

Conférence Aramis 2022

Retour d'expérience du Centre Blaise Pascal
sur ses services de visualisation

... ou comment s'adapter aux nouvelles
contraintes d'exploitation...

Emmanuel Quémener

La visualisation : 1 des 4 piliers de l'informatique scientifique



Référence napoléonienne (détachée de tout contexte politique)



*« Un bon croquis vaut
toujours mieux qu'un
long discours ! »*

Napoléon Bonaparte

By Henri Félix Emmanuel Philippoteaux - Transferred from de.wikipedia to Commons by Stefan Bernd.Alt source: [1], Public Domain, <https://commons.wikimedia.org/w/index.php?curid=5556134>

Un croquis, ça aide en quoi ?

Exemple numérique : Runge Kutta

The Runge–Kutta method [edit]

The most widely known member of the Runge–Kutta family is generally referred to as "RK4", the "classic Runge–Kutta method" or simply as "the Runge–Kutta method".

Let an [initial value problem](#) be specified as follows:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0.$$

Here y is an unknown function (scalar or vector) of time t , which we would like to approximate; we are told that $\frac{dy}{dt}$, the rate at which y changes, is a function of t and of y itself. At the initial time t_0 the corresponding y value is y_0 . The function f and the [initial conditions](#) t_0, y_0 are given.

Now pick a step-size $h > 0$ and define

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4),$$

$$t_{n+1} = t_n + h$$

for $n = 0, 1, 2, 3, \dots$, using^[3]

$$k_1 = f(t_n, y_n),$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right),$$

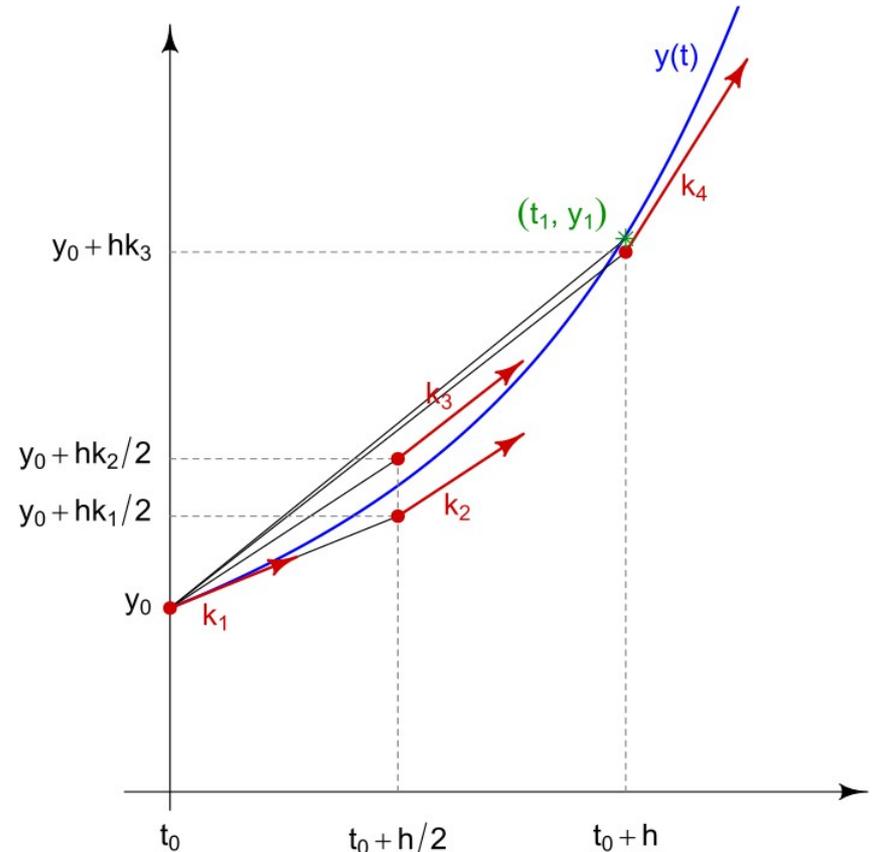
$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right),$$

$$k_4 = f(t_n + h, y_n + hk_3).$$

(Note: the above equations have different but equivalent definitions in different texts).^[4]

Here y_{n+1} is the RK4 approximation of $y(t_{n+1})$, and the next value (y_{n+1}) is determined by the present value (y_n) plus the [weighted average](#) of four increments, where each increment is the product of the size of the interval, h , and an estimated slope specified by function f on the right-hand side of the differential equation.

