

Analyse de sensibilité globale

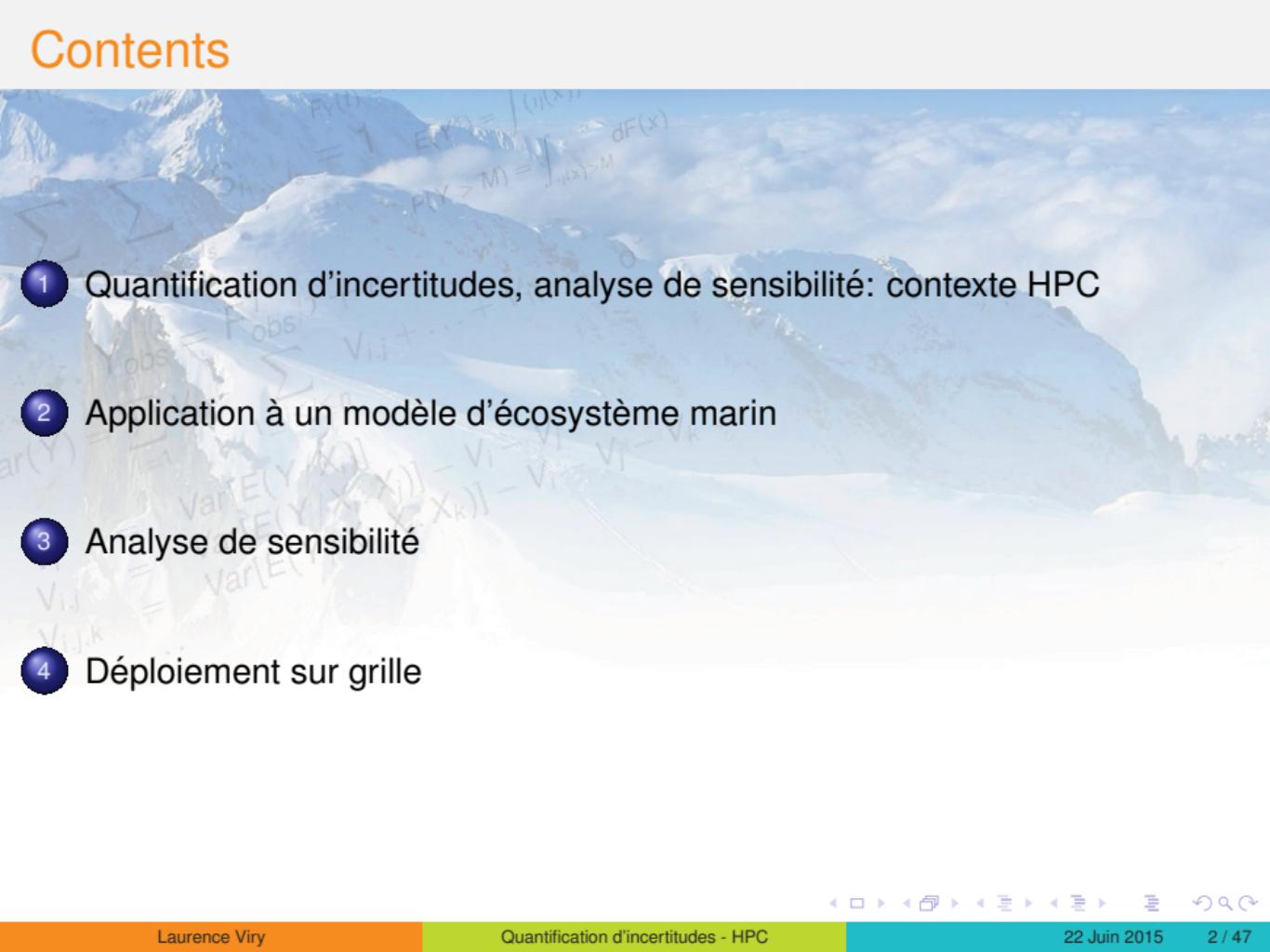
Application à un modèle d'écosystème marin

Laurence Viry

SMAI - AMIES

22 Juin 2015

Contents

- 
- 1 Quantification d'incertitudes, analyse de sensibilité: contexte HPC
 - 2 Application à un modèle d'écosystème marin
 - 3 Analyse de sensibilité
 - 4 Déploiement sur grille

Collaborations

Ce travail se fait au sein de l'équipe AIRSEA (LJK/INRIA) sur la thématique

Sensitivity analysis and uncertainty quantification for environmental modeling systems

en collaboration avec :

Clémentine Prieur (AIRSEA -LJK/INRIA)

Jean-Michel Brankart (LEGI -Grenoble)

Laurence Viry (AIRSEA - LJK/INRIA)

Bruno Bzeznik (CIMENT)

CIMENT : mésocentre Grenoblois

Sommaire

- 1 Quantification d'incertitudes, analyse de sensibilité: contexte HPC
- 2 Application à un modèle d'écosystème marin
- 3 Analyse de sensibilité
- 4 Déploiement sur grille

Quantification d'incertitudes et analyse de sensibilité

Phénomènes physiques \Rightarrow Modèles Mathématiques \Rightarrow Codes de Simulation

Le modèle mathématique étudié est représenté de façon générique par une fonction déterministe G définie sur un domaine de \mathbb{R}^p à valeurs dans \mathbb{R}^n .

- Les modèles sont **de plus en plus complexes** (non linéarité, couplage,...),
- ils prennent en compte **de nombreuses variables en entrée** et peuvent délivrer de **nombreux résultats en sortie**.
- Les codes peuvent être **gourmands en ressources informatiques** (CPU, stockage, communications,...).
- les sources d'erreurs sont **multiples**.

Motivations

Développer des méthodes performantes, même en grande dimension, permettant

➤ Quantification d'incertitudes

- Quantifier les incertitudes sur les entrées du modèle et estimer la part d'incertitude sur les sorties qui en résulte.

➤ Analyse de sensibilité

- Identifier les paramètres ou variables d'entrée qui ont une forte influence individuellement ou conjointement sur les sorties du modèle.
 ⇒ important de les connaître avec précision
- Identifier les paramètres ou variables d'entrée qui ont une faible influence sur les sorties du modèle.
 ⇒ moins important de les connaître avec précision, on peut les fixer
- Quantifier cette influence (corrélation, régression, indices de Sobol,...).
- Construire un modèle simplifié, un métamodèle

Sources d'incertitude

Toutes les sources d'incertitudes doivent être identifiées et caractérisées.
Elles peuvent être réparties dans ces trois catégories:

➤ Les entrées du modèle peuvent être aléatoires ou épistémiques.

- ① Géométrie, paramètres du modèle, conditions initiales.
- ② Données externes au système incluant conditions frontières, excitation du système.

