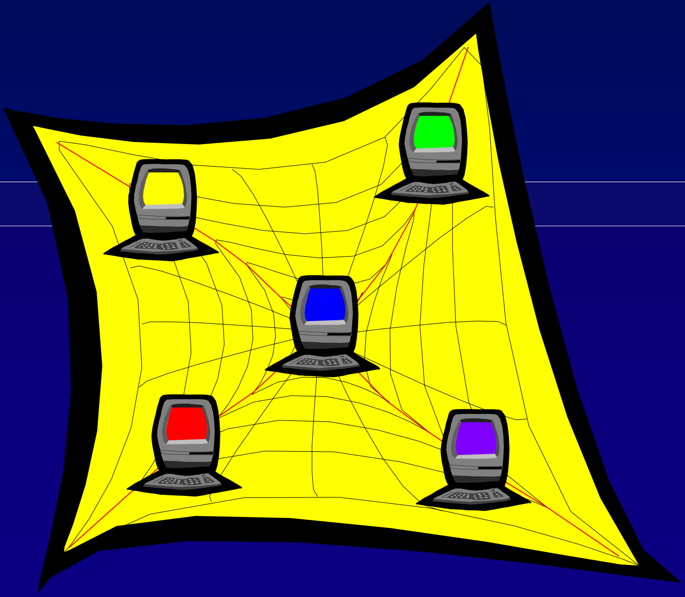
Scheduler

A Distributed Computing Environment



U.S. National Library of Medicine

Agenda

- History
- Overview
- Usage Summary
- Future Direction
- Underlying Programs

History

- System required to speed-up the yearly reprocessing of large collections.
- Tried Parallel Virtual Machine (PVM) and Condor (7/98).
- Moved to our own custom built program starting in 5/99.
- Reasons for Move:
  - ✓ Problems with Condor not working during system backups
  - ✓ Wanted ability to run more than 1 job at a time
  - ✓ Wanted more control over how things were run

Design Goals

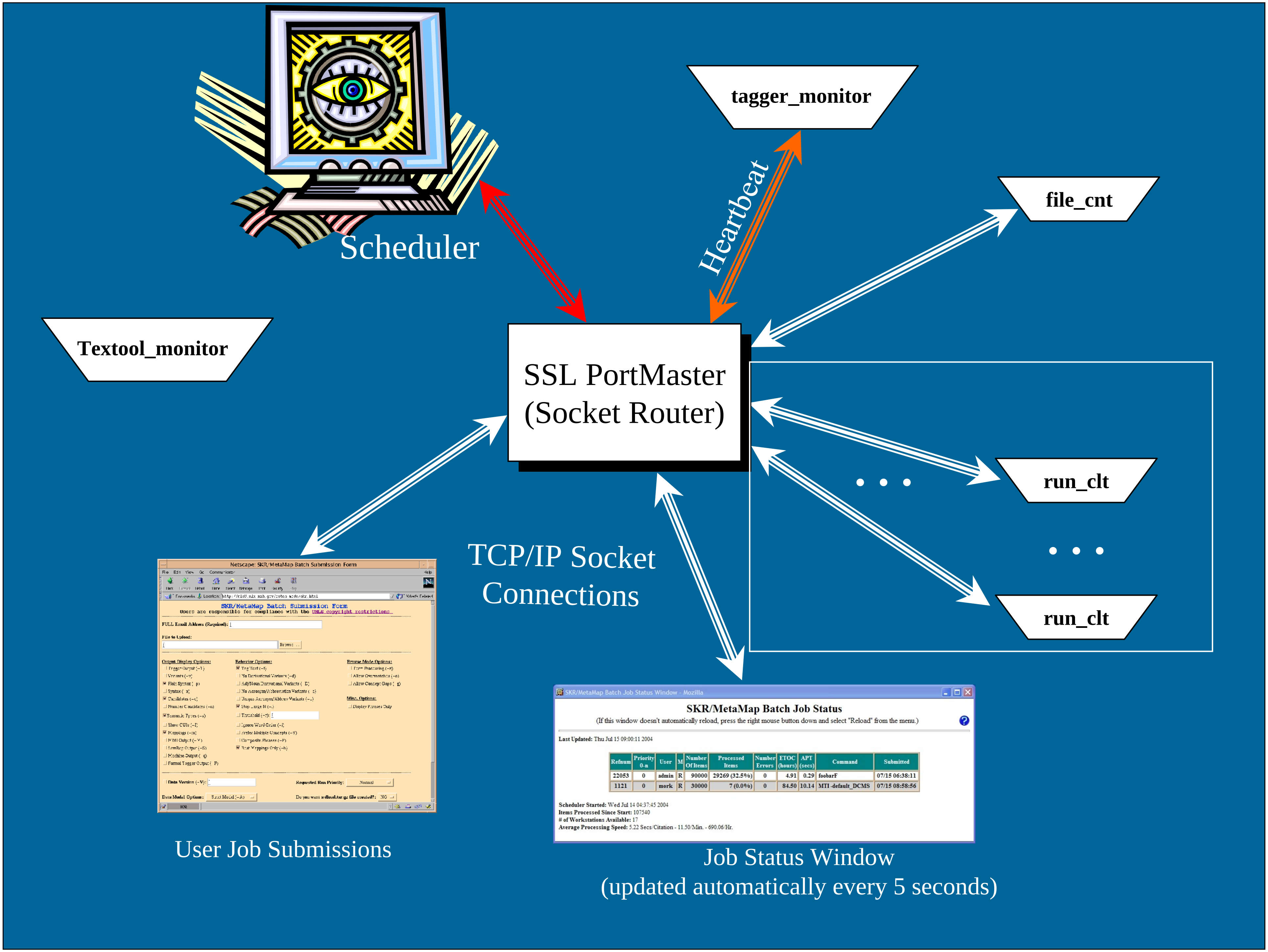
- Provide easy to use web-based access to the facility
- Use unused computing cycles on workstations
- Allow users to schedule access to their workstation
- Provide fault tolerant system to reduce potential for data loss
- Allow multiple jobs to run at the same time versus having to sequence jobs
- Ensure user and data security
- Has to be extensible
- Has to be easy to administer and do much automatically

