Production data export and archiving system for the BaBar experiment

Artem Trunov

Stanford Linear Accelerator Center

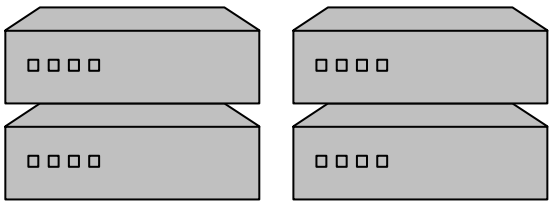
for the BaBar Computing Group

Data Production in BaBar

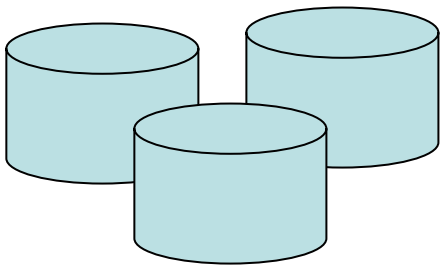
- Taking data for 5 years
  - 931 TB in Objectivity/DB based (“Objy”) event store, 120 federations
  - 143 TB in ROOT based (“Kanga”) event store since Dec 2003 (including old converted data)
- Distributed production
  - Event reconstruction done 100% in INFN (Italy)
  - Skimming done at In2p3, Karlsruhe, Padova
  - Monte Carlo production at ~25 sites in the US and other countries.

BaBar Production

Objectivity/DB

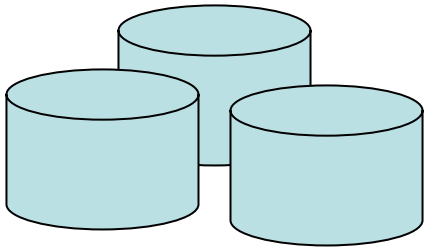


Lockservers, Catalog,
Journal servers
(All servers total - 120)

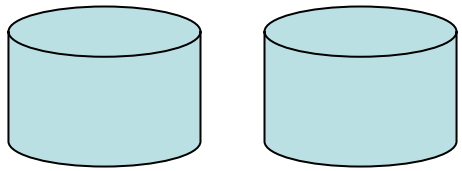


Disk space – 70 TB
on 20 servers

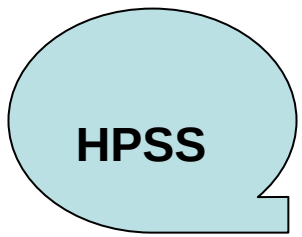
Common



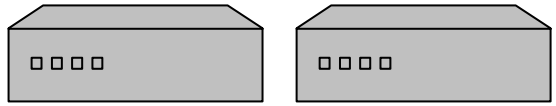
Import/Export servers (7)



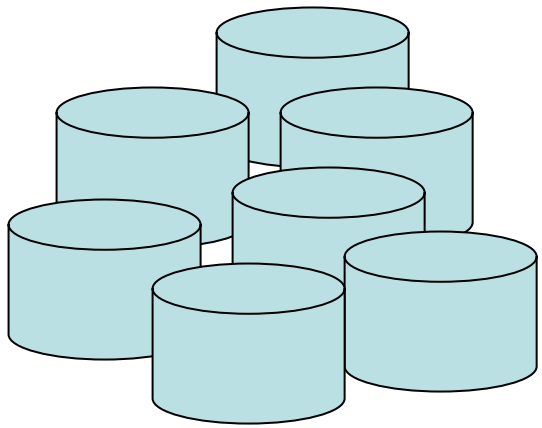
CDB – Objy based
conditions servers (6)



Kanga



Load balancers (2)



Total Disk – 200TB
(including production)
on 42 servers

- **Data volume: Objy – 931 TB, Kanga 143 TB**

Past experience with data management (Objy)

- DB Administration: 2-4 people at different times
- Different import procedures for data produced at SLAC and at remote sites
- No higher level tools to check data consistency
  - Database files could be checked with Objy utilities, but very slow
  - No tools to check collections
- Very little automation of error handling
- Can not control HPSS logic
  - Proprietary code
  - Tape mount order
  - Files write and read order
  - Experience with HPSS congestion under heavy load from 60 servers

Motivation

- Full automation of data transfers and archiving.
- Unify all export and archiving procedures
- Reduce human involvement in error handling
- HPSS-friendly system
- Low resource utilization, with the focus on disk bandwidth
- Protocol level backward compatibility with Objy transfer tools.
- Streamlined procedures for further data processing
- Assisting with exporting data to remote institutions

