AUTHOR: Danny Sallurday

DATE: 16 July 2019

OBJECTIVE:
Calculate and visualize SALT overhead stats for each instrument.

- Here, the overheads are the spans of time it takes between the system receiving a point command and acquiring the first science image of the corresponding target

IMPLEMENTATION:
A Python script which takes in two arguments (start date & end date) and generates a PDF with a barplot of overhead stats for the given range.

COMPONENTS:

- Function to calculate overhead stats and populate the database
- Python script to run the above function over a given range of dates
    - **Note:** This will be updated to a script automatically running (daily?)
- Function to fetch the relevant stats for one night
- Python script to run the above function over the given range of dates, accumulate and calculate overall statistics, and plot on a PDF

PROCEDURE:

Calculating Stats & Populating the Database

Script (run_overheadstats.py)

- Call on the command line with two arguments, start date and end date, formatted as: YYYYMMDD YYYYMMDD
- Iterates over the dates within this range, calling the overheadstats function on each

Function (overheadstats.py)

- Uses parameter (the string of an observation date) to obtain the date's NightInfo_Id (**nid**)
- Uses **nid** for a query to create a list of all accepted BlockVisit_Id (**bvid)** for that night
- Uses **nid** for a query to create a list of all Events (via the SoLogEvent table) for that night
- Gets a list of all images using a query that looks for FileNames containing the date
- Gets a list of all RSS images using a similar query (to handle MOS case later)
- Gets a list of all point commands using a query with **nid**
- Iterates through accepted **bvid** list:
    - Stores the time of the point command that corresponds with the **bvid**
        - This is the start time of the block

- Stores the time of the next ‘TrackEnd’ event after the start time
    - This is the end time of the block
- Stores the time between the start time and the next ‘TrackStart’ event
    - This is the **Slew Time** of the block
- Stores the time between the ‘TrackStart’ Event and the next ‘OnTarget’ event
    - This is the **Tracker Slew Time** of the block
- Uses **bvid** to obtain the primary instrument (SCAM/RSS/HRS) of the block
- Uses **bvid** to obtain the corresponding Block_Id
- Uses Block_Id for a query which determines if the block uses MOS
- If it is a MOS block:
    - Stores time between ‘OnTarget’ and the first MOS science image
        - This is the **MOS Acquisition Time**
        - This uses the list of RSS images to find the first image that contains the characteristics which signify a MOS image
- If it is not a MOS block:
    - Stores time between ‘OnTarget’ and the first SALTICAM image
        - This is the **Target Acquisition Time**
    - Stores time between the SCAM image and the first science image
        - This is the **Instrument Acquisition Time**
        - This uses the previously determined primary instrument of the block and matches the **bvid**
- Updates the row in the BlockVisit table with the corresponding **bvid**
    - New columns added: SlewTime, TrackerSlewTime, TargetAcquisitionTime, InstrumentAcquisitionTime, MOSAcquisitionTime

Accumulating Stats and Plotting Results

Script (daterangestats.py)

- Called on the command line with two arguments, start date and end date, formatted as: YYYYMMDD YYYYMMDD
- Calls the getnightstats function on each date within the range
- The function returns:
    - **nightstats,** a 2D array of all relevant overheads
    - individual counts of RSS / HRS / MOS blocks
- Each index of **nightstats** contains an array of the overhead values for all blocks that night. Each array holds values for one stage of one instrument’s overheads. (i.e nightstats[0] is the list of every RSS block’s slew time)
- Updates arrays within the script, also separated into stage and instrument, with each night’s values. Also updates the numbers of blocks per instrument each iteration.
- Creates a dictionary for each instrument, with the keys being a stage (i.e Slew Time), and the corresponding value being the median time of the list of blocks’ values.
- Concatenates the 3 dictionaries and converts them into a pandas DataFrame
- Generates a stacked bar plot, with one bar per instrument. Each bar is comprised of distinct colors which represent each of the overhead stages.