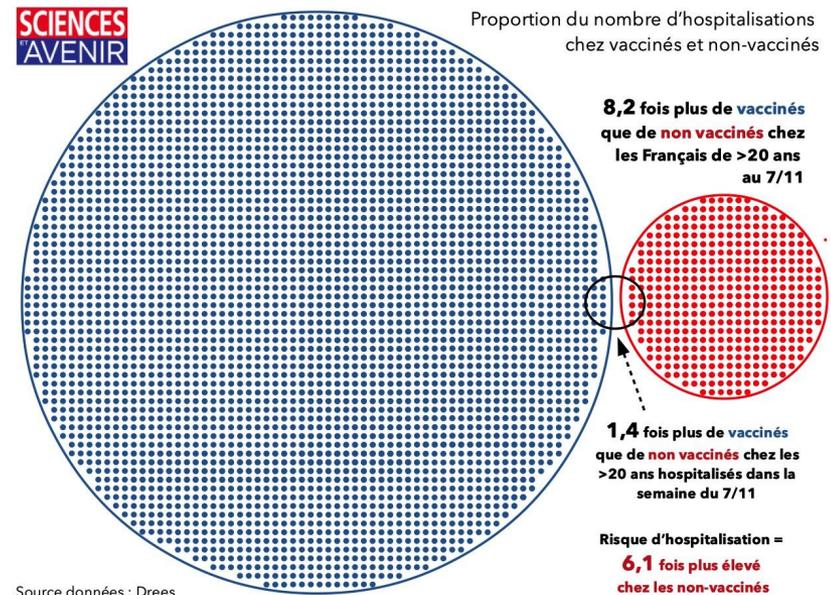
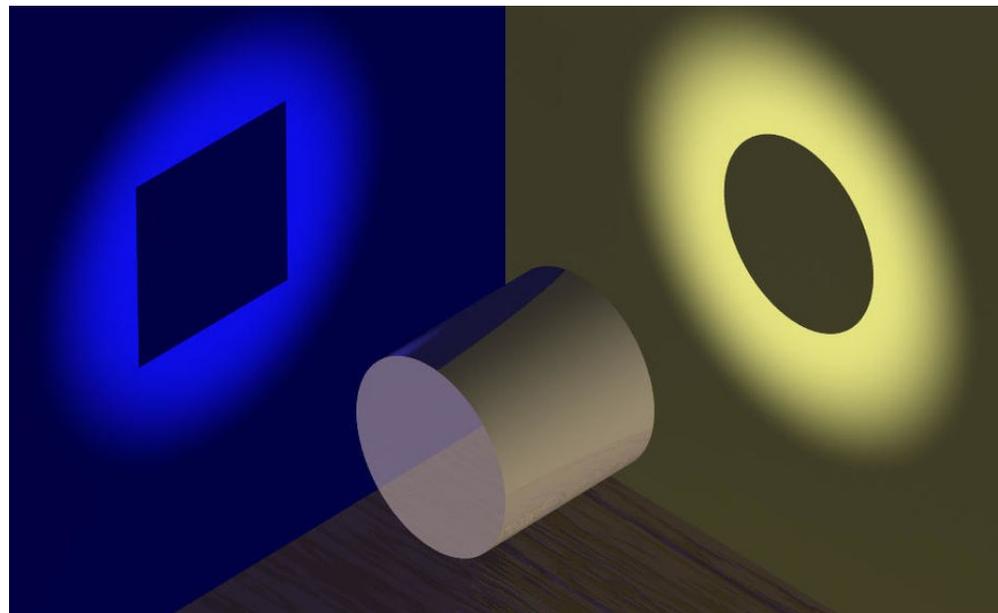
- k_1 is the slope at the beginning of the interval, using y (Euler's method);
- k_2 is the slope at the midpoint of the interval, using y and k_1 ;
- k_3 is again the slope at the midpoint, but now using y and k_2 ;
- k_4 is the slope at the end of the interval, using y and k_3 .



By HilberTraum - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=64366870>

La visualisation, c'est aussi : changer son regard...

- Changer de point de vue... Deux affirmations vraies, mais...
- Changer de perspective... Des faits à leur interprétation...



La visualisation, des caractères aux pixels : une déjà longue histoire...

Au commencement, il y avait la « console »...

