# DSpace statistics - current status and future development

## Current state as of DSpace 5

### Available engines

- **original statistics**
  - generated from usage events in dspace.log
  - Events captured: TODO
  - Fields captured: TODO
- **Solr statistics** (since DSpace 1.6)
  - uses Solr for storage
  - basic presentation available in the UI; additional commerical modules available; easy to query via HTTP
  - restricted to localhost by default
  - Events captured: TODO
  - Fields captured: type, id, ip, time, epersonid, continent, country, countryCode, city, longitude, latitude, owningComm, owningColl, owningItem, dns, userAgent, isBot, referrer, uid, statistics_type
- **ElasticSearch statistics** (since DSpace 3)
  - Use ElasticSearch for storage
  - goal was to improve performance compared to Solr, because continuous writing of new events had negative impact on concurrent reading
  - implements its own UI for presenting the data; easy to query via HTTP
  - currently doesn't work for bitstream download events
  - exposes unsecured read/write access to ElasticSearch on port 9200 by default
  - Events captured: Item, Bitstream, Collection, Community view
  - Fields captured: IP, time, DNS/hostname, User Agent, isBot flag, geo information (Continent, Country, Country Code, City, Latitude /Longitude)

### Available formats and tools

- dspace.log stores both usage events and log events in general, tends to take up much disk space
- stats-log-converter - tool to filter dspace.log and extract usage events into a statistics.log format
- stats-log-importer - tool to import statistics.log format into Solr statistics; useful for one-time migration from original statistics to Solr statistics; logs don't contain all the fields that Solr record
- stats-log-importer-elasticsearch - analogous to stats-log-importer but imports to ElasticSearch statistics

### Problems

#### Persistence

- dspace.log files, Solr or ElasticSearch index are not suitable for persistent storage
- extracting from dspace.log files takes a long time because they don't contain only usage data
- dspace.log files take up a lot of disk space
- Solr and ElasticSearch indexes are not meant for reliable persistent storage; Solr even says so: http://wiki.apache.org/solr/HowToReindex#Using_Solr_as_a_Data_Source
- historically we have treated Solr indexes as a cache that can be rebuilt from persistent data (search, oai indexes)
- Solr data can be exported, e.g. in CSV; there's a problem with multivalued fields and a trivial export/import may not yield the same result you had before

#### Usage events mixed with errors in dspace.log

- good for debugging (correlated events visible in one place)
- bad for keeping around
- you may want to keep access data forever, because we currently don't have persistent storage
- you likely don't want to keep error, info and debug-level information forever
- filtering is slow

#### Displaying statistics

- DSpace doesn't provide extensive display and visualization options out-of-the-box
- this may be what we want; let others build them

#### Do we even want keeping statistics to be the responsibility of DSpace?

- we already provide a dispatcher/consumer model for events, so it's possible to capture them
- it's possible to write a consumer that will do any serialization, including persistent storage types
- a consumer may be written to export usage events in a standardized format and/or protocol for feeding into specialized systems

### Keeping certain data forever may be against certain laws

- particularly in EU and regarding to storing IP addresses indefinitely; solution would be to only store aggregated or anonymized data indefinitely
- http://security.stackexchange.com/questions/52517/data-protection-laws-and-regulation-for-storing-ip-addresses-for-registered-user

## Possible solutions

### Persistence

- Solr CSV export - this is akin to backup using database dumps, there will always be a time period between last export and now that is not backed up, therefor this should be considered an interim solution
- event consumer - implement an event consumer that writes to a persistent storage, e.g.
    - a file in append mode
    - RDBMS - these are continuous writes, some users might dislike that

### Storage format

- CSV - the data maps naturally to a tabular format. The Solr CSV format might be most convenient as this would ensure interoperability with data previously exported from Solr.
- statistics.log - described above; again, there would be interoperability advantage - we already have existing importers for Solr and ES (though it would need to be extended to include missing fields like geo information and User Agent)
- COUNTER - standardized XML-based format for usage statistics

# Related tickets

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

# Related tools

## Google Analytics

- easy to configure, easy to use service, free of charge tier available
- detail is not unlimited - doesn't let you see individual IP addresses
- possible problems - third-party service without guarantees (SLA available as a paid option), limit on number of processed events per day, possible limit on how many years back they store data

## Piwik

- collects events using a JavaScript snippet (like Google Analytics)
- uses RDBMS for storage
- PHP-based interface, visualizations available

## Logstash

- solution to store and analyze log files (in general, not just for usage data)
- no development needed on the DSpace side to start using Logstash - it works with any log file
- good for correlating data from various sources (e.g. error log with an access log; or logs from distinct systems)

# Related projects

IRUS (predecessors - PIRUS, PIRUS2) - a JISC project (United Kingdom)

- project that collects COUNTER-compliant article-level usage data
- http://www.cranfieldlibrary.cranfield.ac.uk/pirus2/tiki-index.php?page=Project+Plan+and+Progress

SCEUR (Portugal)

- http://sceur.rcaap.pt/index_en.html
- http://projeto.rcaap.pt/index.php/lang-en/sobre-o-rcaap/servicos/sceur