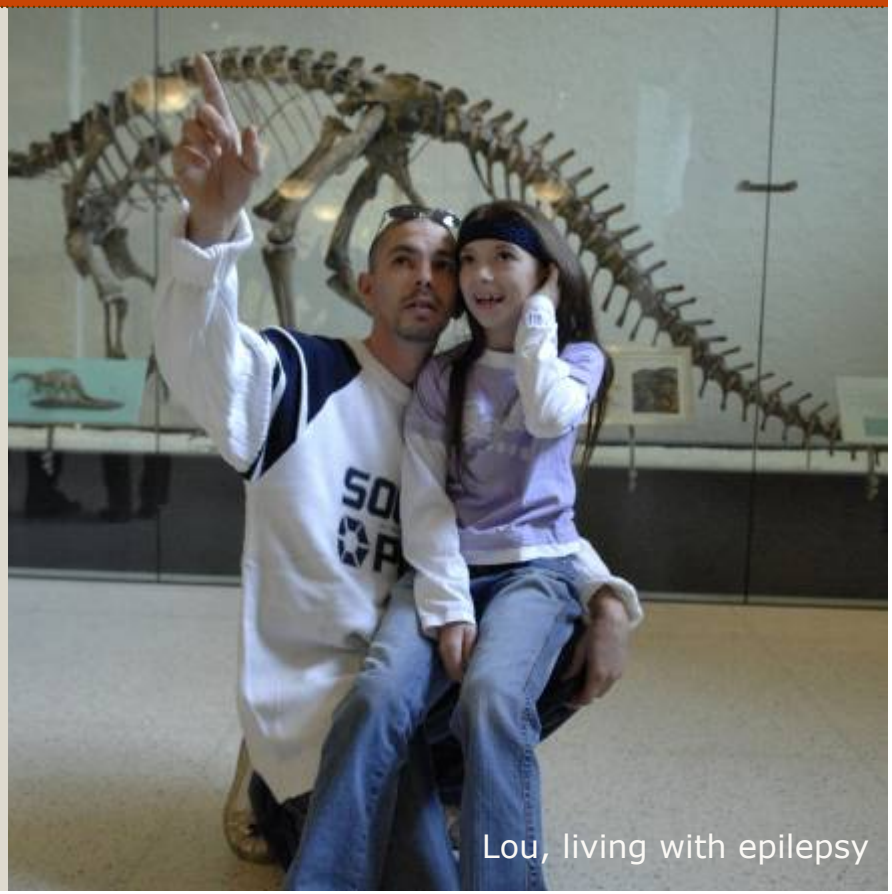
Implementation of a compute cluster for R/ BUGS simulations

Maud Destrée
Alain Dalis
Will Talbot
Charles Castelain
UCB – IT Department


Lou, living with epilepsy

26 mai 10

Agenda

- Background
- Goal & Requirements
- Compute cluster architecture
- Changes in R code
- Conclusions and next steps

Background

- Statisticians run R/BUGS simulations on their laptops -> performance problems
- Not enough CPU available
- In case of BUGS “trap”, the simulations had to be restarted manually
- Upgrading the R/BUGS versions must be done on several machines
- Difficult to check simulations status from home

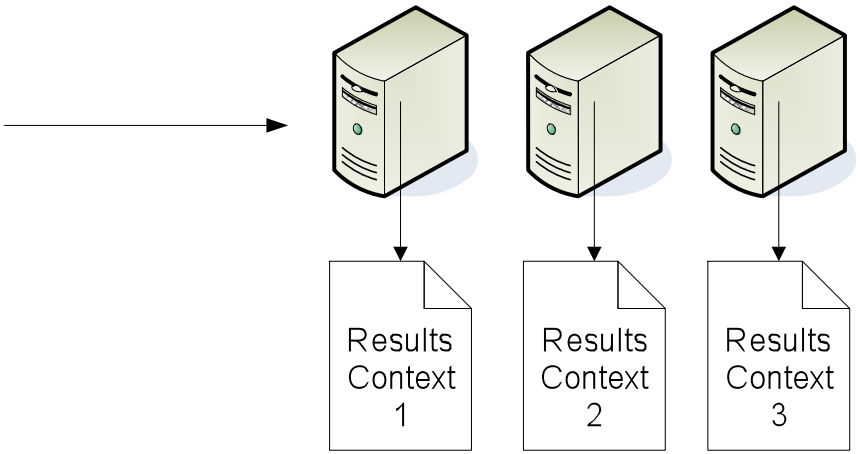
Goals & Requirements

- Remove heavy simulations from the laptops -> remote environment
- Increase the number of CPU available for BUGS simulations
- Window based environment to run WinBUGS 1.4
- Run simulations in batch mode
- Work on UCB environment from home