➤ Approximation numérique

- ① Discretisation,
- ② erreur de convergence des méthodes itératives,
- ③ erreur d'arrondi,
- ④ erreurs dans le code (les effets sur la solution numérique sont difficiles à estimer).

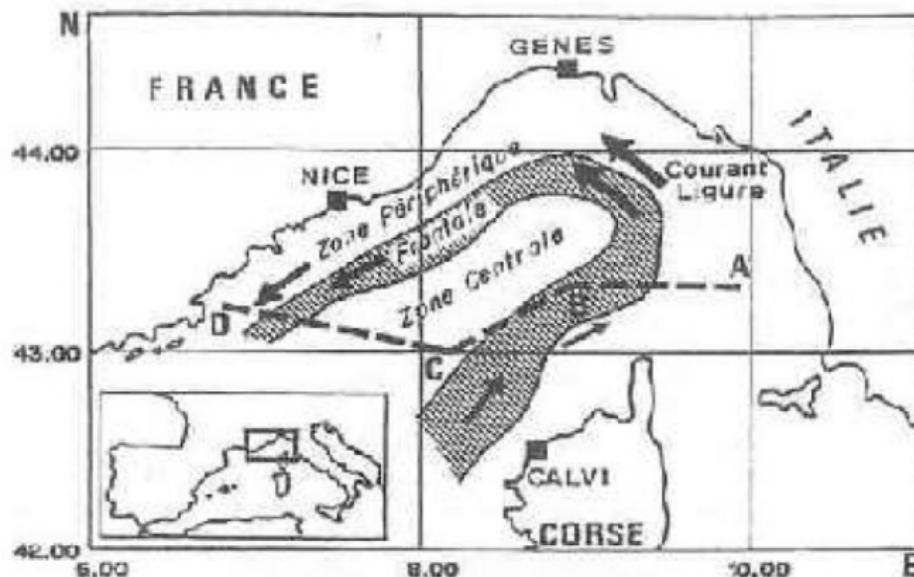
➤ Incertitude sur le modèle (hypothèses, abstractions, conceptions, formulation mathématique, . . .).

Sommaire

- 1 Quantification d'incertitudes, analyse de sensibilité: contexte HPC
- 2 Application à un modèle d'écosystème marin
- 3 Analyse de sensibilité
- 4 Déploiement sur grille

Modèle d'écosystème marin

MODECOGeL = MODèle d'ECOsystème du Gher¹ et du Lobepm²



¹GeoHydrodynamic and Environment Research Laboratory, Université de Liège, Belgique

²Laboratoire d'Océanographie Biologique et d'Ecologie du Plancton Marin, Université Pierre et Marie Curie

Présentation générale du modèle

Couplage entre un modèle hydrodynamique et un modèle biologique (Mer Ligure)

➤ Modèle hydrodynamique (équations primitives de l'océan 1D)

5 variables d'état

- température
- salinité
- vitesse horizontale (2 composantes)
- énergie cinétique turbulente

➤ Modèle biologique (biologie marine) 12 variables d'état

- nitrate, ammonium
- pico-, nano- et microphytoplanctons (Pp,Np,Mp)
- nano-, micro- et mézozooplanctons
- bactéries
- matières organiques particularisées
- azote organique dissous

Présentation générale du modèle (2)

- Modèle déterminisme (EDP discrétisées par volumes finis)
- 87 paramètres d'entrée fixés à leur valeur nominale (constante en temps et en espace) contrôlant les équations des processus biologiques (taux de croissance, température léthales, températures optimales, etc).
- Sorties spatio-temporelles : 1D en espace (profondeur de 0 à -200m) + temps (\sim 1 année).
- Temps de calcul (sur les machines de CIMENT en séquentiel)
 - basse résolution: \sim 5 secondes
 - haute résolution: \sim 55 secondes

Analyse de sensibilité globale

- On s'intéresse à la concentration en phytoplancton (3 types), bon témoin de l'activité biologique.

$$phyto(t, z) = (pp(t, z) + np(t, z) + mp(t, z)) * 1.59$$

- En particulier:

$$Y_{max} = \max_{t \in [0, 365]} phyto(t, 0)$$

$$Y_{moy} = \int_0^{365} phyto(t, 0) dt$$

$$Y_{prof} = \max_{t \in [0, 365]} \int_{-45}^0 phyto(t, z) dz$$

- Relativement à 79 paramètres dont la distribution est fournie par les physiciens.

Sommaire

- 1 Quantification d'incertitudes, analyse de sensibilité: contexte HPC
- 2 Application à un modèle d'écosystème marin
- 3 Analyse de sensibilité
- 4 Déploiement sur grille

3 types de méthodes

On choisira l'approche suivant trois critères.

- 1 Le coût en nombre d'évaluations du modèle,
- 2 la complexité du modèle,
- 3 le type d'information apportée (indices,...)

➤ **Cribleage** (OAT,Morris,...): méthode à faible coût permettant d'identifier les facteurs les plus influents. **Informations qualitatives**.

➤ **Analyse locale**: variabilité des sorties induite par une petite variation de x autour de x_0 . **Basée sur le calcul des dérivées**.

Approche locale, peut-être utilisée pour une grande nombre de variables par le calcul de l'adjoint.

➤ **Analyse globale** : variabilité de $G(x)$ induite par la variation de x sur l'ensemble de son domaine de variation. Approche stochastique, souvent très coûteuse.

Approche stochastique

➤ Pourquoi une approche stochastique

- Une approche déterministe suppose implicitement que les facteurs varient uniformément sur leur intervalle de valeurs. C'est une méthode locale.
- Une approche probabiliste $Y = G(X)$, X vecteur aléatoire contenant les paramètres incertains. On associe à X , une loi de probabilité.

$$F_Y(t) = P(Y \leq t) = \int_{G(x) \leq t} dF(x)$$

$$E(Y^k) = \int G(x)^k dF(x)$$

Calcul effectué par échantillonnage sur la loi de X

➤ Comment quantifier l'incertitude : il s'agit de localiser au mieux les valeurs de Y en caractérisant sa loi. Il est habituel d'utiliser la moyenne