Technologies Used

- “The Simple Sockets Library” (SSL) from NASA Goddard Space Flight Center’s Intelligent Robotics Laboratory (IRL). SSL (written in C) “C-like” wrapper to TCP/IP sockets
- Main programs written in C
- CGI scripts, HTML, JavaScript, and Perl for web-based interface
- Apache HTTP server with Server Side Includes and security (user authentication and data access)

Highlights

- User’s can schedule when their workstation is used in the pool
- Single non-root user (condor) has access to all workstations within pool
- Users are emailed upon completion of job and told status of job and how to retrieve results
- There is a behind the scenes “scrubber” program to remove jobs after 15 days
- Easy to add new programs as long as they conform to our standard input and output API.
- Currently have 18 workstations in our pool


Scheduler
tagger_monitor
Heartbeat
file_cnt
Textool_monitor
SSL PortMaster
(Socket Router)
run_clt
• • •
• • •
run_clt
TCP/IP Socket
Connections
Netscape: SKR/MetaMap Batch Submission Form
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://ii.nlm.nih.gov/batch_mode/skr.html What's Related
Browse...
100%
SKR/MetaMap Batch Job Status Window - Mozilla

SKR/MetaMap Batch Submission Form

Users are responsible for compliance with the UMLS copyright restrictions

FULL Email Address (Required):

File to Upload:

Output Display Options:

- □ Tagger Output (-T)
- □ Variants (-v)
- ☑ Plain Syntax (-p)
- □ Syntax (-x)
- ☑ Candidates (-c)
- □ Number Candidates (-n)
- ☑ Semantic Types (-s)
- □ Show CUIs (-I)
- ☑ Mappings (-m)
- □ MMI Output (-N)
- □ SemRep Output (-S)
- □ Machine Output (-q)
- □ Formal Tagger Output (-F)

Behavior Options:

- ☑ Tag Text (-t)
- □ No Derivational Variants (-d)
- □ Adj/Noun Derivational Variants (-D)
- □ No Acronym/Abbreviation Variants (-a)
- □ Unique Acronym/Abbrev Variants (-u)
- ☑ Stop Large N (-l)
- □ Threshold (-r):
- □ Ignore Word Order (-i)
- □ Prefer Multiple Concepts (-Y)
- □ Composite Phrases (-P)
- ☑ Best Mappings Only (-h)

Browse Mode Options:

- □ Term Processing (-z)
- □ Allow Overmatches (-o)
- □ Allow Concept Gaps (-g)