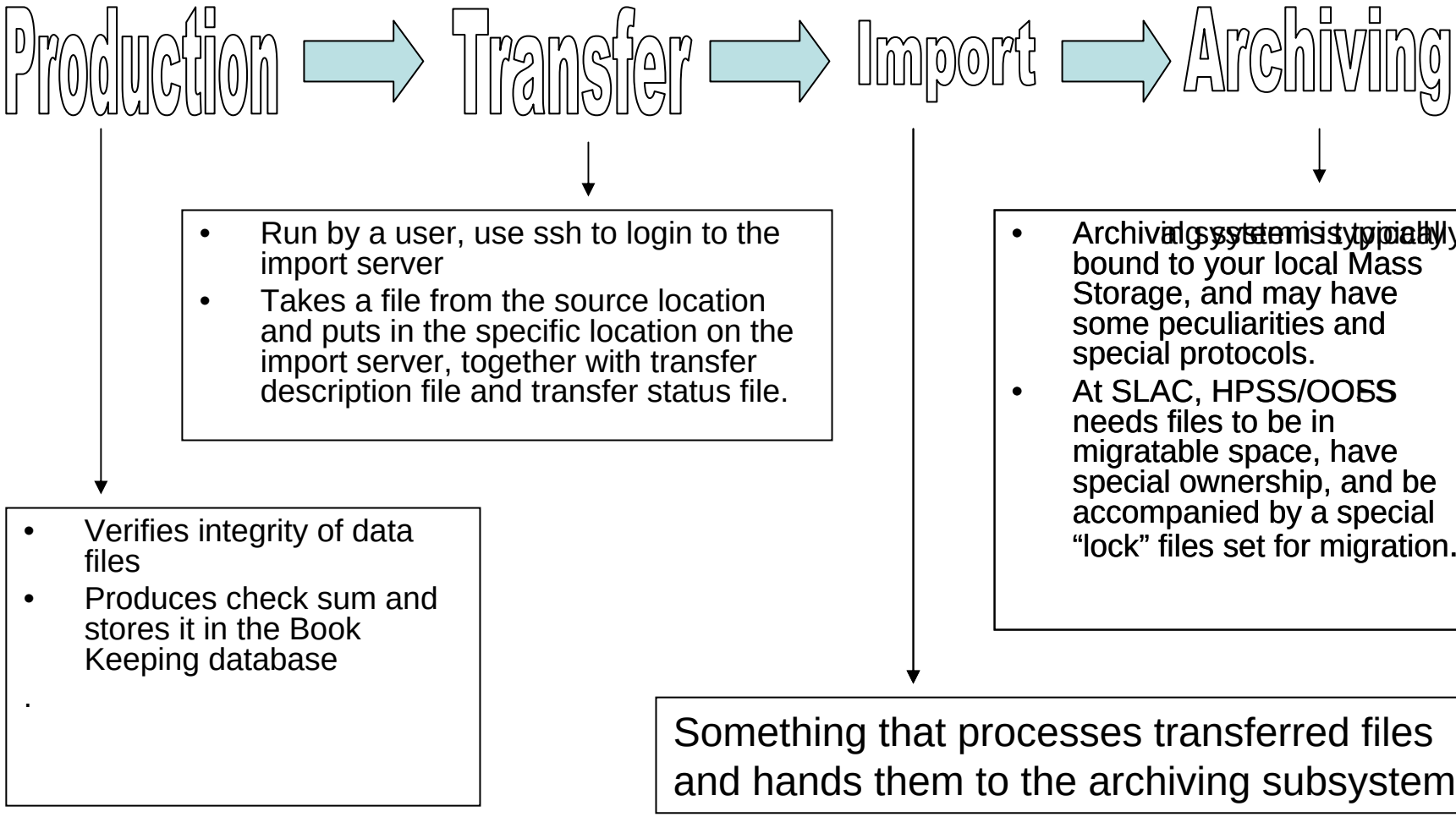
KanTransfer: what this tool is

- Set of client and server side tools written in Perl.
- Ssh authorization for clients.
- Transferring kanga and objy collections
  - As set of files or a tar archive
- Support for custom copy programs
  - bbcp, bbftp used in BaBar (server-less tools)
- Checksum verification
  - Compute or use supplied checksum from Book Keeping database
- Import is detached from transferring
  - Allows accumulating of data on import servers in case of problems.
- Transfer configuration as a Web service.
  - Import side decides where to place data based on the content and server load
- Easily scalable
  - Only need to add more import servers and change the configuration.