R code - Example

- Initialisation
  - data
  - initial values, contexts
- BUGS Simulations
  - "For" loop on contexts
    - "For" loop on Markov chains

      bugs(...)
- Final analysis



Initial investigation - Tests

- Metrum: "Open source software for parallel computing of multiple MCMC chains with WinBUGS" W.R.Gillepsie, M.R. Gastonguay, W.Knebel, G.Georgalis
- MPI (Message Passing Interface): library of functions for C or Fortran to execute program on distant computers
- RMPI: implementation of MPI for R
- bugsParallel: R module to allow starting multiple BUGS simulations on several processors
- Easy to implement
- Not robust enough in case of WinBUGS crash -> kill the process manually

Compute cluster

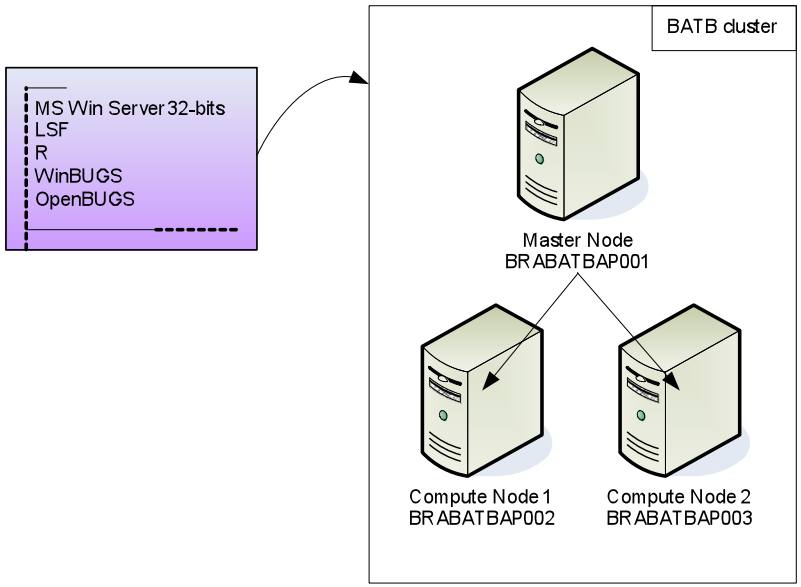
- Compute cluster: several (similar) computers grouped together in a (private) network in order to expedite calculations
- Job scheduler: manage queues of incoming requests and their priorities

Example	Simulation 1	Run
	Simulation 2	Run
	Simulation 3	Run
	Simulation 4	Run
	Simulation 5	Pending

- Master node: computer that receives the request and dispatch them to available processors on other computers, the compute nodes.

Cluster architecture (1)