Misc. Options:

- □ Display Phrases Only

□ **Data Version** (-V):

Requested Run Priority: Normal

Data Model Options: Strict Model (-A)

Do you want a **dload.tar.gz file created?:** NO

User Job Submissions

SKR/MetaMap Batch Job Status

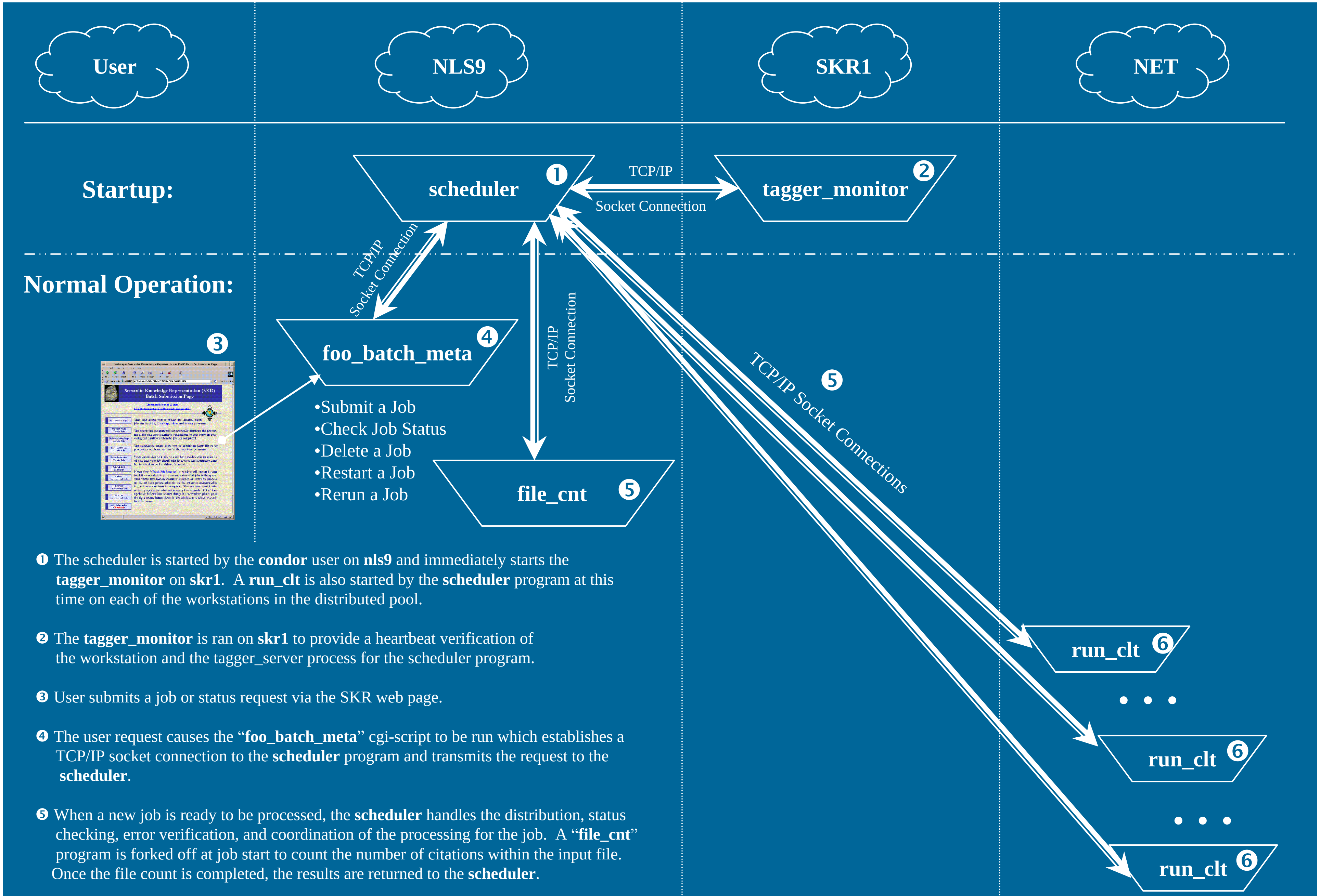
(If this window doesn't automatically reload, press the right mouse button down and select "Reload" from the menu.)