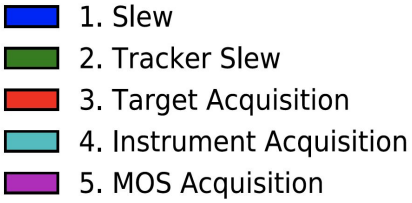
- Labels for all of the values are added on the plot
- Saves the plot on a PDF stored in the working directory

Function (nightstats.py)

- Uses parameter (the string of an observation date) to obtain the date's NightInfo_Id (**nid**)
- Uses **nid** for a query to create a list of all accepted BlockVisit_Id (**bvid)** from that night
- Iterates through accepted **bvid** list
    - Determines primary instrument associated with the **bvid**
        - This is to know which stats to pull. For RSS/HRS, it selects 4 columns, 1 for each stage. For MOS, only 3 columns will be populated.
    - Obtains values for each overhead stage of the **bvid**
    - Appends each value to the corresponding array dedicated to that specific instrument and stage (i.e RSS Slew)
    - Increments counts of each type of block (RSS/HRS/MOS)
- Creates a 2D array of the stats
- Returns the 2D array and individual counts of RSS / HRS / MOS blocks

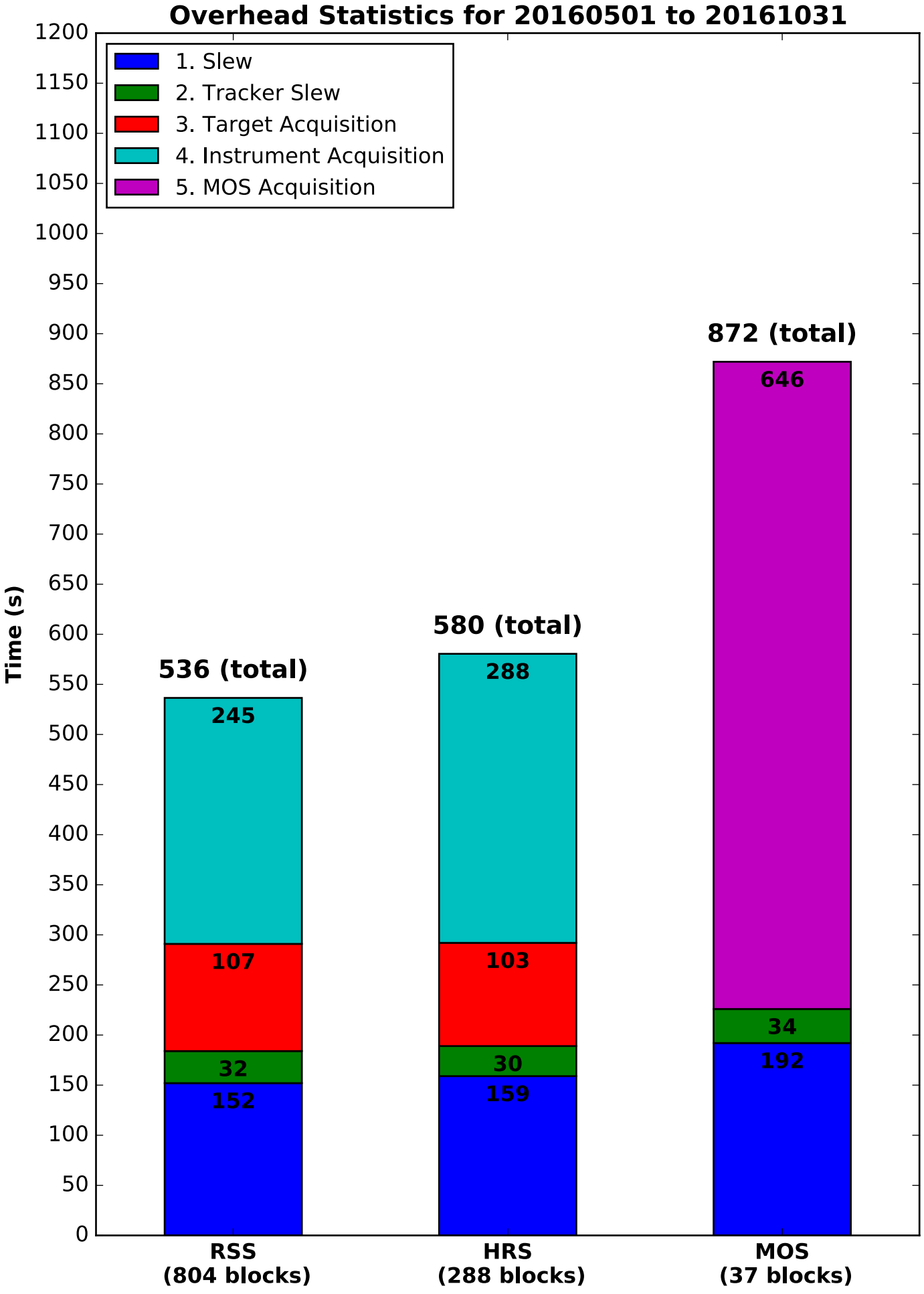
INTERPRETING THE RESULTS:

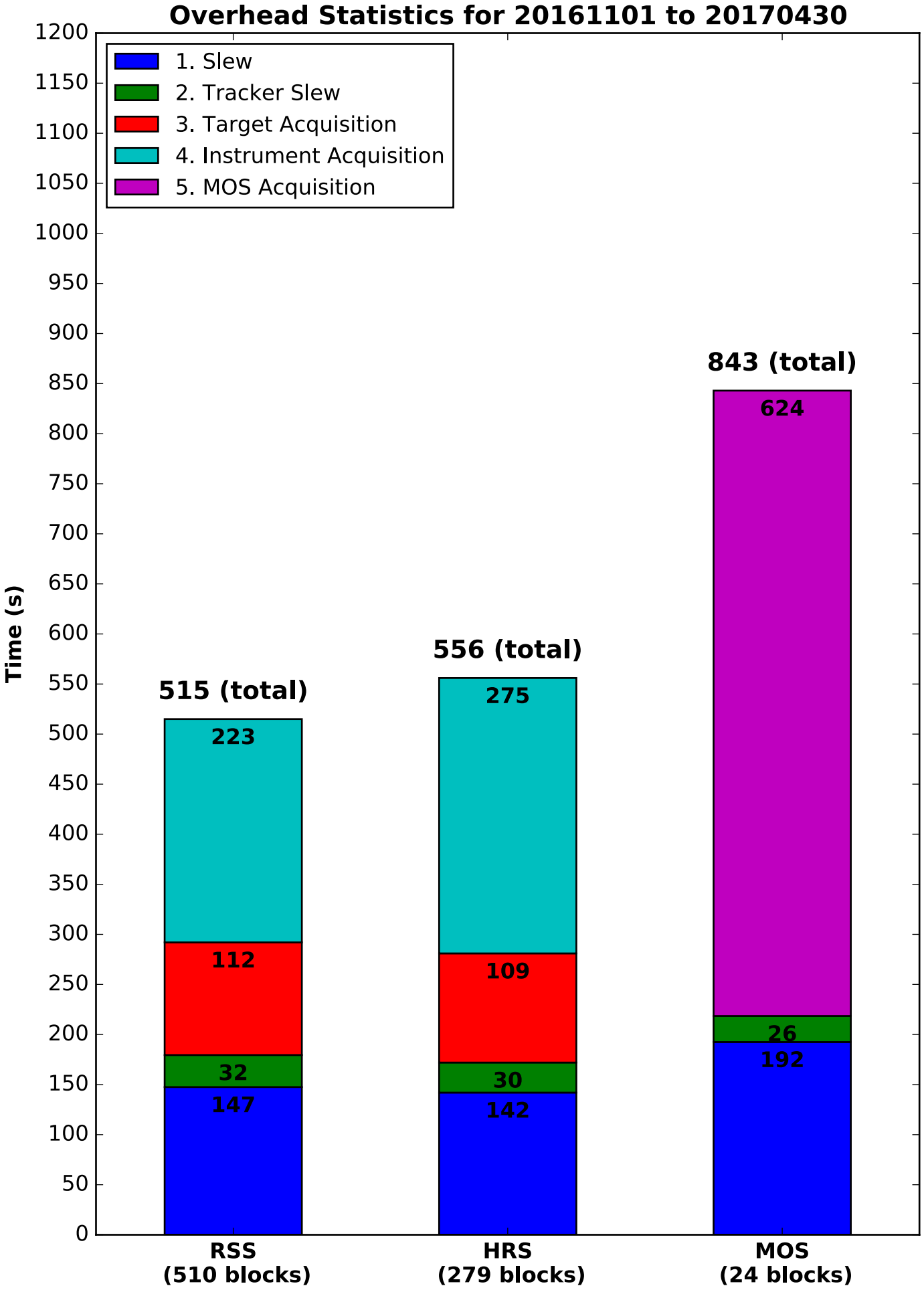
The plot consists of 3 separate bars labelled for the instrument they represent: RSS, HRS, MOS
The y-axis is time in seconds, signifying how long overheads are taking. The durations displayed are the median values of the set of times from all blocks in the date range. The bars are split up into 4 (RSS / HRS) or 3 (MOS). Within the bars, each segment represents a different stage of overhead and is colored distinctly. The legend contains the labels for each color.