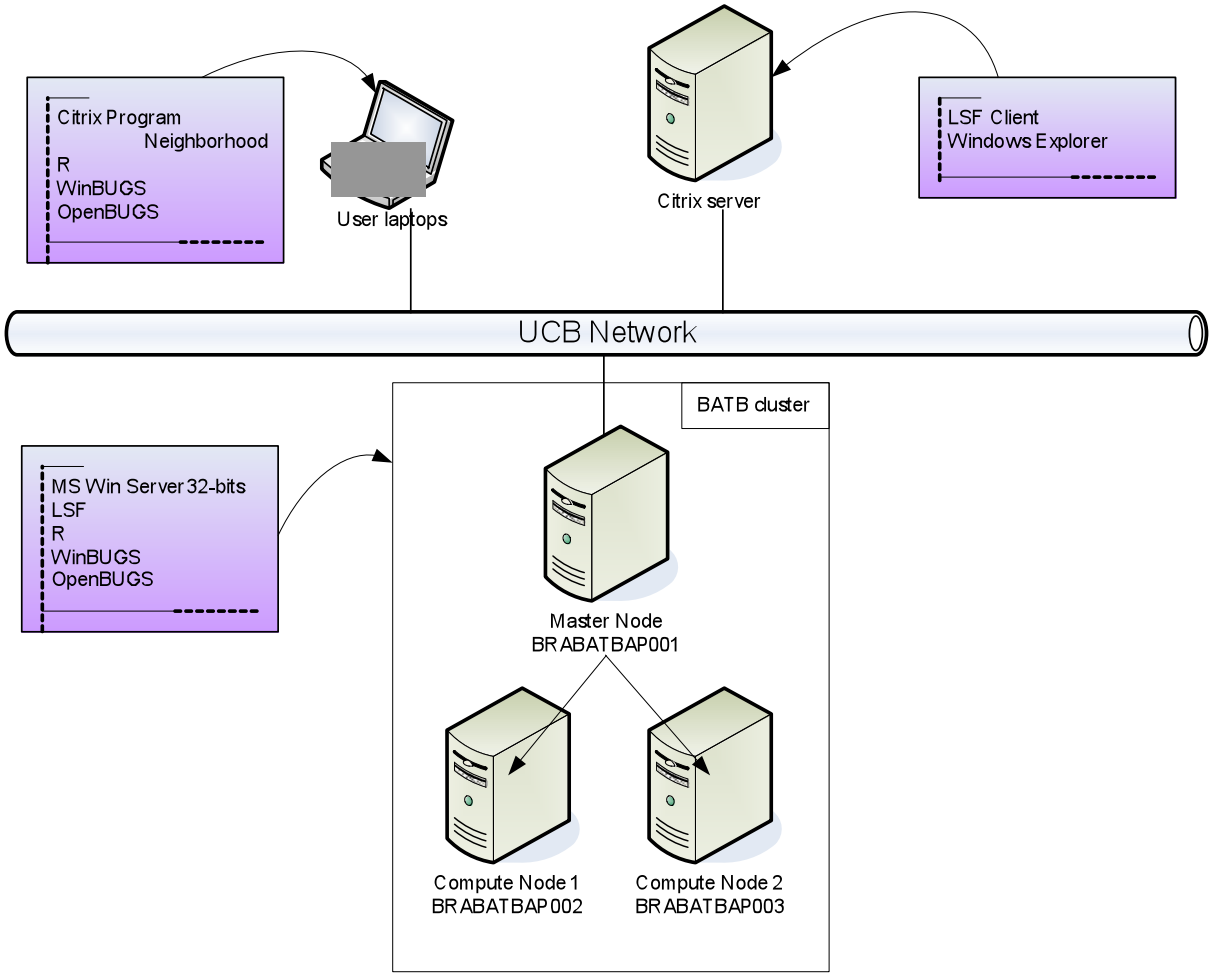
- 3 servers bi-quad cores, 4GB RAM
- Job scheduler: LSF (Load Sharing Facilities - Platform Computing)
- 1 master node and 2 compute nodes
- Master node: 4 cores reserved for managing incoming requests
- 20 cores available for running R/BUGS simulations
- Shared drive on master node (250 GB)



Cluster architecture (2) - Client part

- Access to cluster:
  - LSF client: submit the simulations to the cluster, view simulation jobs running/pending, check servers status
  - Windows Explorer: access to shared drive on cluster master node, launch simulation via DOS scripts
- Client components are installed on Citrix server:
  - Easy to upgrade LSF client
  - No additional installation required in case of a new user