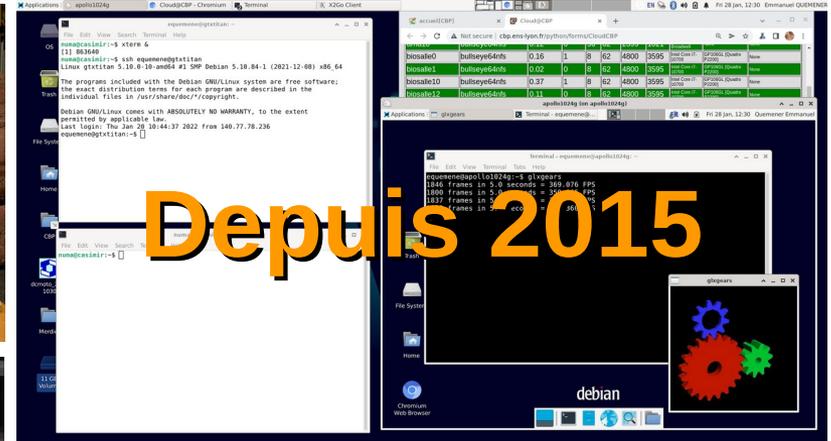
Accès
Distant



1970



1990



Depuis 2015

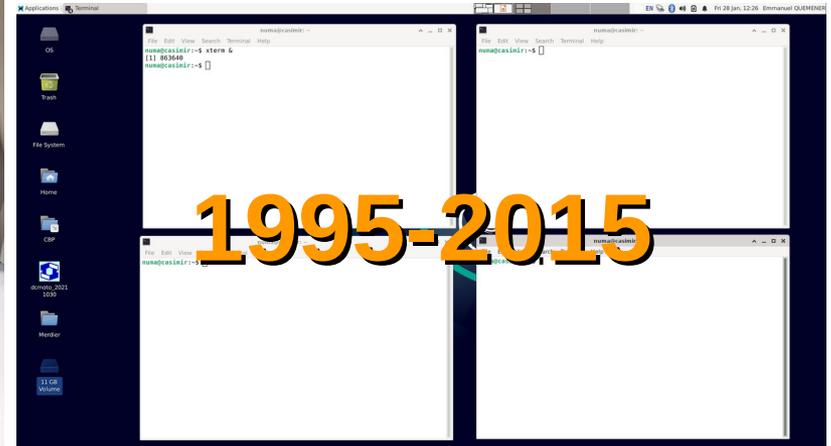
Accès
Local



1981



1984



1995-2015

Le Centre Blaise Pascal : c'est aussi ... plus de 300 machines actives

Cloud@CBP : État des ressources

Bonjour, utilisateur d'adresse IP 140.77.78.236.
Vous semblez surfer avec le navigateur Mozilla sous GNU/Linux

Le 2021-10-01, Heure Locale 18:02 131 machines "chargées" à 51.27 et utilisées par 72 utilisateurs
A cet instant, CPU : 208 sockets avec 2038 coeurs dans 54 modèles différents
le Cloud@CBP, c'est : GPU : 158 cartes dans 72 modèles différents.

Liens rapides : Configuration X2go Demande d'accès ou d'assistance

Sélection d'une machine

- Machine générique
- Machine multi-coeurs (n=32)
- Machine à grosse RAM (n=256GB)
- Machine avec gros GPU de Gamer
- Machine avec GPGPU (Hélio)

Submit Query Reset

Liste des machines avec caractéristiques techniques

Hostname	SIDUS	AvgtLoad	Users	Mem	Storage	OS	Model	Vendor	OS	Vendor
112473	bulseye4r4s	0.1	0	32	187	1300	1340	Intel Xeon E5-2680 v4	Linux	Red Hat
apollo1524g	bulseye4r4s	0.19	1	32	991	1000	2400	Intel Xeon E5-2680 v4	Linux	Red Hat
apollo151g1	bulseye4r4s	0.19	2	32	188	1467	989	Intel Xeon E5-2680 v4	Linux	Red Hat
apollo152g2	bulseye4r4s	0.32	3	32	188	1270	3417	Intel Xeon E5-2680 v4	Linux	Red Hat
apollo2048g	bulseye4r4s	2.25	1	32	1976	1270	13310	Intel Xeon E5-2680 v4	Linux	Red Hat
arma10	bulseye4r4s	0.16	1	56	62	2399	0	Intel Core i7-9700E	Linux	Red Hat

Servers@CBP : État des ressources

Bonjour, utilisateur d'adresse IP 140.77.78.236.
Vous semblez surfer avec le navigateur Mozilla sous GNU/Linux

Le 2021-10-01, Heure Locale 18:04 26 machines "chargées" à 22.58 et utilisées par 6 utilisateurs
A cet instant, CPU : 54 sockets avec 372 coeurs dans 17 modèles différents
le Servers@CBP, c'est : GPU : 36 cartes dans 5 modèles différents.
Stockage : 560 toques dans 32 pools et 685 datasets ZFS.

