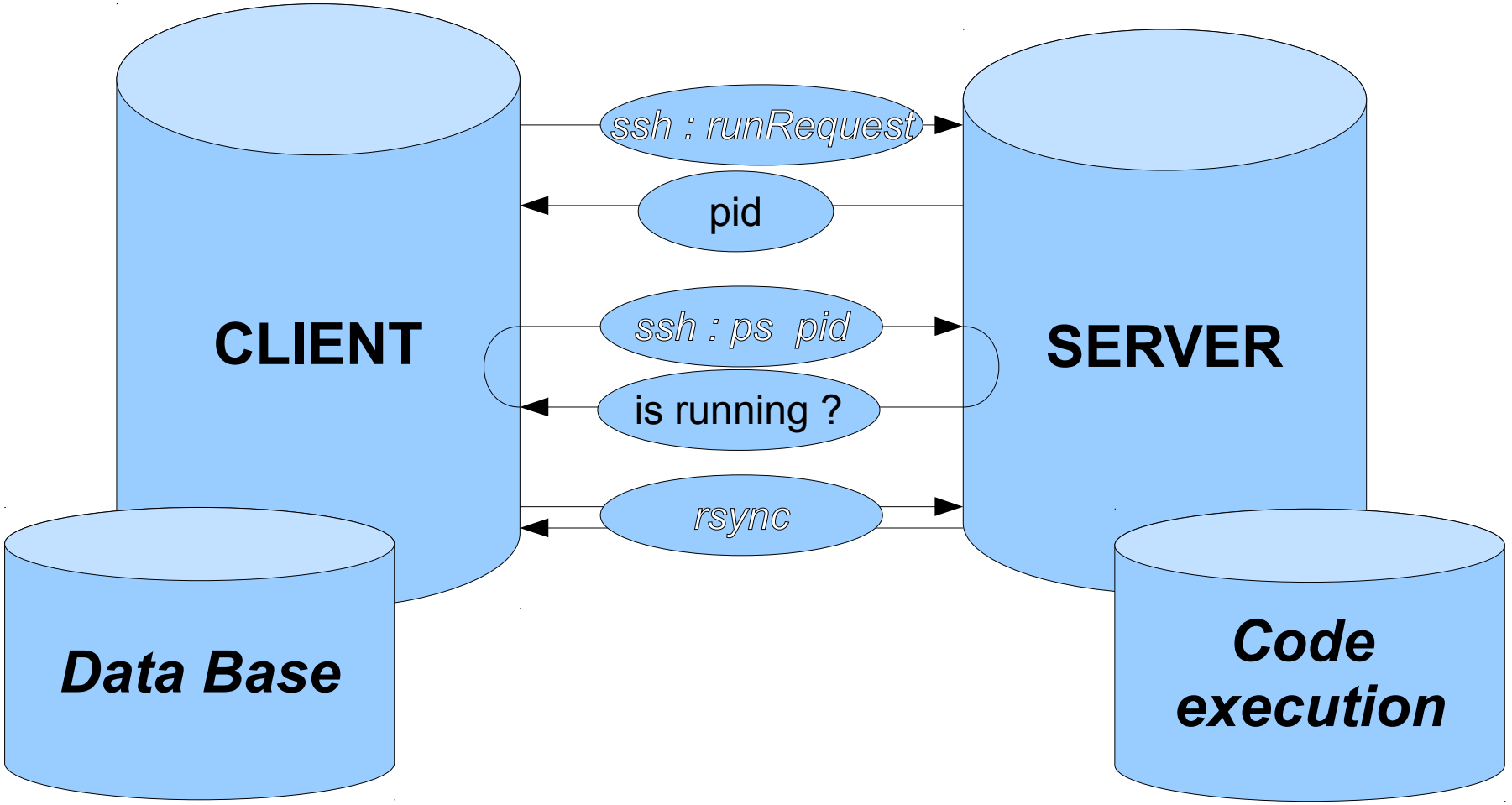
Tao SW2 (Several Reference Angles) MAJ

How it works

Remote Procedure Call



remoteUpdate.sh (1)

- Input arguments : *target, sw_source, [phase]*
- Set 'request-dependent' environment :

  . swenv.sh targetMAJ, sw_source
- Get corresponding 'TaoStop' :

  TaoStop = php getTaoStop.php
- Get corresponding 'SwStop' :

  SwStop = php getSwStop sw_source

remoteUpdate.sh (2)

- Run request @ remote:

  ssh : runRequest tgtVi swVi TaoStop SwStop

- Loop : while *ssh : ps pid != 0* continue
- Get results from remote (*rsync*) and update database :

  UpdateLocalDataBase target

remoteUpdate.sh : input args

- Target (Mission):
  - planets, moons : *[mercury,...,saturn]*

    tgtVi = name_orb_all
  - missions : *[rosetta, juno]*

    tgtVi depends on mission itself (Rosetta) or mission phase (Juno)
- SW Source :

  OMNI (by default), ACE_RealTime, STA, STB
- Mission phase (optional) : *cruise, planet. If defined - tgtVi is constructed as tgtVi = mission_phase_all*