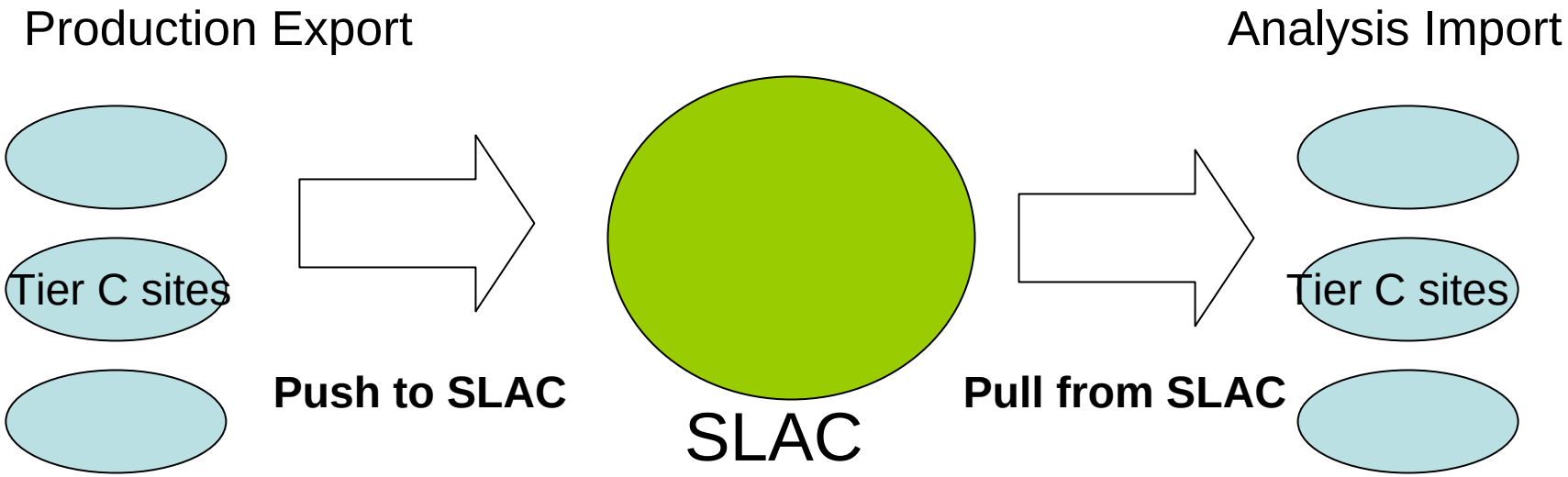
What this tool isn’t

- Not a grid tool or service.
- Not a full-featured data replication service
- Not a heavy-weight one-size-fit-all tool.
- Does not have a replica catalog or another database
- Does not require any daemon or agent on ether client or server side.

