vahid Dastjerdi, A., Buyya, R.: Sla-aware provisioning and scheduling of cloud resources for big data analytics. In: Proceedings of the 2014 IEEE International Conference on Cloud Computing in Emerging Markets. pp. 1-8. IEEE (2014)
- Aniello, L., Baldoni, R., Querzoni, L.: Adaptive online scheduling in storm. In: Proceedings of the 7th ACM International Conference on Distributed Event-based Systems. pp. 207-218. ACM, New York, NY, USA (2013)
- Barbierato, E., Gribaudo, M., Iacono, M.: Performance evaluation of nosql big-data applications using multi-formalism models. Future Generation Computer Systems 37(0), 345-353 (2014)
- Brunnert, A., Vögele, C., Danciu, A., Pfaff, M., Mayer, M., Krcmar, H.: Performance management work. Business & Information Systems Engineering 6(3), 177-179 (2014)
- Casado, R., Younas, M.: Emerging trends and technologies in big data processing. Concurrency and Computation: Practice and Experience 27(8), 2078-2091 (2015)
- Castiglione, A., Gribaudo, M., Iacono, M., Palmieri, F.: Modeling performances of concurrent big data applications. Software: Practice and Experience (2014)
- Faulstich, S., Hahn, B., Tavner, P.J.: Wind turbine downtime and its importance for offshore deployment. Wind Energy 14(3), 327-337 (2011)
- Martnez-Prieto, M.A., Cuesta, C.E., Arias, M., Fernnde, J.D.: The solid architecture for real-time management of big semantic data. Future Generation Computer Systems 47, 62-79 (2015), special Section: Advanced Architectures for the Future Generation of Software-Intensive Systems
- Marz, N., Warren, J.: Big data: principles and best practices of scalable real-time data systems. Manning Publications Co. (2015)
- Nabi, Z., Wagle, R., Bouillet, E.: The best of two worlds: integrating ibm infosphere streams with apache yarn. In: Proceedings of the 2014 IEEE International Conference on Big Data. pp. 47-51. IEEE (2014)
- Rychly, M., Skoda, P., Smrz, P.: Heterogeneity-aware scheduler for stream processing frameworks. International Journal of Big Data Intelligence 2(2), 70-80 (2015)
- Verma, A., Cherkasova, L., Campbell, R.H.: Aria: automatic resource inference and allocation for mapreduce environments. In: Proceedings of the 8th ACM International Conference on Autonomic Computing. pp. 235-244. ACM, New York, NY, USA (2011)
- Vianna, E., Comarela, G., Pontes, T., Almeida, J., Almeida, V., Wilkinson, K., Kuno, H., Dayal, U.: Analytical performance models for mapreduce workloads. International Journal of Parallel Programming 41(4), 495-525 (2013)

CONTACT US

Johannes Kroß

kross@fortiss.org

performancegroup@fortiss.org
pmw.fortiss.org