Client side : swenv.sh

Set environmental vars, should be edited manually

- targetMAJ = Tao VI location in DD Base,
  normally targetMAJ = upper(target)
- Remote settings : *USER@REMOTEHOST*, *SW2ROOT*, *REMOTEPATH* (dir with result nc files)
- Local settings : *TAOROOT* (VI location in DD Base), *sw_vi* (depends on SW SOURCE)

Client side : getTaoStop.php

- Returns GlobalStop (ISO) of corresponding Tao VI by parsing vi_info.xml
- In case of ACE_RealTime returns “TAO_OMNI” GlobalStop – NNN days

Client side : getSwStop.php

- Returns corresponding GlobalStop of SW src VI.
  - OMNI : read SwStop file (SW 'real' stop) in *OMNI/HOUR*

    SwStop is updated by sw_stop.py while MAJ OMNI/HOUR (in depotDECODER)
  - ACE : read ACE/RealTime/MAG[SWEPAM]/vi_info.xml

    Return min of 2 times

Server side : runRequest

- Remote procedure call
  ssh runRequest tgtVi swVi TaoStop SwStop
- *RunRequest :*
  - *delete previous results/requests*
  - *create request.xml in **REQ** dir with*

    python makeRequest.py
  - *DDLogin USER PASSWD*
  - *python run.py*

runRequest : settings (env.sh)

- SW2ROOT
- DDUSER
- DDPWD
- DDPATH (**DD.res**)
- REQ=$SW2ROOT/Requests
- DD_DIR=/usr/local/DDClient (**DDLogin**)
- TINYDIR=/usr/local/tinyxml
- LD_LIBRARY_PATH=$DD_DIR/lib:$TINYDIR/li
- PATH=/bin:/usr/bin:/usr/local/bin:./:$SW2ROOT/Bin:$DD_DIR/bin
- SW2DATA=$SW2ROOT/Data (**working dir**)
- SW2NC=$SW2ROOT/Nc (**final results**)

runRequest : makeRequest.py

- Request name : *target_YYYY*.xml
- <START>*startTime*</START> : **YYYY-01-01** (or **TAO_OMNI_Stop – NNN days** if ACE_RealTime**)**
- <STOP>*stopTime*</STOP> : **SwStop**
- <PLASMA_VI>*plasmaVI*</PLASMA_VI> : **SW SRC VI**
- <SOURCE_VI>earth_orb_all</SOURCE_VI>
- <SOURCE_R_PARAM>R</SOURCE_R_PARAM>
- <SOURCE_LON_PARAM>LON_HCI</SOURCE_LON_PARAM>
- <TARGET_VI>*tgtVI*</TARGET_VI> : **tgtVI**
- <TARGET_R_PARAM>*tgtRParam*</TARGET_R_PARAM> : **R**
- <TARGET_LON_PARAM>*tgtLonParam*</TARGET_LON_PARAM> : **LON_HCI**

runRequest : run.py

- Important !!! Uses New DD Server !!!!
- *get_R_LON_HCI*
- *get_OMNI_1H, get_ACE_RT*
- *sw.exe*

Tao Code : sw.out & sw.exe

- Executable ***sw.out*** : Tao original code
  - Executable only !!!
  - Errors while Fortran src compilation.
- Executable ***sw.exe*** :
  - Minor modifs in Tao Fortran code (marked in the src files) – compilation errors corrections.
  - Modifs in Tao Fortran code (marked in the src files) to 'stabilise' MHD runs.
  - *safety* = *0.4* (original value)

Tao Code : Input / Output

- **Input File Format** (any time sampling):

  time[sec j2000], density[/cc], T[K], Vx[km/s], Vy[km/s], Vz[km/s], Bx[nT] , By[nT], Bz[nT],

  src (input) radius [AU], src (input) longitude [deg], tgt (output) radius [AU], tgt (output) longitude [deg],

  #, data_flag(datalack=1, data=0) – not used in the current version

- **Output File Format** :

  time yyyy mm dd hh mm ss, density[/cc],T[K], Vx[km/s],Vy[km/s],By[nT], dynamic pressure [nPa], longitude difference between input and output [deg] #, data lack flag (1=lack, 0=data)

Tao Code : control parameters (*namelist*)

- idp_in=1/0 – produce input files yes/no
- idp_prop=1/0 – run simulation yes/no
- idp_out=1/0 – produce output yes/no
- angref(1)=0.,60.,120.,180.,240.,300.
- ldprop=1/-1 – outward/inward propagation
- fnin=inputs.txt - input file
- fnout=outputs.txt - output file
- fdirtmp=...
- instop=0
- dtr=150 - time step (sec)
- touts=24 - output sampling => dtr*touts (sec)
- xmin1 => inner boundary (AU) 0.3/0.4/0.8
- xmax1=> outer boundary (AU) 1.3/1.4/.....

Tao Code : schema

Preparing Tao Input

Client side : Get Results : updateLocalDataBase