Last Updated: Thu Jul 15 09:00:11 2004

Refnum	Priority 0-n	User	M	Number Of Items	Processed Items	Number Errors	ETOC (hours)	APT (secs)	Command	Submitted
22053	0	admin	R	90000	29269 (32.5%)	0	4.91	0.29	foobarF	07/15 06:38:11
1121	0	mork	R	30000	7 (0.0%)	0	84.50	10.14	MTI -default_DCMS	07/15 08:58:56

Scheduler Started: Wed Jul 14 04:37:45 2004
Items Processed Since Start: 107540
# of Workstations Available: 17
Average Processing Speed: 5.22 Secs/Citation - 11.50/Min. - 690.06/Hr.

Job Status Window
(updated automatically every 5 seconds)



- Submit a Job
- Check Job Status
- Delete a Job
- Restart a Job
- Rerun a Job

❶ The scheduler is started by the **condor** user on **nls9** and immediately starts the **tagger_monitor** on **skr1**. A **run_clt** is also started by the **scheduler** program at this time on each of the workstations in the distributed pool.

❷ The **tagger_monitor** is ran on **skr1** to provide a heartbeat verification of the workstation and the tagger_server process for the scheduler program.

❸ User submits a job or status request via the SKR web page.

❹ The user request causes the "**foo_batch_meta**" cgi-script to be run which establishes a TCP/IP socket connection to the **scheduler** program and transmits the request to the **scheduler**.

❺ When a new job is ready to be processed, the **scheduler** handles the distribution, status checking, error verification, and coordination of the processing for the job. A "**file_cnt**" program is forked off at job start to count the number of citations within the input file. Once the file count is completed, the results are returned to the **scheduler**.

❻ The **run_clt** runs the specified application (SKR, MetaMap, Edgar, or Arbiter on the individual citation that was given to it by the **scheduler** program and returns the results to the **scheduler** program.

Fault Tolerance

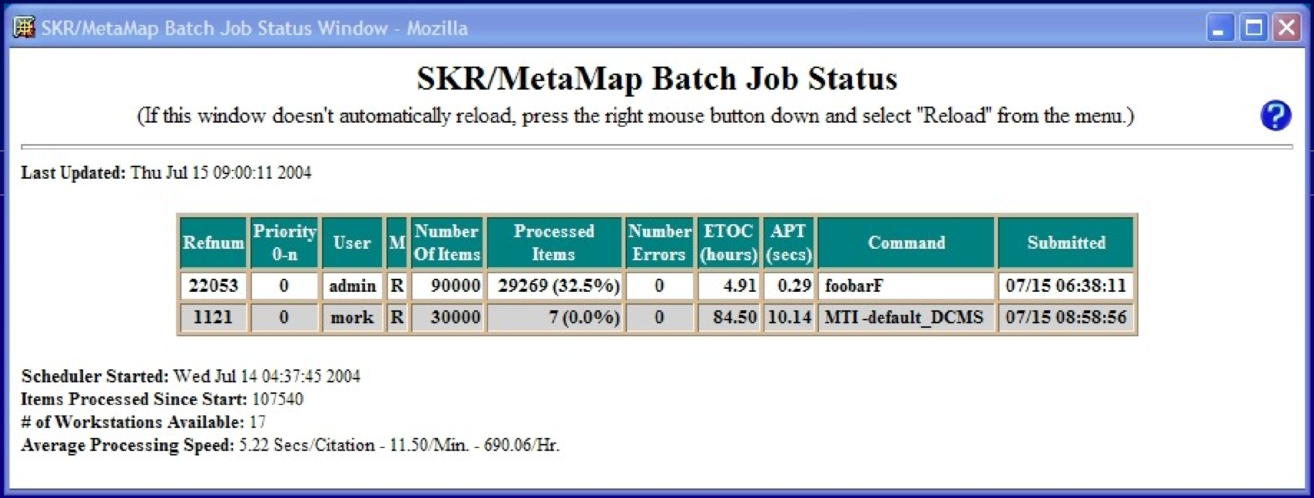
- Tagger monitor serves as a monitor checking that the tagger_server is still available. Scheduler pauses if tagger_server goes down
- If workstation goes down, jobs are resubmitted to other workstations. Workstations automatically added back in.
- If problems with item, retried twice, then marked as error
- Email notification to admin if any problems
- Results file check-pointed whenever proper sequenced items are returned
- Queue check-pointed frequently to allow resume on restart

User Jobs

- Suite of known programs (SKR, MetaMap, MTI, Edgar, Arbiter, and PhraseX) with specific logic and interfaces.
- Two Generic Modes
  - With or Without EOT Validation
  - Must reside in /nfsvol/nls/bin for security reasons
- Remotely view interim/final results and status of job
- Suspend, Resume, or Rerun jobs (user owned)
- Priority Scheme (Normal, Medium (10:1), and High (100% of queue). Medium and High are reserved.