Liste des machines avec caractéristiques techniques

Hostname	AvgtLoad	OS	Wan	Wholesale	Online	Mem	Storage	OS	Model	Vendor	
hercule	0.0	0.01	0	12	12	31	1506	5	1	22	0
r410speed	2.96	0.05	2	9	12	62	2925	4	1	6	0
F510	0.0	0.0	1	11	12	62	1589	14	1	4	0
F510server1	0.41	0.19	8	9	12	62	1711	14	1	61	0
F510server2	0.21	0.04	1	10	12	62	2661	14	1	63	0
F510server3	0.0	0.02	0	29	12	62	2565	26	1	113	0
F510server4	3.43	0.0	1	9	8	70	2679	16	1	3	0
F410server1	0.0	0.03	1	10	8	47	1848	6	1	8	0
r420	0.14	0.06	1	10	20	125	1413	20	2	5	0
r720	2.46	0.03	2	10	8	503	3800	20	2	17	0
r720d	0.0	0.0	0	11	16	94	2900	14	1	1	0
r720d2	2.03	0.45	0	9	16	188	2900	38	3	5	0
r730server1	0.3	0.0	3	10	16	377	1265	8	1	11	0
r730server2	0.28	0.01	2	10	20	251	1766	8	1	7	0
r730server3	0.56	0.0	3	10	28	188	2900	8	1	5	0
r730server4	0.75	0.06	2	423	20	377	2600	68	5	187	0

Cluster@CBP : État des ressources

Bonjour, utilisateur d'adresse IP 140.77.78.236.
Vous semblez surfer avec le navigateur Mozilla sous GNU/Linux

Le 2021-10-01, Heure Locale 18:03 156 machines "chargées" à 24.10 et utilisées par 8 utilisateurs
A cet instant, CPU : 312 sockets avec 2416 coeurs dans 9 modèles différents
le Cluster@CBP, c'est : GPU : 5 cartes dans 3 modèles différents.

Liens rapides : Configuration X2go Demande d'accès ou d'assistance

Liste des machines avec caractéristiques techniques

Hostname	SIDUS	AvgtLoad	Users	Mem	Storage	OS	Model	Vendor	OS	Vendor
g510sd01	bulseye4r4s	0.32	0	12	23	2667	4507	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd10	bulseye4r4s	0.23	0	12	23	2667	4520	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd11	bulseye4r4s	1.83	0	12	23	2667	4670	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd13	bulseye4r4s	0.24	0	12	23	2667	4520	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd13	bulseye4r4s	0.67	0	12	23	2667	4468	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd14	bulseye4r4s	0.49	0	12	23	2667	4320	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd15	bulseye4r4s	0.26	0	12	23	2667	4400	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd16	bulseye4r4s	1.11	0	12	23	2667	4520	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd17	bulseye4r4s	0.35	0	12	23	2667	4337	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd18	bulseye4r4s	1.26	0	12	23	2667	4500	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd18	bulseye4r4s	0.37	0	12	23	2667	4444	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd19	bulseye4r4s	0.67	0	12	23	2667	4113	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd20	bulseye4r4s	0.79	0	12	23	2667	4500	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd21	bulseye4r4s	1.0	0	12	23	2667	4159	Intel Xeon E5-2680 v4	Linux	Red Hat
g510sd22	bulseye4r4s	0.26	0	12	23	2667	4293	Intel Xeon E5-2680 v4	Linux	Red Hat

- +4600 (vrais) coeurs, 28 TiB RAM, +1200 HDD, ~4PiB
- Récupération : PSMN, DSI, labos internes & externes...
- +130 machines directement accessibles en graphique !
- Accès multiples : physique, SSH, VirtualGL, x2go...

Pour bien visualiser, ce qu'il faut : ... ben une « bonne machine » !

- Dans le [cloud@CBP](#), 5 types de machines :
 - Des postes de travail de salles de formations
 - Des stations de travail dans des bureaux ou dans la salle serveur
 - Des serveurs équipés d'équipement généralement spécifiques
 - Des machines « ouvertes » sans boîtier
 - Des machines virtuelles à ressources dédiées
- La « bonne machine » :
 - C'est celle « adaptée » au « cas d'usage » : côté fonctionnel
 - C'est celle qui dispose de l'architecture spécifique : côté technique