$$\mu_Y = E(Y) = \int_{D_Y} y f_Y(y) dy$$

et la variance $\sigma^2 = Var(Y) = \int_{D_Y} (y - \mu_Y)^2 f_Y(y) dy$

1 sortie scalaire – Analyse de sensibilité

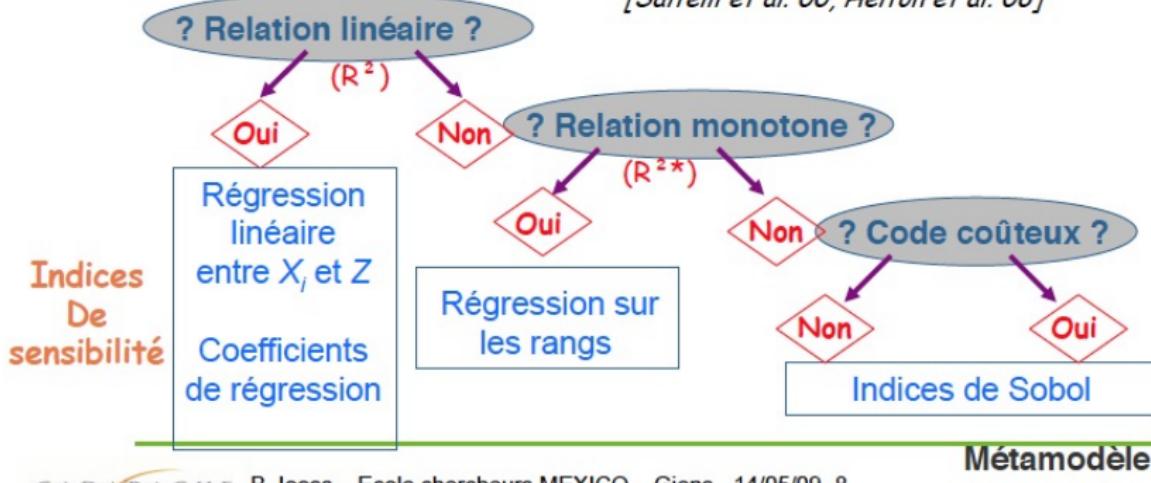
Échantillon ($X, Y(X)$) de taille $N > p$, de préférence de taille $N \gg p$
 $N = 300$ calculs de type Monte-Carlo (LHS)



Étape préliminaire : visualisation graphique (par ex : scatterplots)

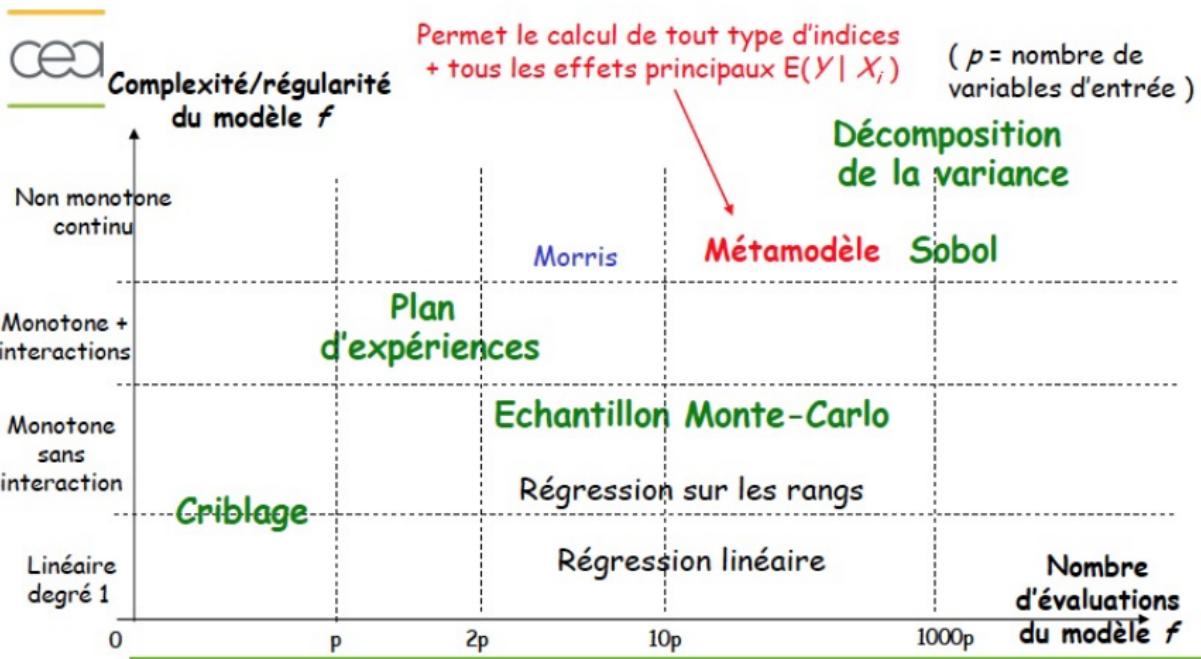
Méthodologie d'analyse de sensibilité quantitative

[Saltelli et al. 00, Helton et al. 06]



Classification des méthodes d'analyses de sensibilité

[Iooss 09]



Indices: Sans hypothèse sur le modèle

La variance de $Y = G(X_1, \dots, X_p)$ se décompose de manière unique.

$$\text{Var}(Y) = \sum_{i=1}^n V_i(Y) + \sum_{1 \leq i \leq j \leq p} V_{i,j}(Y) + \dots + V_{1,\dots,p}(Y)$$

où

$$V_i(Y) = \text{Var}[E(Y/X_i)]$$

$$V_{i,j}(Y) = \text{Var}[E(Y/X_i, X_j)] - V_i - V_j$$

$$V_{i,j,k}(Y) = \text{Var}[E(Y/X_i, X_j, X_k)] - V_i - V_j - V_k$$

Théorème de variance totale : les variables X_i sont indépendantes.

$$\text{Var}(Y) = E[\text{Var}(Y/X_i) + \text{Var}(E[Y/X_i])]$$

Cette décomposition mène naturellement à la définition des **indices de sensibilité de Sobol**, le nombre d'indices ainsi construits est égal $2^p - 1$

Indices de Sobol: Définition

On appelle indice de sensibilité d'ordre s

$$S_{i_1 i_2 \dots i_s} = \frac{V_{i_1 i_2 \dots i_s}(Y)}{\text{Var}(Y)} \quad \text{avec } 1 \leq s \leq p$$

$$\sum_{s=1}^n \sum_{i_1 < \dots < i_s} S_{i_1 \dots i_s} = 1$$

Chaque groupe de variables est responsable d'un pourcentage de variabilité de la sortie Y.

Indice totaux S_{T_i} avec $i = 1, \dots, p$

$$S_{T_i} = \sum_{u \subset \{0, 1, \dots, p\}, i \in u, u \neq \emptyset} S_u$$

Indices mesurant le poids de la variance de Y expliquée par chaque facteur en entrée du modèle et par ses interactions avec les autres facteurs.

Indices de Sobol: Modèle

- ✓ $\sum_{i=1}^n S_i = 1$ **modèle additif** : toute la variance est expliquée par les indices d'ordre 1.
- ✓ $\sum_{i=1}^n S_i \ll 1$ **modèle fortement couplé** : il faudra calculer les indices d'ordre supérieur.
- ✓ Les indices de Sobol s'adaptent aux phénomènes de non-linéarité et aux relations non monotones entre les sorties et les entrées.

Estimation des indices de Sobol

Le calcul des divers indices de sensibilité de Sobol se ramène au calcul numérique d'intégrales multidimensionnelles du type

$$I = \int_{[0,1]^P} G(x) dx$$

⇒ Difficultés lorsque le nombre de paramètres augmente

- L'échantillonnage devient inévitablement "creux"
- les erreurs proviennent du fait que certaines régions sont mal échantillonnées.