Batch Job Status Window

- Shows real-time status of all jobs in the queue
- Provides “estimates” of completion time
- Shows snapshot of performance



SKR/MetaMap Batch Job Status

(If this window doesn't automatically reload, press the right mouse button down and select "Reload" from the menu.)

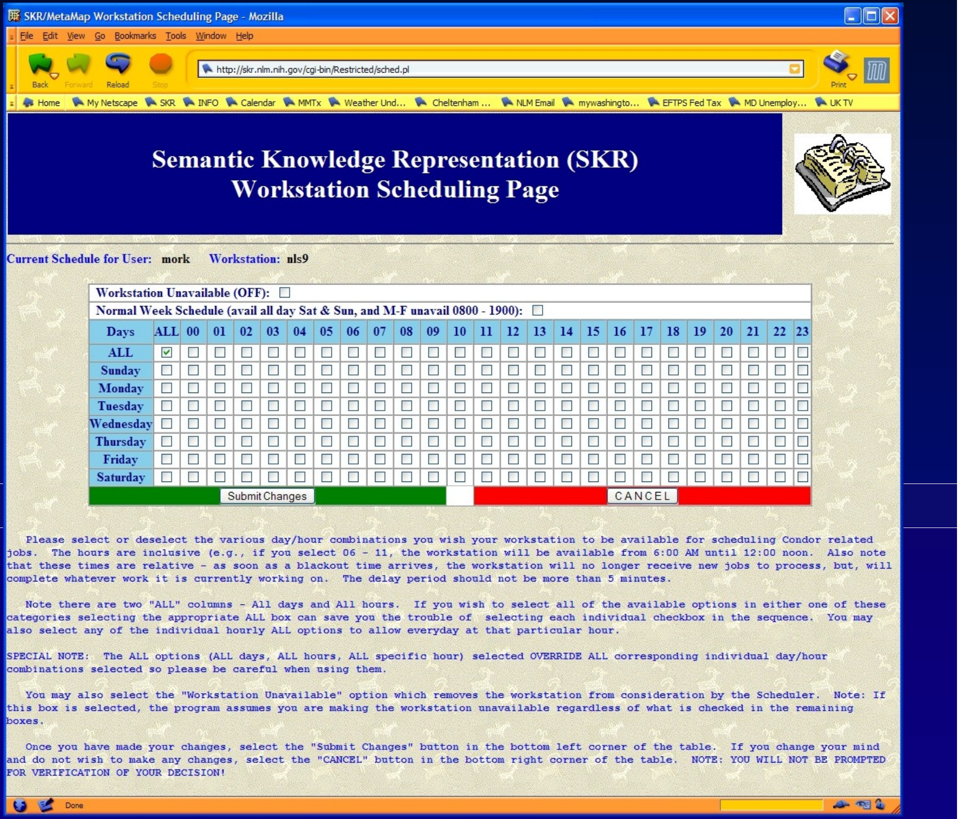
Last Updated: Thu Jul 15 09:00:11 2004

Refnum	Priority 0-n	User	M	Number Of Items	Processed Items	Number Errors	ETOC (hours)	APT (secs)	Command	Submitted
22053	0	admin	R	90000	29269 (32.5%)	0	4.91	0.29	foobarF	07/15 06:38:11
1121	0	mork	R	30000	7 (0.0%)	0	84.50	10.14	MTI -default_DCMS	07/15 08:58:56

Scheduler Started: Wed Jul 14 04:37:45 2004
Items Processed Since Start: 107540
# of Workstations Available: 17
Average Processing Speed: 5.22 Secs/Citation - 11.50/Min. - 690.06/Hr.