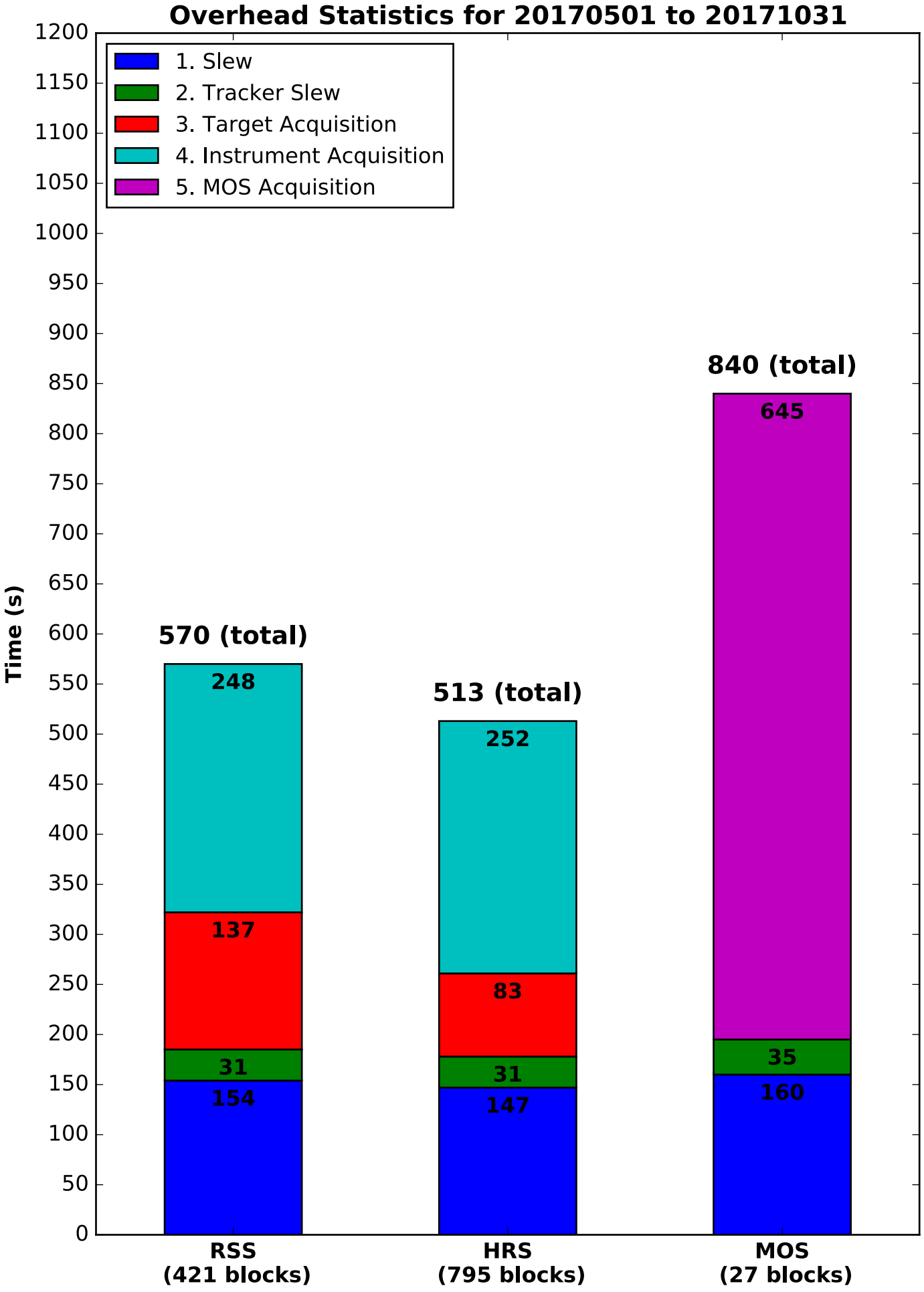


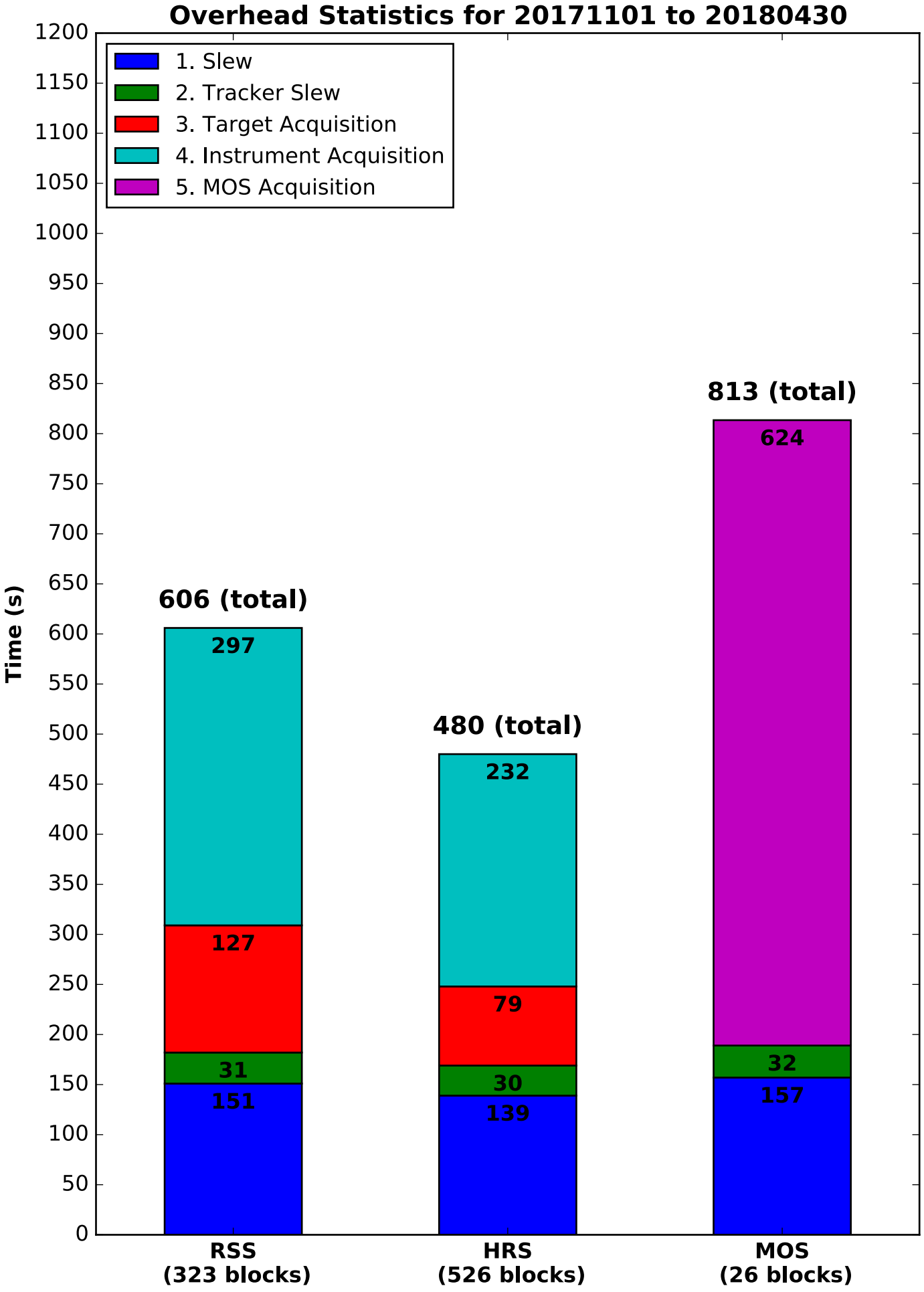
Slew: Point Command to TrackStart
Tracker Slew: TrackStart to OnTarget
Target Acquisition: OnTarget to SCAM image
Instrument Acquisition: SCAM image to science image
MOS Acquisition: OnTarget to MOS image