⇒ on a recours à des procédures de nature statistique de type Monte-Carlo

Estimation des indices de Sobol

Principalement deux types d'approches:

Approche spectrale (FAST,RBD,...) : analyse de Fourier

- **Avantages** : Estimations peu coûteuses
- **Inconvénients**
 - **Plus d'hypothèses de régularité**
 - Pas d'information disponible sur la précision des estimateurs
 - Pas d'estimation possible des indices supérieurs à 1

Approche de type Monte-Carlo

- **Méthodes de Sobol** (Sobol93,Saltelli2002,...) : à partir de deux plans d'expériences de type Monte-Carlo.
- **Méthodes à base d'hypercubes latins répliqués** (Mackay - 1997, Tissot-Prieur - 2012)
- ...

Estimation - Approche de type Monte-Carlo

➤ Méthodes de Sobol (Sobol93, Saltelli 2002, Jansen 1999,...)

Monte-Carlo + échantillonage dans l'espace des entrées

2 échantillons de taille n, coût $(2 + p) * n$ évaluations du modèle.

- **Avantages :**

- Estimations possible de tous les indices.
- Information sur la précisions des estimateurs
- **Inconvénients :** coût très important lorsque p devient grand.

➤ Hypercubes latins répliqués (MacKay - 1995, Tissot/Prieur - 2012)

Monte-Carlo + Hypercubes latins répliqués

- **Avantages :**

- Estimations moins coûteuses grâce à l'utilisation des hypercubes latins répliqués
- Information sur la précisions des estimateurs

- **Inconvénients :** Estimation des indices d'ordre 1.

Généralisation aux indices d'ordre 2 (Tissot-Prieur).

Méthodologies

➤ Caractéristiques de calcul:

- Temps d'une évaluation du modèle.
- Volume du stockage (échantillon, I/O du modèle, quantités d'intérêts).
- Communications: évaluation du modèle, distribution des données et des résultats.
- Environnement software, gestion de la complexité des processus.
- ...

➤ Les processus utilisés:

- (P_1) Paramétrisation, assigner des lois de probabilité aux variables du modèle étape de criblage éventuel.
- (P_2) Échantillonage ou planification à partir des lois des entrées.
- (P_3) Répartition des données en entrée du modèle et/ou du métamodèle.
- Évaluation du modèle (P_7) et/ou du métamodèle (P_8): calcul séquentiel ou parallèle.
- (P_4) Récupération des résultats (quantités d'intérêt).
- (P_5) Estimation des indices de sensibilité.
- (P_6) Construction d'un métamodèle (dans certains cas).

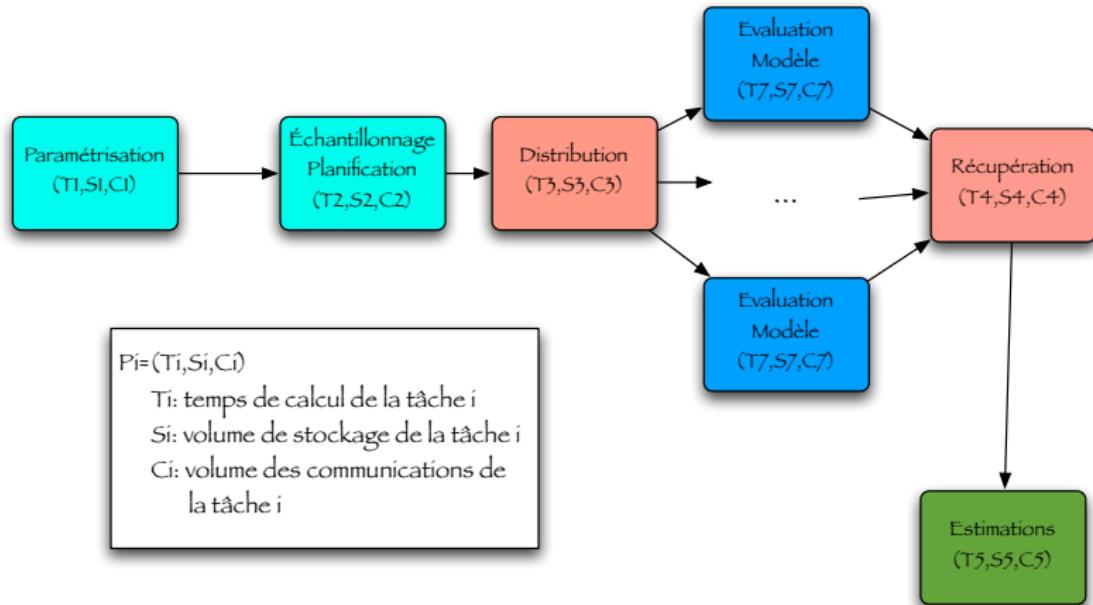
Analyse de sensibilité: Estimations à l'aide du modèle

Échantillon de taille n

$$\sum_{s=1}^n (X(w_1), \dots, X(w_n))$$

Estimations des quantités d'intérêts

$$\hat{\varrho}_n(G(X, \theta)) \rightarrow \varrho(G(X, \theta))$$



Analyse : Estimations à l'aide du modèle

➤ Temps de calcul global ($\textcolor{brown}{T}$):

$$\textcolor{brown}{T} = T_1 + T_2 + T_3 + n * T_7 + T_4 + T_5 \quad n \text{ évaluations du code}$$

Dépend de T_7 , de la taille de l'échantillon (n), du volume des I/O.

➤ Stockage ($\textcolor{blue}{S}$):

- S_1, S_2, S_5 : espace de stockage peu volumineux.
- S_7 : dépend du volume des I/O du code.
- S_4 : dépend du volume des quantités d'intérêts.

➤ Volume d'échange des données ($\textcolor{brown}{C}$), concerne les tâches P_3, P_4, P_7 .