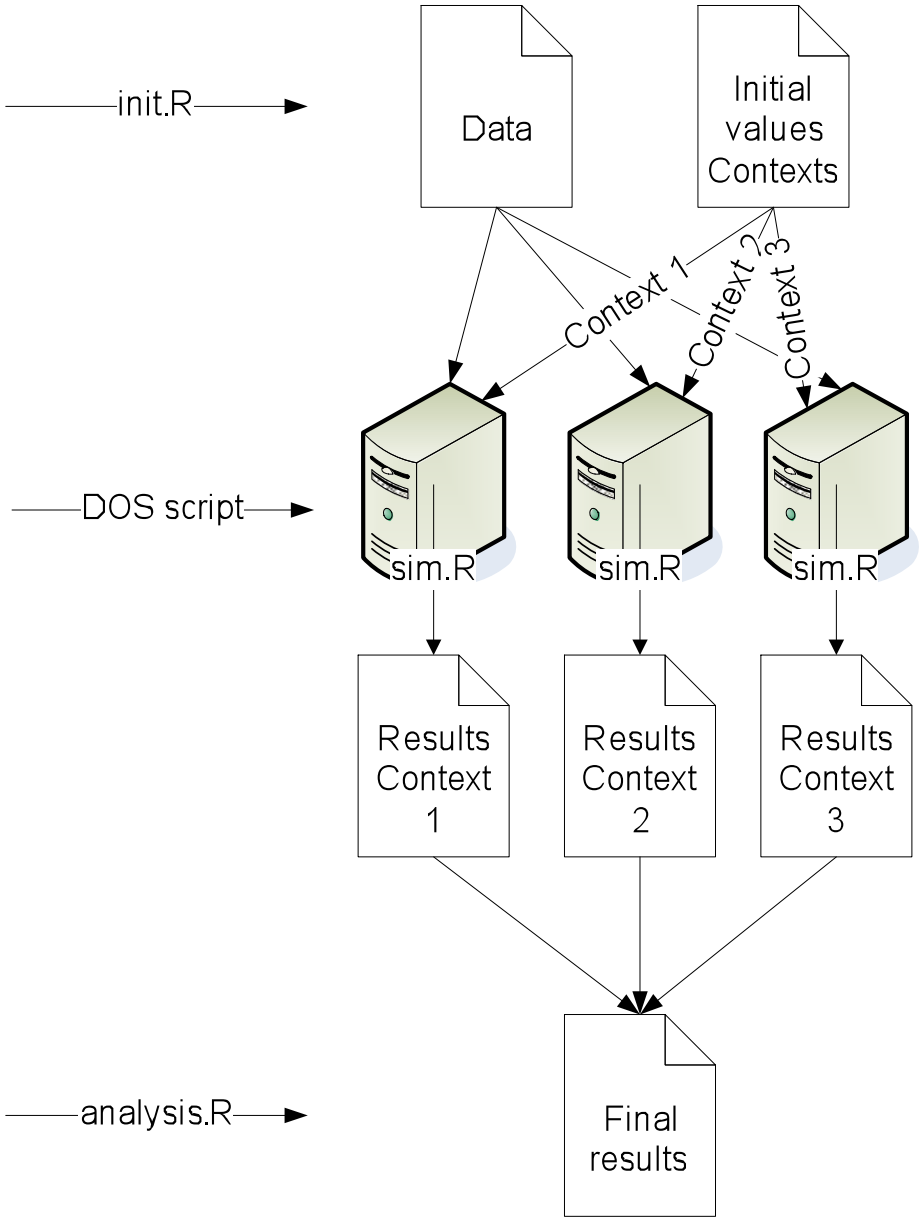
Cluster architecture (3)



Changes in R code - Example

- Initialisation
  - data
  - initial values, contexts
- BUGS Simulations
  - "For" loop on contexts
    - "For" loop on Markov chains

    bugs(...)
- Final analysis



DOS script


```
SET SUBDIR=Project                                          Project name
SET JOBDIR=\\BRABATBAP001\\USERS\\%USERNAME%\\%SUBDIR%      Project folder
SET LOGDIR=%JOBDIR%\\Logs

SET NCTX=27                                                 Number of contexts

for /L %%i in (1,1,%NCTX%) do (
  MD %JOBDIR%\\%%i                                          Create 1 folder / context
  COPY %JOBDIR%\\Data.txt %JOBDIR%\\%%i\\.                  Copy files in context folder
  COPY %JOBDIR%\\Init.txt %JOBDIR%\\%%i\\.
)

bsub -J %SUBDIR%_[1-%NCTX%] -o %LOGDIR%\\%SUBDIR%_%%I.txt -Q 128 %JOBDIR%\\Sim.bat
```

-J Job name

-o Output file name

-Q Specify that the job must be requeued for the given error code

Submit commands in Sim.bat as NCTX separate runs named Project_1, Project_2, ... Log files are located in Logs folder as Project_1.txt, Project_2.txt, ...Runs are requeued if the application exit with code 128.

```
pause
```

sim.R program

- Define working directory in R code: JOBDIR is defined in DOS script, LSB_JOBINDEX is defined by LSF

```
setwd(paste(Sys.getenv("JOBDIR"),"\\",Sys.getenv
("LSB_JOBINDEX"),sep=""))
```

- Make sure the R program can be restarted in case OpenBUGS crashes (no recovery possible with WinBUGS)

- Optimize the use of memory:
  - the BUGS model variables to be saved must be defined thoroughly
  - Remove large R objects when not used

Summary

- Organize the simulations to take advantage of the compute resources : split the problem into several bunch of simulations that can run independently on the different servers

- DOS script:
  - create working directories for simultaneous runs,
  - define environment variables value used in R program,
  - start the simulations

- Adapt R program to run on the cluster:
  - working environment variables,
  - restart from intermediate results file

Conclusions & Next Steps

- Compared to previous environment, simulation times are divided by 5 at least
- OpenBUGS crashes are automatically restarted
- Software is installed on remote computers, no additional installation needed for a new user

- Explore solutions to start parallel BUGS simulations directly from R code: package RLSF is available under Linux
- OpenBUGS is very slow for complicated model: investigate other samplers and MCMC packages