Workstation Scheduling

- Only allowed by designated responsible user/admin
- Tells us when we can and can’t use a workstation
- We allow the last running item to complete before removing ourselves from the workstation


SKR/MetaMap Workstation Scheduling Page - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
Print
Home My Netscape SKR INFO Calendar MMTx Weather Und... Cheltenham ... NLM Email mywashingto... EFTPS Fed Tax MD Unemploy... UK TV
Done

http://skr.nlm.nih.gov/cgi-bin/Restricted/sched.pl

Semantic Knowledge Representation (SKR) Workstation Scheduling Page

Current Schedule for User: mork **Workstation:** nls9

Workstation Unavailable (OFF): □

Normal Week Schedule (avail all day Sat & Sun, and M-F unavail 0800 - 1900): □

Days	ALL	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ALL	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Sunday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Monday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Tuesday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Wednesday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Thursday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Friday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Saturday	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Please select or deselect the various day/hour combinations you wish your workstation to be available for scheduling Condor related jobs. The hours are inclusive (e.g., if you select 06 - 11, the workstation will be available from 6:00 AM until 12:00 noon. Also note that these times are relative - as soon as a blackout time arrives, the workstation will no longer receive new jobs to process, but, will complete whatever work it is currently working on. The delay period should not be more than 5 minutes.

Note there are two "ALL" columns - All days and All hours. If you wish to select all of the available options in either one of these categories selecting the appropriate ALL box can save you the trouble of selecting each individual checkbox in the sequence. You may also select any of the individual hourly ALL options to allow everyday at that particular hour.

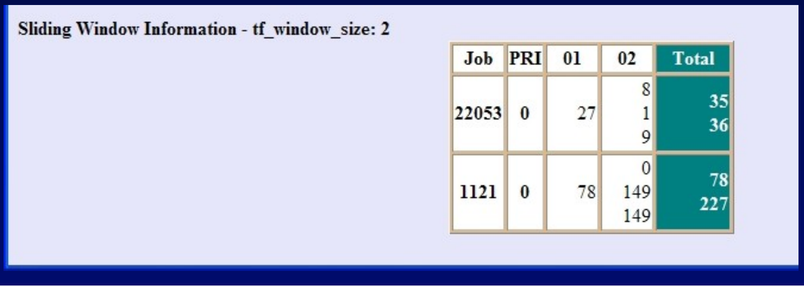
SPECIAL NOTE: The ALL options (ALL days, ALL hours, ALL specific hour) selected OVERRIDE ALL corresponding individual day/hour combinations selected so please be careful when using them.

You may also select the "Workstation Unavailable" option which removes the workstation from consideration by the Scheduler. Note: If this box is selected, the program assumes you are making the workstation unavailable regardless of what is checked in the remaining boxes.

Once you have made your changes, select the "Submit Changes" button in the bottom left corner of the table. If you change your mind and do not wish to make any changes, select the "CANCEL" button in the bottom right corner of the table. NOTE: YOU WILL NOT BE PROMPTED FOR VERIFICATION OF YOUR DECISION!

Resource Allocation

- The Scheduler uses resource utilization (actual + projected) within a sliding window to determine which job receives the next available resource.
- Job with lowest total score receives next resource.
- Takes into account both actual usage + estimated usage based on items running but not completed yet.
- Also takes into consideration Medium priority by dividing overall usage by 10 to provide 10:1 ratio of usage.



Sliding Window Information - tf_window_size: 2

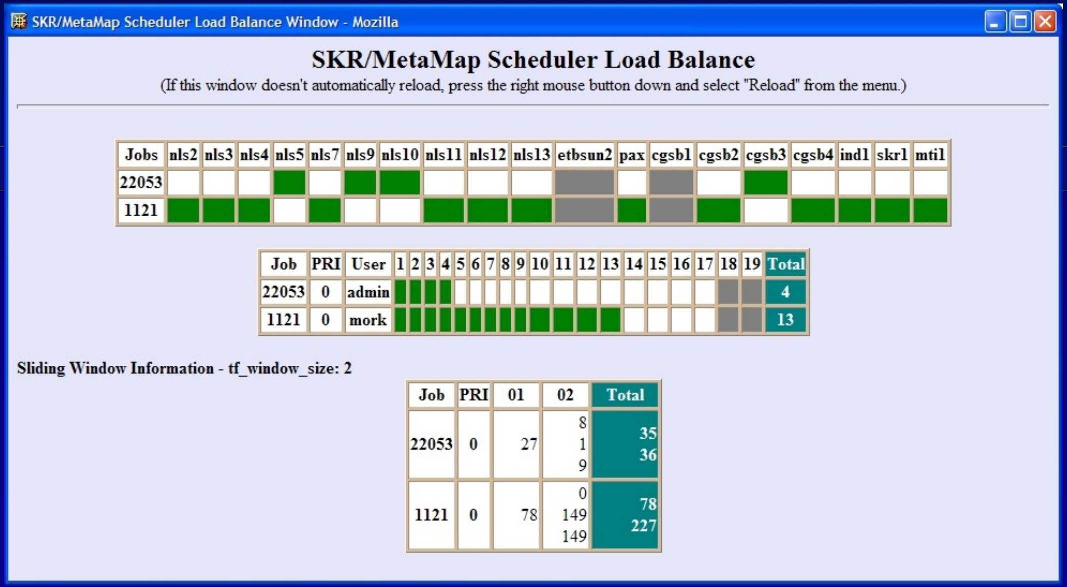
Job	PRI	01	02	Total
22053	0	27	8 1 9	35 36
1121	0	78	0 149 149	78 227