Sharing responsibilities, concerns between subsystems

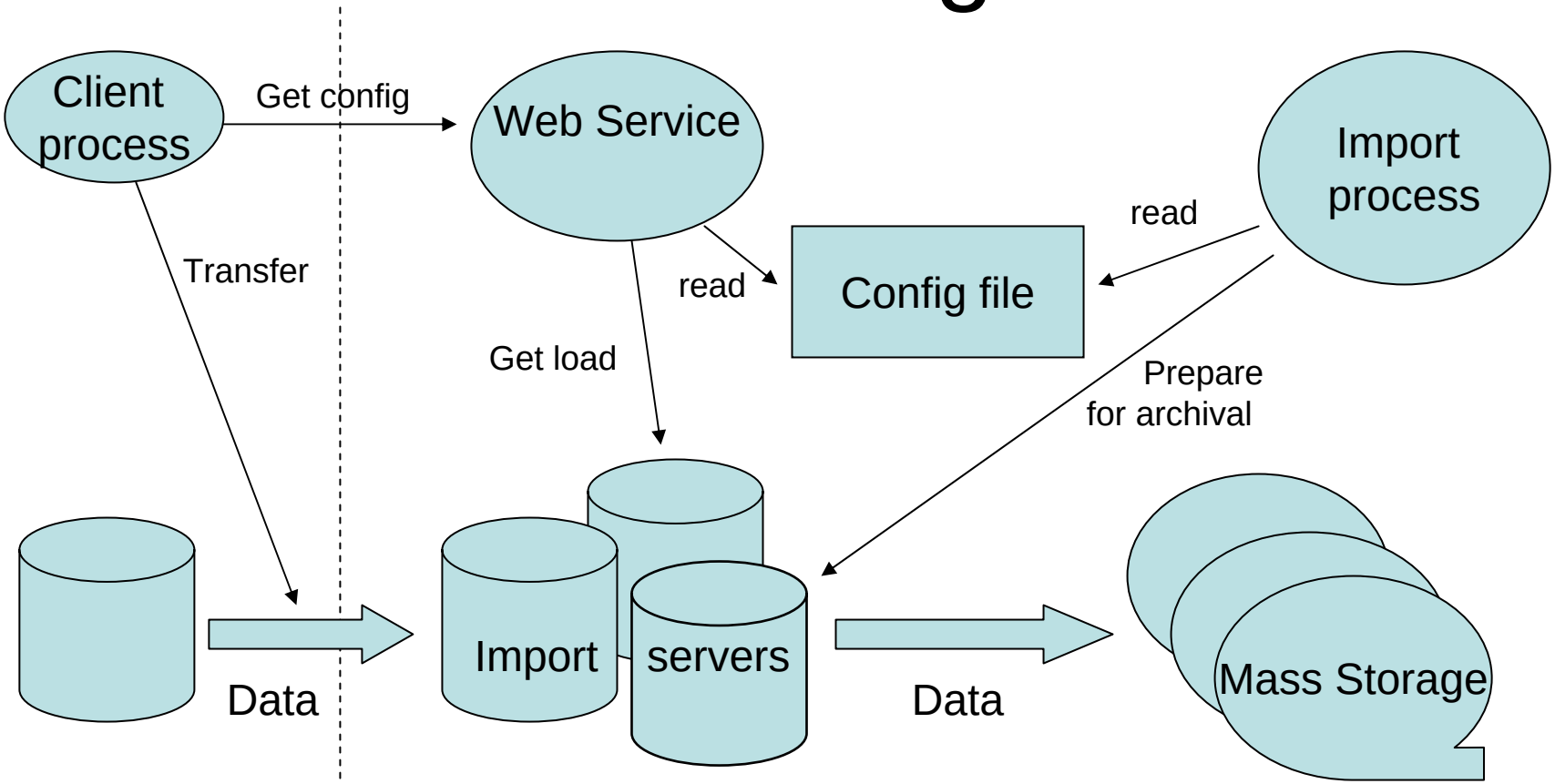


Push – Pull



- In this Tier A site centric model it's easier to manage data.
- SLAC is expected to be available 24/7 and provide timely support in case of incidents.
- Production at smaller Tier C institutions is typically run by 1 part time person, who can import/export when it's convenient.
- SLAC doesn't need to account for site's outages, failures and other things affecting production and transfers.

Central management



- Single configuration file for all transfers
- All transfer parameters are managed by the receiving side
  - Target host, directory, transfer parameters.
  - Can transparently change server allocation due to failure or scheduled outage

Setup at SLAC

- 5 identically configured import servers
- Each has 3 x 0.5TB filesystems with OOFS layer for HPSS access.
- OOFS automatically archives files to HPSS, purges when necessary to make a room for new data.
- Only few servers write to HPSS – helps to reduce it’s load.
- Import is run infrequently, allowing to accumulate some data before archiving to HPSS and reduce tape mounts; and synchronously on all import servers in order to cluster relevant data on tapes.
- Import latency is 0.5-4 hours. (time between end of transfer and availability of files for users).
- Disc cache is kept 90% full at all times, making more files available online for import by external sites.
- After archival, files are copied to read-only analysis pool, and immediately available for users.

Work in Progress

- Load-adaptive transfer
  - Adjusting network options (number of streams, TCP window) based on real transfer performance
  - Varying them and “learning” to achieve best performance
  - Taking into account *real* disk/network conditions at the time of transfer
- Failure report
  - Way for admin to debug and communicate user’s problem.
  - All client’s configuration and errors are logged to the import server.
- Tracking status of a transfer as it passes through all stages

Summary

- Simple tool to transfer and archive BaBar's data.
- In production since the end of 2003.
- All BaBar's kanga data is archived via KanTransfer.
- After initial deployment difficulties, running pretty smooth.
- Requires little attention from SLAC admins.
- Certain requested features been or being implemented.