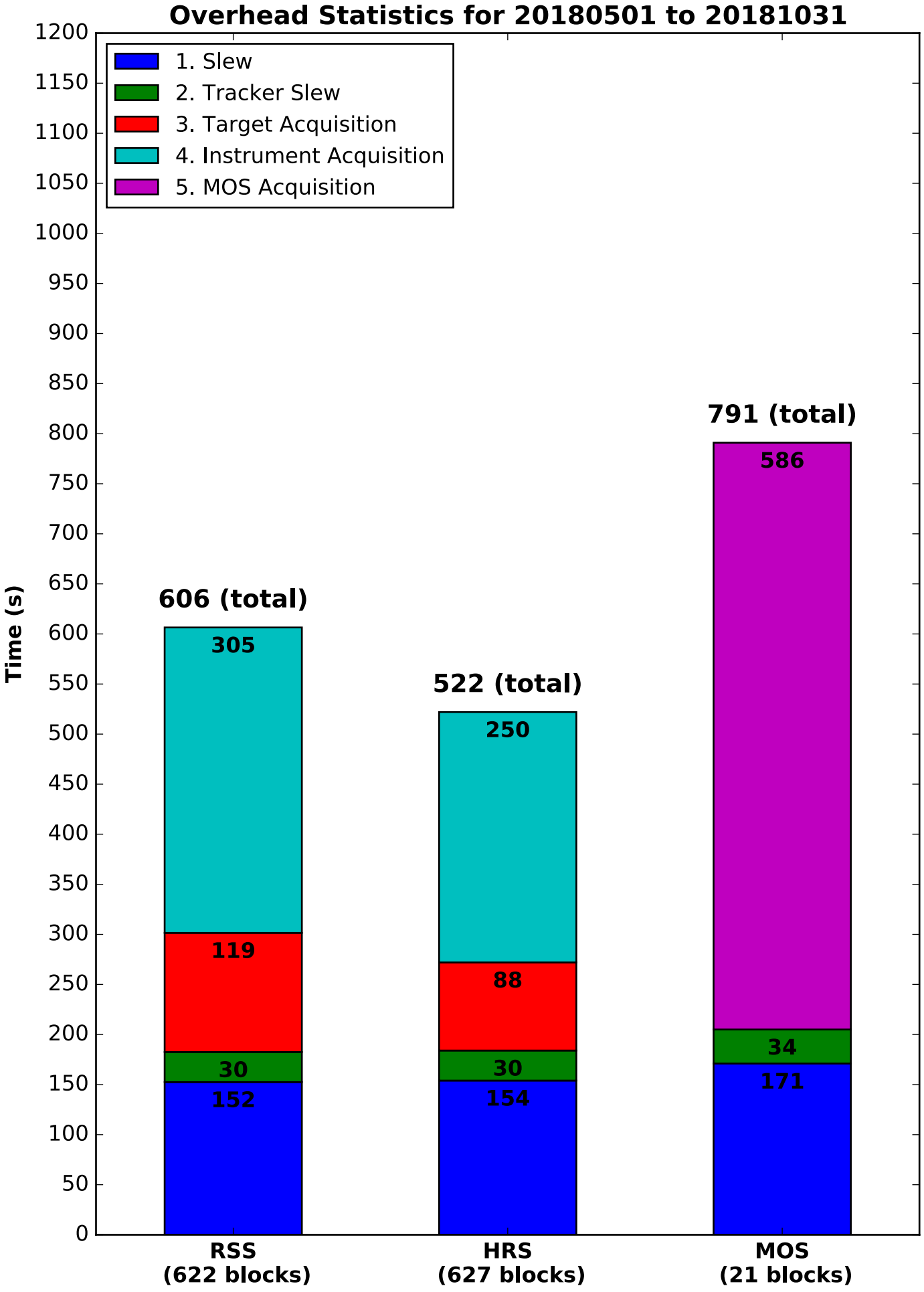
In addition, the total time is displayed above the bar. This is the total overhead for the instrument. Under the instrument labels on the horizontal axis, the amounts of blocks taken into account to acquire the median values are also displayed.