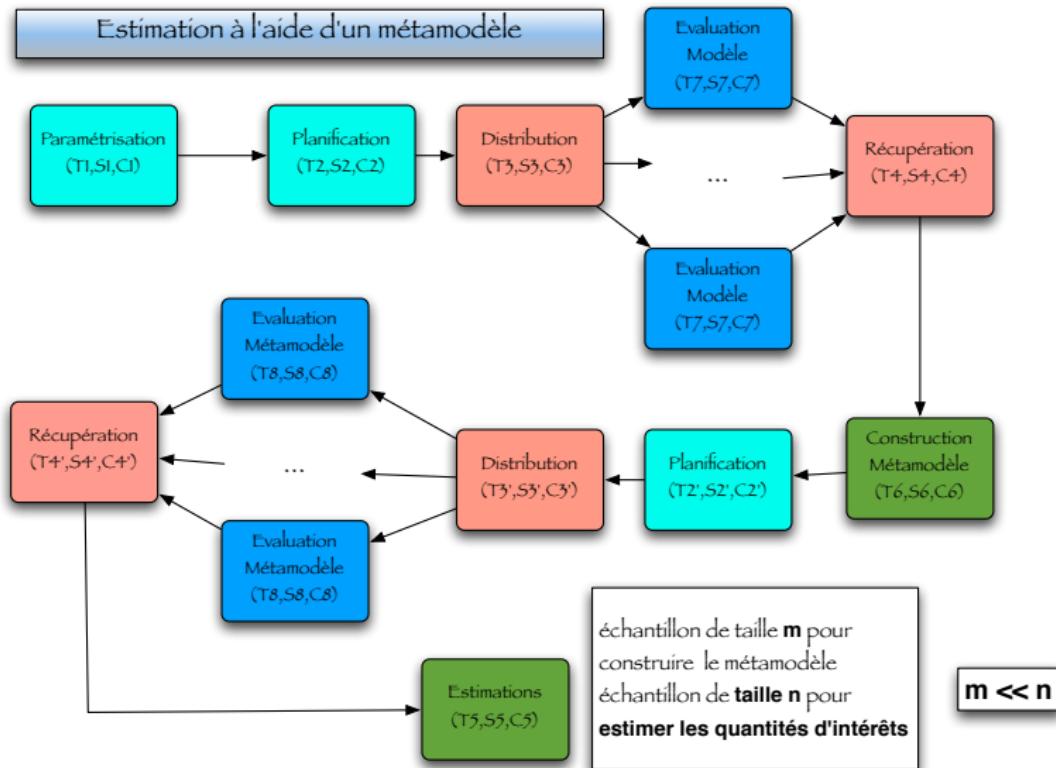
- C_3 : distribution des entrées et de l'échantillon des paramètres:
 $C_3 = \text{taille}(X) * n$
- C_4 : distribution des sorties du code et des quantités d'intérêt pour l'estimation des indices:
 $C_4 = \text{taille}(Y) * n$
- C_7 : dépend du code (parallèle).

➤ Solutions HPC suivant la complexité de l'application et des paramètres à estimer:

- Parallélisation du code.
- Distributions des calculs sur un gros cluster ou sur une grille de calcul.

Échantillon de taille m pour estimer le métamodèle

Échantillon de taille (n) pour estimer les quantité d'intérêts: ($n \gg m$)



Analyse : Estimations à l'aide du modèle

➤ Temps de calcul global (T):

$$T = T_1 + T_2 + T_3 + n * T_7 + T_4 + T_6 + T2' + T3' + m * T8' + T4' + T5$$

Dépend de T_7 et de la taille de l'échantillon (n).

➤ Stockage (S):

- $S_1, S_2, S_{2'}, SS_5$: espace de stockage peu volumineux.
- S_7 : dépend du volume des I/O du code.
- $S4, S4'$: volume des quantités d'intérêts.

➤ Volume d'échange des données (C), concerne les tâches P_3, P_4, P_7 .

- C_3 : distribution de l'échantillon des paramètres: $C_3 = \text{taille}(X) * n$
- C_4 : distribution des résultats du code pour l'estimation des paramètres:
 $C_4 = \text{taille}(Y) * n$
- C_7 : dépend du code.

Estimation des indices de Sobol

- Le nombre de variables (**p = 79**) est important, le coût d'évaluation du modèle est peu important, on a adopté une approche de type Monte Carlo par hypercubes latins répliqués (RLHS), moins coûteuse et adaptée aux modèles irréguliers.
- 1ère étape: calcul des indices d'ordre 1 à partir d'hypercubes latins répliqués de taille $n = 10^6$ observations:
 $\Rightarrow 2 * n (2 * 10^6)$ évaluations du modèle

La somme des indices d'ordre 1 est comprise entre 0.30 et 0.60

\Rightarrow il a été nécessaire d'estimer les indices totaux et les indices de deuxième ordre de chaque paramètre.

- 2 ième étape:

- Calcul des indices totaux: estimateurs de Jansen(*M.J.W Jansen 1999*)
 $\Rightarrow (2+p)*n$ évaluations du modèle
- Indices d'ordre deux: méthodes RLHS.

Mise en oeuvre du calcul des indices de Sobol

- Utilisation du package “sensitivity” de R pour:
 - Construire les plans d’expérience.
 - Traiter les données en sortie du modèle (valeurs aberrantes)
 - Calculer les indices de sensibilités: méthode RLHS (indices ordre 1), Jansen (indices totaux, indices d’ordre 1 et 2)
- Hypercubes latins répliqués ($n = 10^6$): $2 * 10^6$ évaluations du modèle.
 - Basse résolution: $2 * 10^6 * 1$ secondes: $\sim 555h$
 - Haute résolution: $2 * 10^6 * 40$ secondes: $\sim 22222h$
- Le nombre important de runs impose de porter une attention:
 - à la gestion des exceptions (exécution du code, soumission sur les machines de la grille, mouvements des I/O,...).
 - à la stabilité numérique du modèle dans l'espace de variation des paramètres (réécriture du code - J-M. Brankart).

⇒ L'utilisation d'une technique de déploiement sur grille (CIGRI/IRODS) est appropriée

Sommaire

- 1 Quantification d'incertitudes, analyse de sensibilité: contexte HPC
- 2 Application à un modèle d'écosystème marin
- 3 Analyse de sensibilité
- 4 Déploiement sur grille

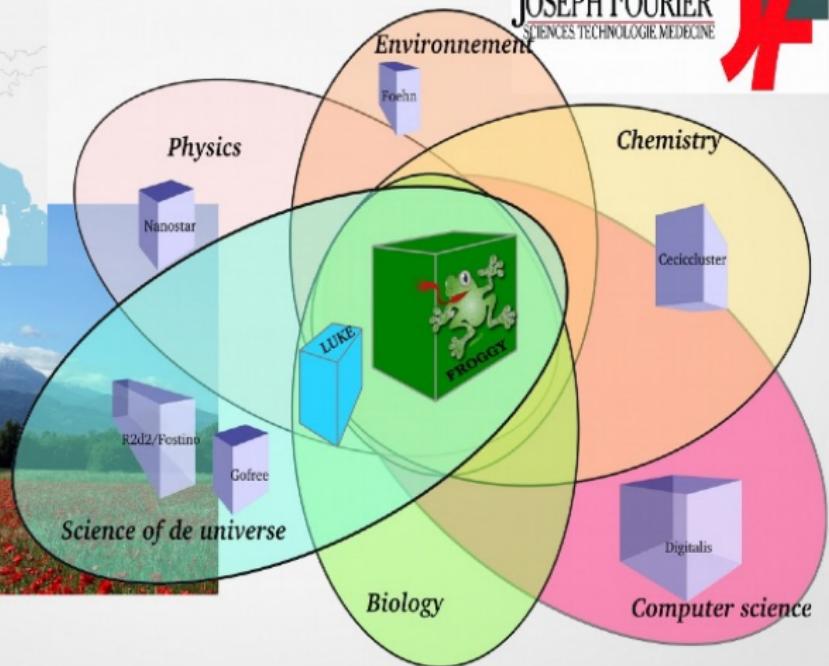
Déploiement sur grille - mésocentre grenoblois

CIGRI grille de calcul locale de CIMENT (mésocentre grenoblois).