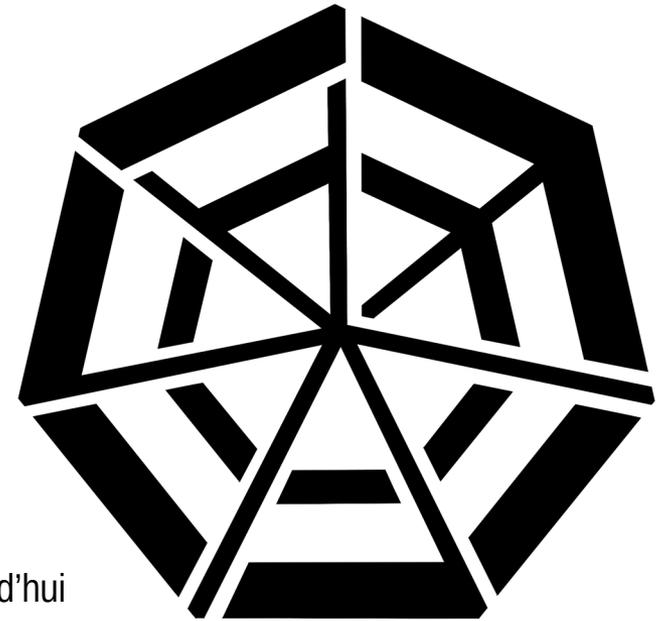
La « bonne machine » : la machine adaptée au « cas d'usage »

- Des composants : en reprenant Von Neumann
 - **Processeurs** : de 2 à 128 coeurs physiques, 55 modèles différents
 - **Mémoire vive** : de 16 GiB à 2 TiB
 - **Stockages réseau adaptés aux usages** : \$HOME, projets, scratch
 - **Stockage local** : de 500 GB à 50 TB, HDD, SSD, NVMe
 - **GPU ou GPGPU** : de 2008 à 2021 encore actifs,
 - de C1060 à A100, de HD7900 à RX6900XT, de 1 GB à 40 GB
 - **Système unifié** : SIDUS (couramment Debian Bullseye)
- De la diversité (des équipements) vient la force :
 - Et du système unifié vient la maîtrise...

Sur les Machines du CBP : SIDUS

Je n'installe pas, je démarre !

- **Quoi ?**
 - Déployer un système simplement sur un parc de machines
- **Pourquoi ?**
 - Assurer l'unicité des configurations
 - Limiter l'empreinte du système sur les disques
- **Pour qui ?**
 - Étudiants (vous quoi!), enseignants, chercheurs, ingénieurs, ...
- **Quand & Où ?**
 - Centre Blaise Pascal : depuis 2010, plus de 280 machines aujourd'hui
 - PSMN : depuis 2011, plus de ~800 nœuds (sa propre instance) aujourd'hui
- **Comment ?**
 - Utiliser un partage en réseau d'une arborescence
 - Détourner le mécanisme de LiveCD



« Deux machines ayant démarré SIDUS ne peuvent pas ne pas avoir le même système ! »

Bureau à distance : le couple x2go/VirtualGL

- Pourquoi visualiser ?
 - Parce que la puissance d'analyse est derrière les yeux !
- Pourquoi visualiser à distance ?
 - Parce que l'accès physique n'est pas possible 7j/7, 24h/24
 - Parce que c'est plus pratique pour suivre les évolutions d'un job...
- Quelles contraintes de la visualisation à distance :
 - Sur un canal RDP : multi-plateforme, assez efficace, restreint en 3D
 - Sur un canal TeamViewer : très efficace, mais double tunnel...
 - Sur un canal SSH : lourd, passage OpenGL, multi-plateforme difficile

VirtualGL : efficace mais pénible...

- Shading par la carte graphique
- Transport par canal SSH
- Utilisation :
 - Sous GNU/Linux, pas trop difficile :
 - Sur le poste client : `vglconnect -s <MonLogin>@<MaVisu>`
 - Sur le poste <MaVisu> : `vglrun <MonAppli3D>`
 - Sous les autres OS, bonne chance...