Resource Allocation (continued)

- Each window represents 10 seconds
- Currently based allocation on two windows
  - More windows penalizes running jobs versus new jobs
  - Less windows insufficient data to properly decide
- Still working out best number of windows to use
- We want to allow faster running jobs more access to the queue to balance their utilization over a job that takes more time to process an item.

Load Balance Window

- Access by Administrator only
- Provides real-time display of load balancing algorithm
- Green (active), White (not active), Grey (not available)



SKR/MetaMap Scheduler Load Balance

(If this window doesn't automatically reload, press the right mouse button down and select "Reload" from the menu.)

Jobs	nls2	nls3	nls4	nls5	nls7	nls9	nls10	nls11	nls12	nls13	etbsun2	pax	cgsb1	cgsb2	cgsb3	cgsb4	ind1	skr1	mtil
22053	⬜	⬜	⬜	🟩	⬜	🟩	🟩	⬜	⬜	⬜	▨	⬜	▨	⬜	🟩	⬜	⬜	⬜	⬜
1121	🟩	🟩	🟩	⬜	🟩	⬜	⬜	🟩	🟩	🟩	▨	🟩	▨	🟩	⬜	🟩	🟩	🟩	🟩

Job	PRI	User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Total
22053	0	admin	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	▨	▨	4
1121	0	mork	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬜	⬜	⬜	⬜	▨	▨	13

Sliding Window Information - tf_window_size: 2

Job	PRI	01	02	Total
22053	0	27	8 1 9	35 36
1121	0	78	0 149 149	78 227

Administrator

- Start/Stop/Pause Queue
- Reprioritize Jobs
- Suspend/Resume/Rerun Jobs
- Modify all Workstation Schedules
- View Load Balance Information
- Modify Load Balance Sliding Window
- Modify Individual Item Timeout

Usage Summary

- 58 Unique Users
- 4,218 Unique Jobs since April 25, 2000
- Used every night for overnight DCMS processing automatically resubmitting same job with new data.
- Jobs range from 1 item to well over a million items. The only limit is disk space.
- Has cut yearly processing time from weeks to days
- System has evolved into a general purpose tool capable of assisting researchers in processing large datasets.

Future Direction

- Move towards Linux for servers
  - New multi-cpu Linux server for Scheduler
  - New multi-cpu Linux server for Textool server
  - Looking to move tagger_servers to Linux as well
- Integrate Textool_monitor into Scheduler heartbeat
- Further refinement of resource allocation algorithm
- Look at ways to improve efficiency

Underlying Programs

- run_clt – Simply waits for the Scheduler to give it a job to run, runs it, and returns the results.
- tagger_monitor – Runs on skr1 and runs ps to see if a tagger_server process is running, sleeps 5 seconds, and repeats. When prompted by the Scheduler, it responds with a heartbeat.
- Textool_monitor – Runs on mti1 and doesn't communicate with the Scheduler. Monitors the Textool server via ps command and kills if > 95% CPU usage. The server is restarted automatically.
- force_error – Allows admin to force an error on a job which forces a checkpoint write.

Underlying Programs

- fix_error – Allows us to manually rerun any errors that occur in batch and splice the correct results into the results file in the proper location.
- clean_hist – Handles the cleaning of batch directories and Interactive files that are more than 15 days old and updates the list of available batches for rerun/resume.