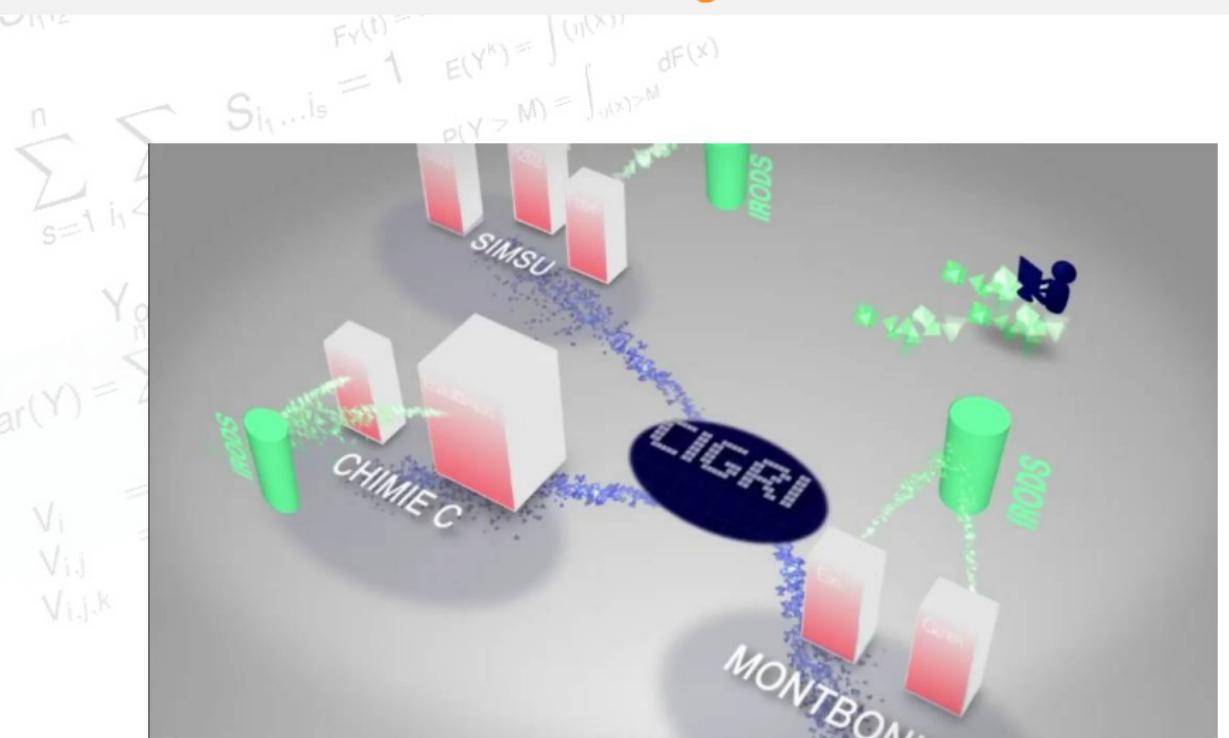
- Gérée par l'intergiciel **CIGRI**.
- Basée sur le gestionnaire de batch OAR.
- Mode de fonctionnement **besteffort** de OAR.
- Adapté aux **jobs multiparamétriques**.
- **IRODS** est un complément idéal de CIGRI
 - Récupération des données au fil de l'eau
 - L'accès aux données devient transparent, quel que soit le cluster qui les produit ou qui y accède
 - ...
- **Gestion des exceptions à l'exécution** en partie prise en charge par le middleware de la grille.

What is CIMENT?

CIMENT : High Performance Computing center of the univ. Grenoble-Alpes



Grille de calcul et de stockage de CIMENT



Computing platforms

HPC platform	Data processing platform	Other thematic platforms
  Froggy	 Luke	<i>r2d2, foehn, ceciccluster, digitalis, ...</i>
3200 Xeon E5 cores @2.6Ghz +18 GPUS K20m	~400 cores – heterogeneous systems and continuously evolving	~3000 cores heterogeneous systems federated from 10 clusters of member laboratories
High performance distributed storage (Lustre): 90 TB	Local scratches on nodes : 450 TB	NFS filesystems: a few TB per cluster
Infiniband FDR network	10 GE network	Infiniband QDR networks

Common storage (IRODS) : 1 PB



Au total en 2015

- 14 plateformes de calcul
- 6748 cpu-cores
- 24 Tera Bytes de mémoire
- 1,2 Peta Bytes de disques de stockage
- 120 Tflops (dont 20 Tflops pour les accélérateurs)

CIGRI: la grille de calcul

➤ OAR : gestionnaire de tâches et ressources installé sur tous les clusters de CIMENT (LIG/INRIA : projet Mescal)

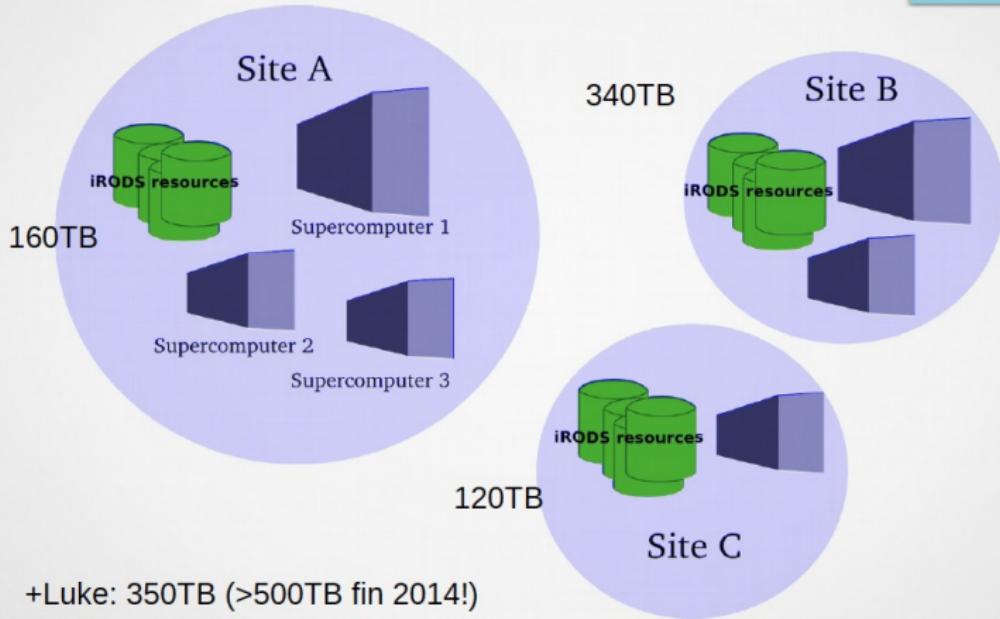
➤ Best-effort

- Un job best-effort a une priorité nulle
- Il est tué dès que d'autres utilisateurs ont besoin de ressources pour les jobs normaux de CIMENT.
- Un job best-effort doit être raisonnablement court et petit.