VirtualGL
3D Without Boundaries

x2go : la transformation d'une session en vidéo

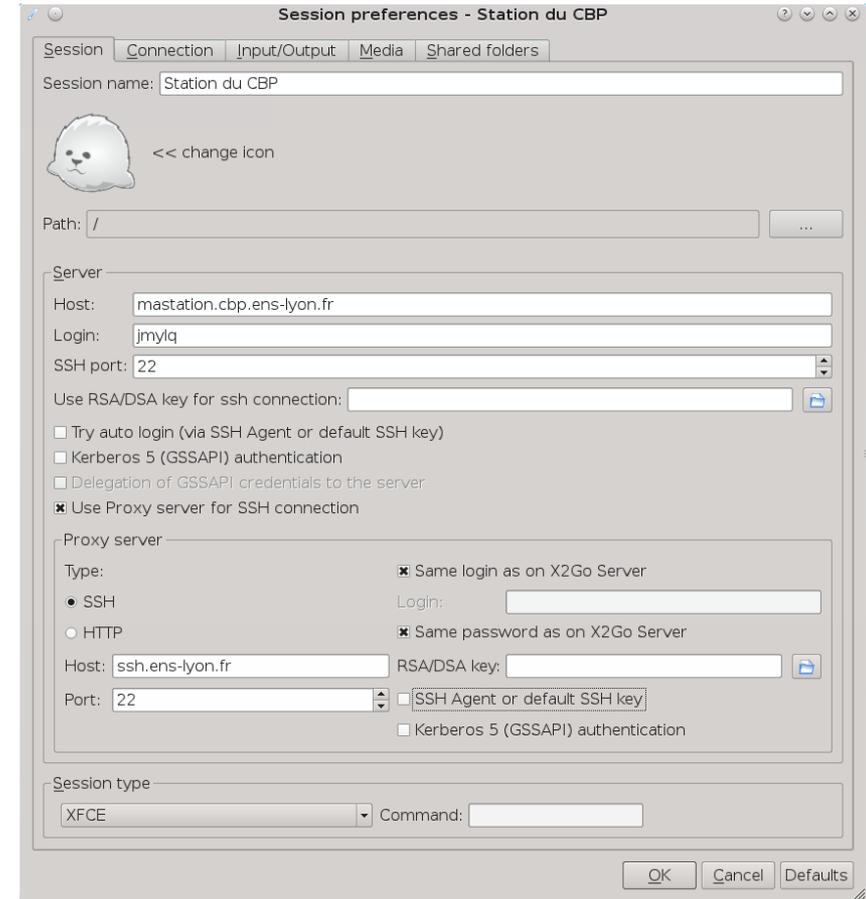
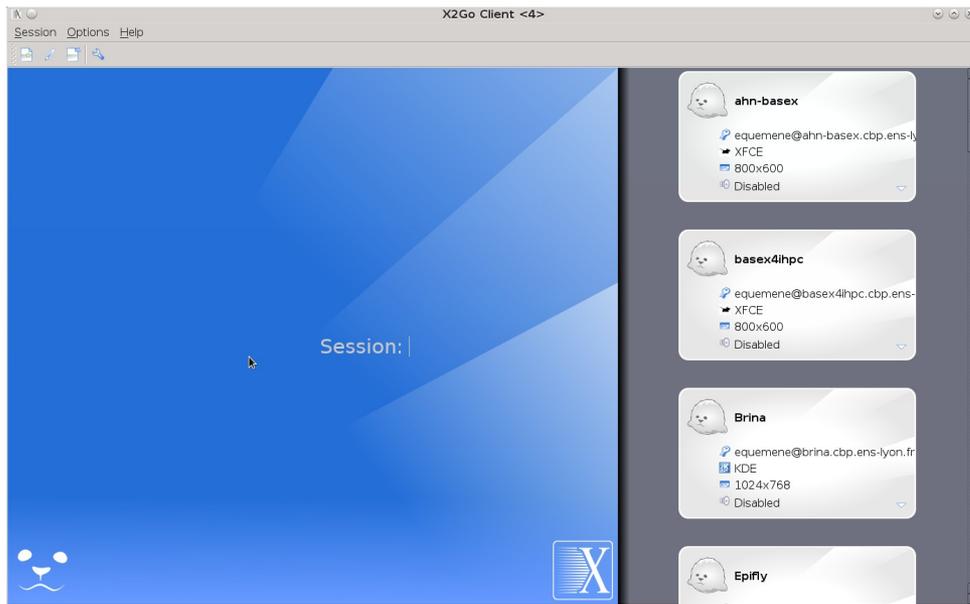
- 3 principes de base :
 - SSH, avec sa polyvalence, est le canal rêvé
 - Chaque image est transférée en jpg
 - Seule la modification de l'écran est transférée (comme MP4)
- Ses avantages :
 - Son multiplateforme : GNU/Linux, Windows, MacOSX, ChromeBook
 - Très faible bande passante utilisée : un 4G correct suffit
 - Passage de périphériques (son), espace de stockage, etc...
 - Exploitation de MesaGL pour la 3D, mais insuffisante...

x2go & VirtualGL :

enfin un mariage qui marche !