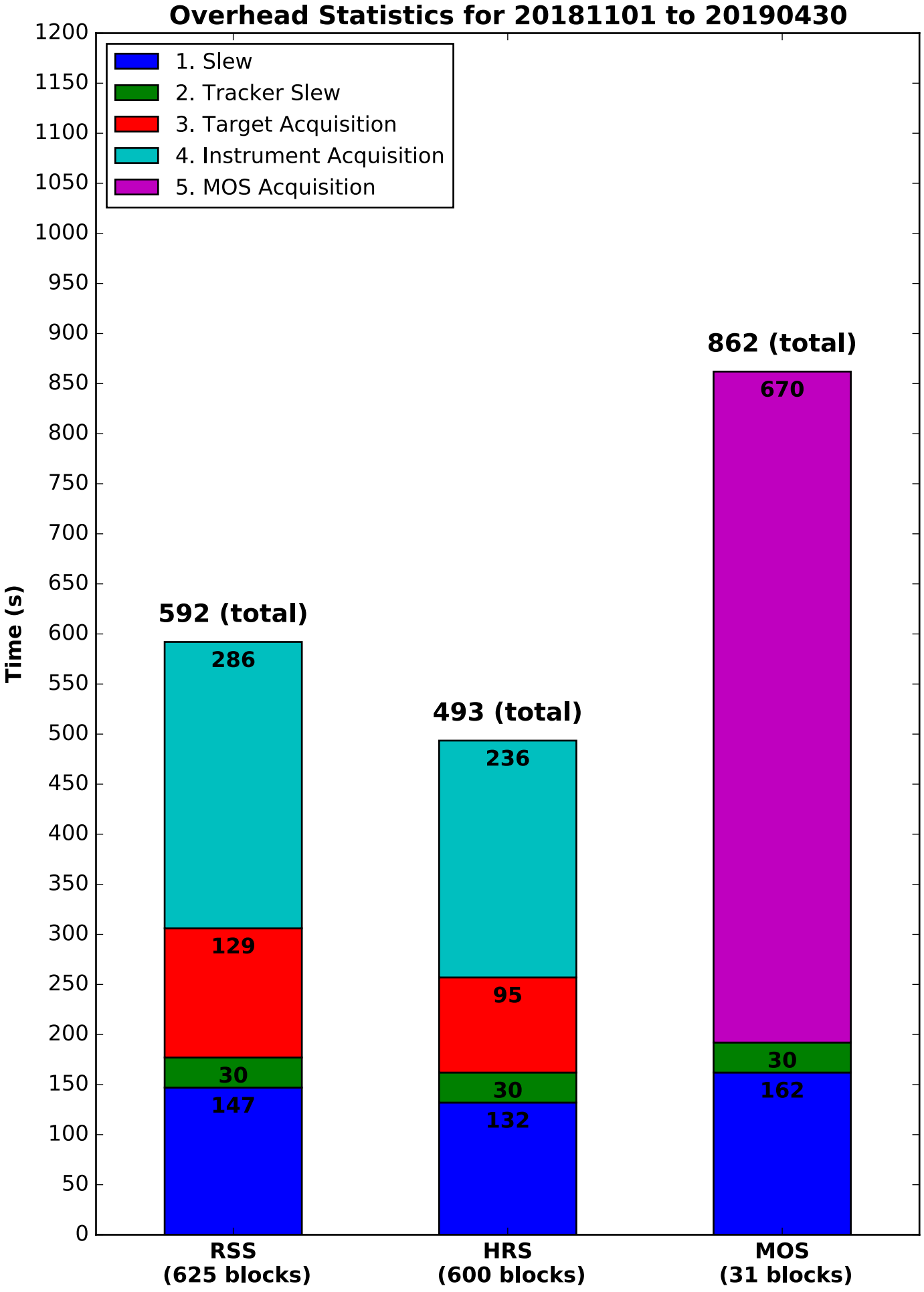

Overhead Statistics for 20160501 to 20161031
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
536 (total)
245
107
32
152
580 (total)
288
103
30
159
872 (total)
646
34
192
RSS
(804 blocks)
HRS
(288 blocks)
MOS
(37 blocks)


Overhead Statistics for 20161101 to 20170430
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
515 (total)
223
112
32
147
556 (total)
275
109
30
142
843 (total)
624
26
192
RSS
(510 blocks)
HRS
(279 blocks)
MOS
(24 blocks)


Overhead Statistics for 20170501 to 20171031
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
570 (total)
248
137
31
154
513 (total)
252
83
31
147
840 (total)
645
35
160
RSS
(421 blocks)
HRS
(795 blocks)
MOS
(27 blocks)


Overhead Statistics for 20171101 to 20180430
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
606 (total)
297
127
31
151
480 (total)
232
79
30
139
813 (total)
624
32
157
RSS
(323 blocks)
HRS
(526 blocks)
MOS
(26 blocks)


Overhead Statistics for 20180501 to 20181031
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
606 (total)
305
119
30
152
522 (total)
250
88
30
154
791 (total)
586
34
171
RSS
(622 blocks)
HRS
(627 blocks)
MOS
(21 blocks)


Overhead Statistics for 20181101 to 20190430
1. Slew
2. Tracker Slew
3. Target Acquisition
4. Instrument Acquisition
5. MOS Acquisition
Time (s)
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
1050
1100
1150
1200
592 (total)
286
129
30
147
493 (total)
236
95
30
132
862 (total)
670
30
162
RSS
(625 blocks)
HRS
(600 blocks)
MOS
(31 blocks)