➤ CIGRI

- A la fois le nom d'un intergiciel de grille légère et le nom de la grille CIMENT
- Exploite très bien le mode best-effort de OAR, permettant ainsi de "remplir les trous"
- Très adapté aux jobs multiparamétriques, "embarrassingly parallel" (type Monte-Carlo par exemple)
- Permet de gérer des campagnes de plusieurs millions de "petits jobs"

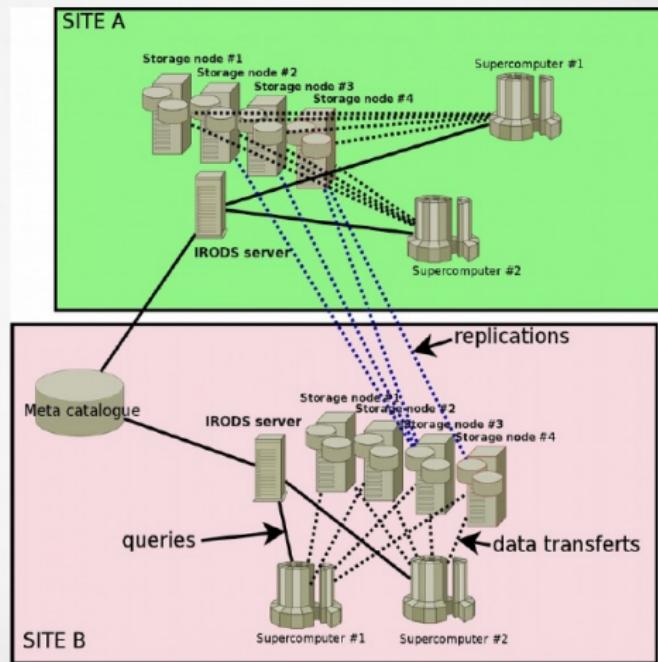
Grille de stockage (IRODS)



+Luke: 350TB (>500TB fin 2014!)

IRODS: principe de fonctionnement

- Ce n'est pas un système de fichiers !
 - On fait des « put » et des « get »
 - Nombreux moyens d'accès, dont une API python, une interface web, Webdav,...
 - Gestion de métadonnées (en SQL)



Lien entre CIGRI et IRODS

Le lien lien se fait:

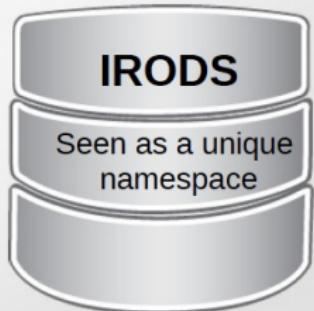
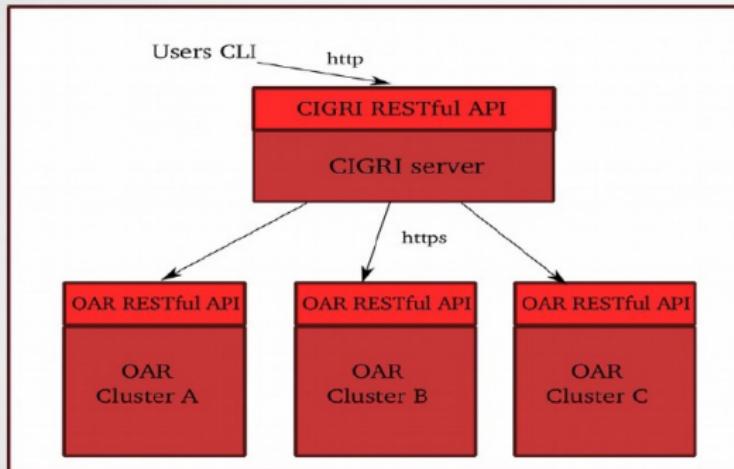
- ✓ par des scripts **prologue/epilogue**

- Déploiement de l'application et des fichiers communs.
- Initialisation de l'espace IRODS

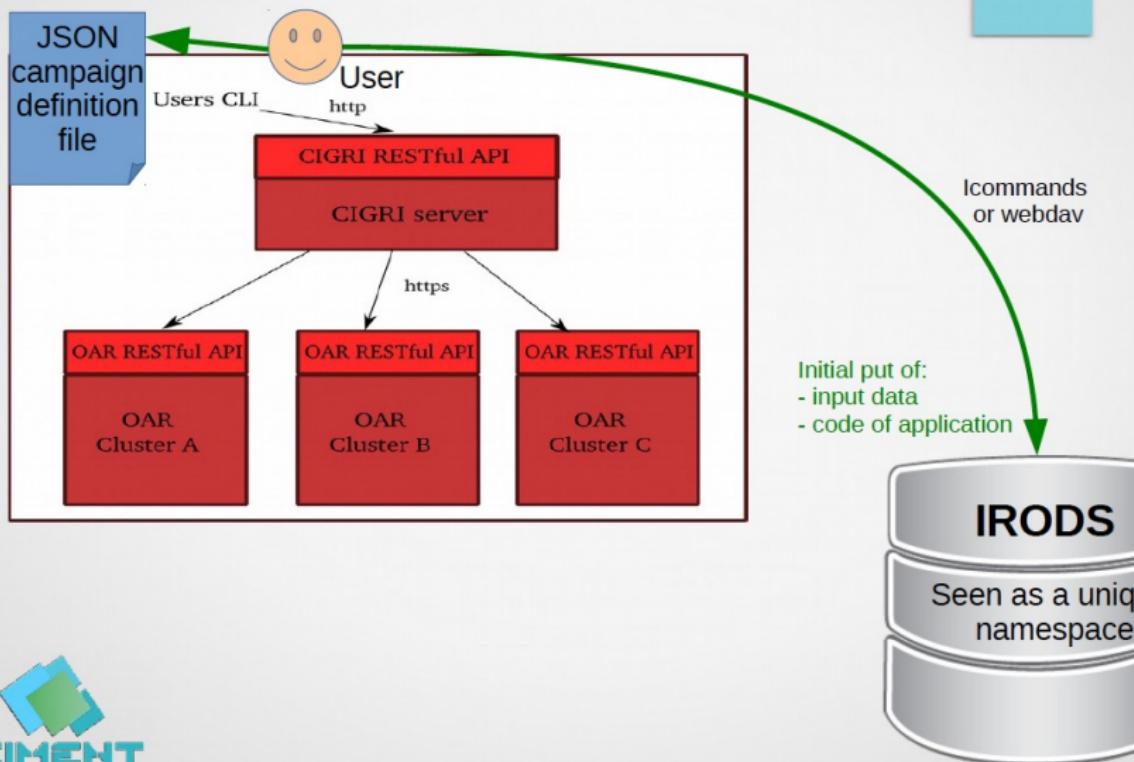
- ✓ par des scripts **de lancement de jobs**

- Récupération des données d'un job unitaire
- Récupération des résultats (sur IRODS par exemple)
- Affectation des méta-données.