- Utiliser x2go pour :
 - Son multi-plateforme : Windows, MacOSX, GNU/Linux, ChromeBook
 - Sa faible bande passante utilisée (300KB/s pour une vidéo HD)
 - Son passage de périphériques :
 - Même la carte son peut passer !
 - Mais pour MacOSX, inférieur à 10.5 pour passer les périphériques
- Utiliser VirtualGL pour :
 - Les grosses applications 3D & GPU (Cuda & OpenCL) : MorphoGraphX, Paraview, VMD, ...
 - Préfixer « juste » la commande par vglrun
- Comme ça :
 - Transfert des données primaires plus accessibles
 - Exploitation plus rationnelle des stations de travail

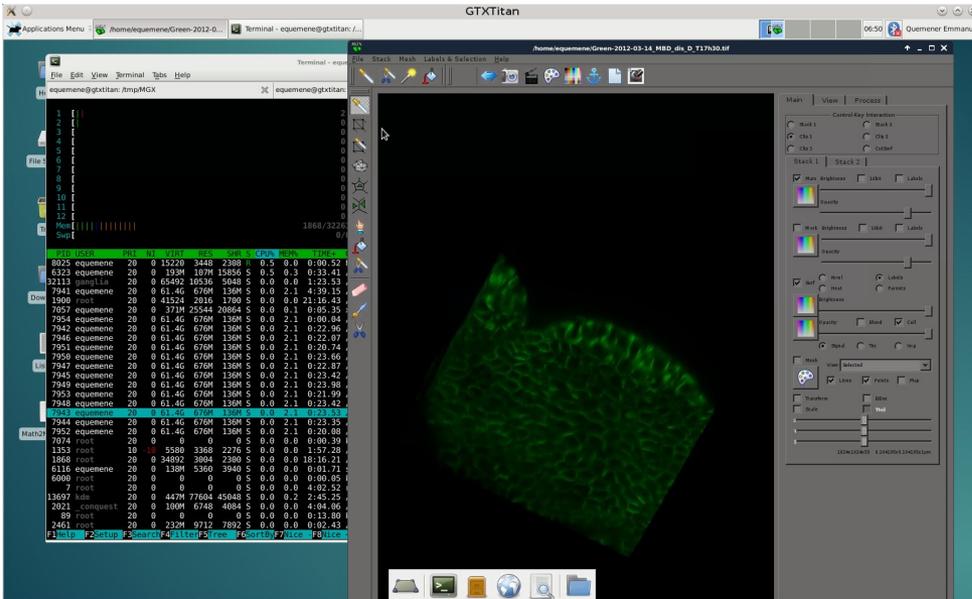
Connexion par x2go Une configuration aisée...



Le plus dur, c'est de suivre scrupuleusement la doc :-/

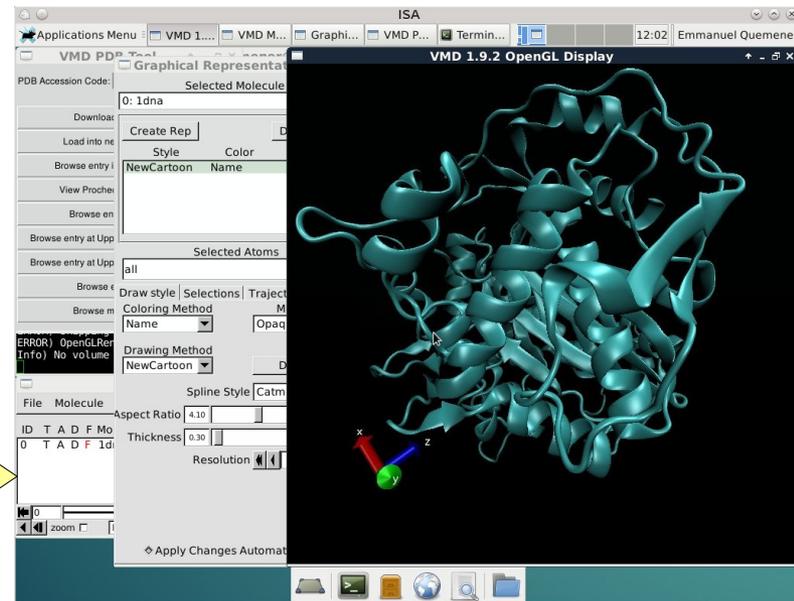
Exploitation x2go/VirtualGL

Pas seulement la visualisation...



Traitement
avec MorphoGraphX

Visualisation
avec VMD



Une petite « démo » ?

X2go & VirtualGL

Une petite démonstration ?

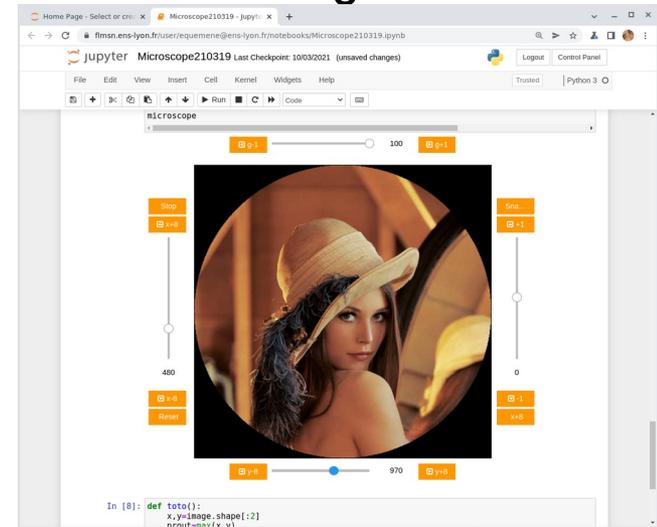
- Se connecter par x2go au travers d'une passerelle SSH
- Ouvrir une application graphique standard
- Ouvrir une application graphique avec accélération locale
- Appréciez ;-) !

De la diversité des « cas d'usage » à l'exigence d'espaces dédiés...

- Le Centre Blaise Pascal : une diversité insoupçonnée
 - +60 modèles de CPU, +100 modèles de GPU, +5 types de stockage, ...
- 3 approches proposées :
 - 1) Environnement Debian « de base » : le SIDUS de chaque machine
 - 2) Environnement Conda « commun » : Machine Learning & Visualisation
 - 3) Environnement Conda « projet » ou « jetable » : chacun le sien...
- Pour ses propres process (et visualisation) :
 - 1) Evaluer les 2 premières solutions, si insuffisantes (ou trop génériques)
 - 2) Créer son propre environnement : Conda, VirtualBox, Singularity...

JupyterHub : tous les notebook dans une approche « Full Web »

- A l'origine :
 - Demande Ralf Everaers (directeur CBP) pour ses Notebook
 - Requête de développement de l'UJM : application de biologie
- Authentification Eduroam :
 - Connexion indispensable au RADIUS
- Environnement CONDA spécifique
 - Pour éviter les problèmes de dépendances
- Fonctionnel depuis un an, mais encore peu exploité...



La 3D : demandes, moyens... Mais peu d'exploitation...

- Pourquoi ?
 - S'immerger dans la scène d'une visualisation physique
- Quoi ?
 - 1 station dans la salle de formation : k4000 + moniteur 24p
 - 1 station au M7-1H07 : k4000 + moniteur 27p
 - 4 paires de lunettes 3D
- Comment ?
 - Demander les lunettes
 - Accès direct obligatoire
- Applications validées : glxgears, PyMol, VMD, Paraview, ...



Les contraintes des lunettes 3D

Du « propriétaire » mais pas que...

- Disposer de :
 - Une carte Nvidia Quadro
 - Hors de matinfo parce que dans matinfo, livrée dans adaptateur PS2
 - Un moniteur avec un rafraichissement spécifique : au moins 120Hz
 - Un câble DVI bidirectionnel : pas si courant...
 - Des lunettes IR ou RF bien « chargées » : batterie dans les branches
 - Un vidéoprojecteur avec les bonnes spécifications (mais pas que...)
- Se placer bien en face (pour les IR)
- Utiliser les logiciels « prévus pour » : pas des masses...

3D : immersion avec casque VR

« *Work in progress...* »

- Pourquoi :
 - Créer une « cave » d'exploration de données avec partage de vidéo
- Comment :
 - 1) Trouver un équipement compatible GNU/Linux
 - 2) Déterminer les spécifications matérielles minimales
 - 3) Tester l'équipement sous Windows avec des logiciels éprouvés
 - 4) Passer sous GNU/Linux avec StreamVR
 - 5) Suivre les projets « scientifiques » exploitant les casques
- Projet suspendu : arrêté au point 3, 2020Q1

En conclusion

La visualisation au CBP : futur...

- Domaine en perpétuelle évolution
 - Les exigences évoluent, les équipements aussi, et je dois suivre !
- Le « Virtual Reality » devrait se développer :
 - Mais le support Open Source reste marginal (rédhibitoire...)
- Les accès distants sont déjà majoritaires :
 - Mais la présence de machines « physiques » reste indispensable

Appel aux dons !!!

Computhèque comme sanctuaire

- Qui pourrait me fournir les composants suivants :
 - Carte contrôleur IDE sur port ISA 16 bits, disquettes 5.24 pouces
- Autrement, la computhèque du CBP accueille :
 - Tout équipement informatique le plus ancien possible :
 - Les vieux 8 bits des années 1980 : Sinclair ZX, Commodore, Oric, etc...
 - Les vieux PC avec des cartes ISA : 80286, 80386, ...
 - Les vieux périphériques : SCSI, scanner, disques durs, lecteurs de bande, etc...
 - Tout équipement informatique un peu exotique :
 - Machines de technologie : Dec Alpha 21264, HPPA, Sun...
- Merci pour votre générosité : james.mylq@ens-lyon.fr