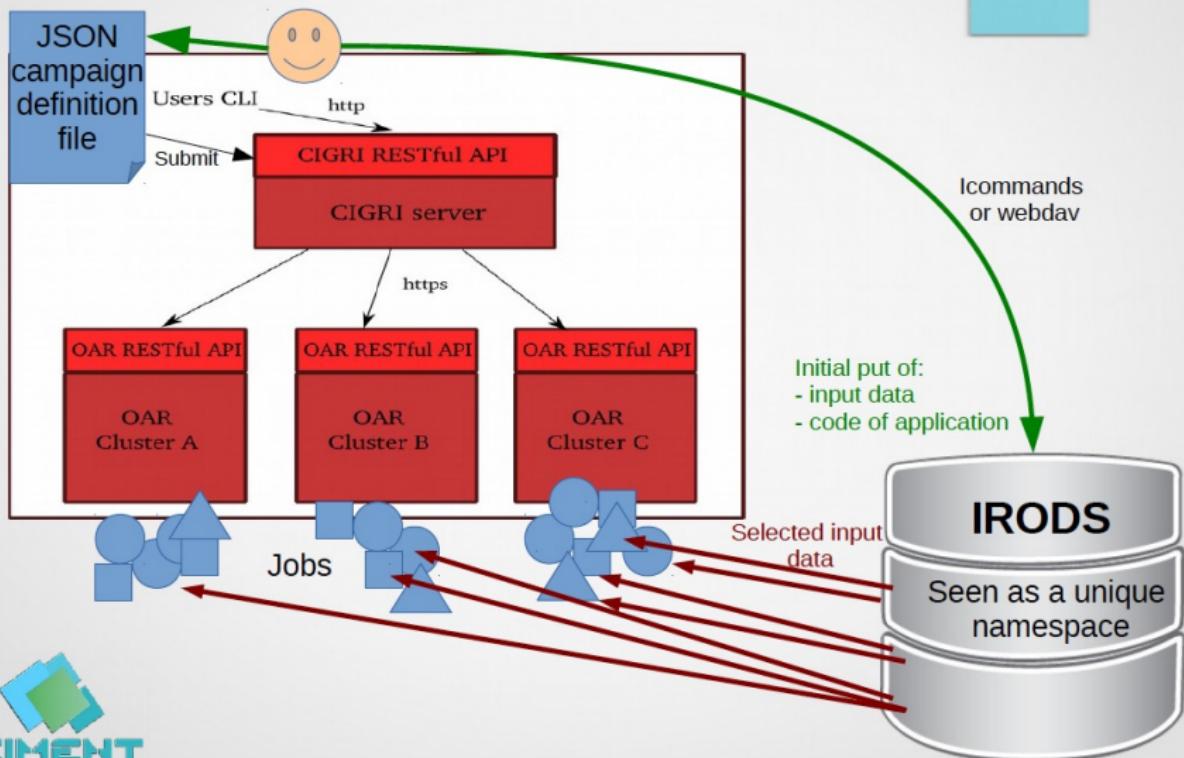
CIGRI and IRODS



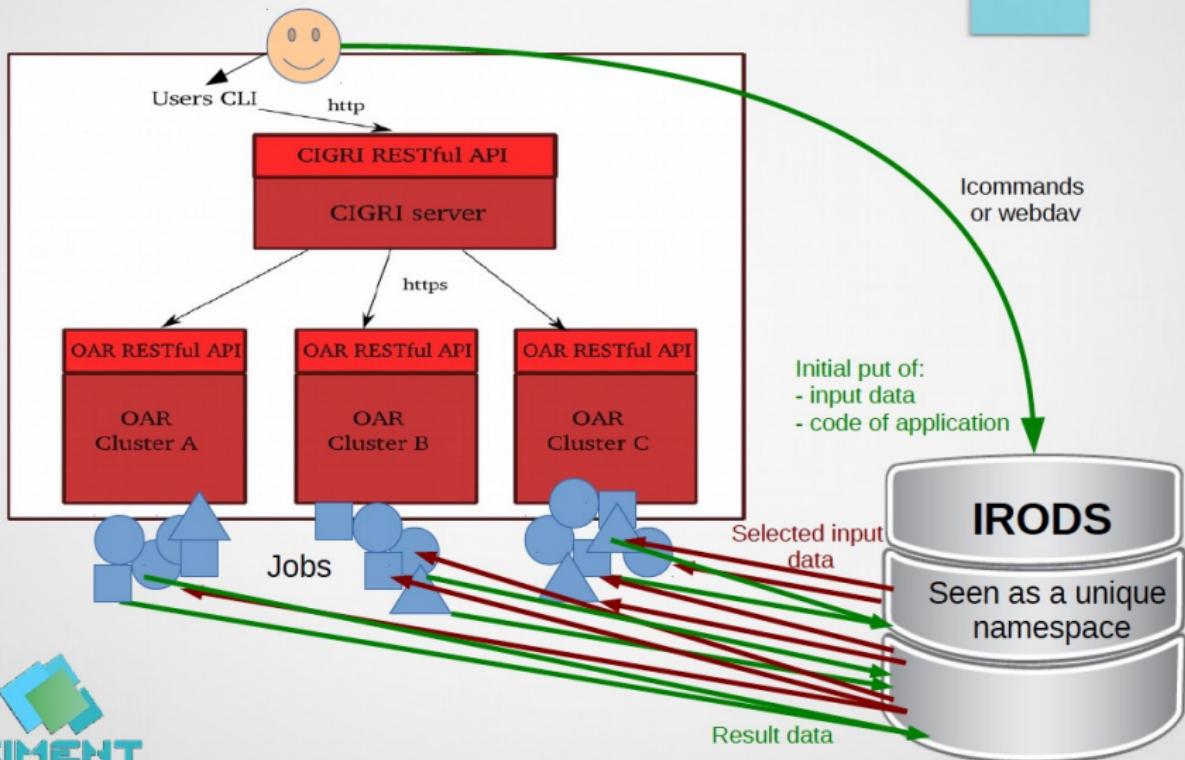
CIGRI and IRODS



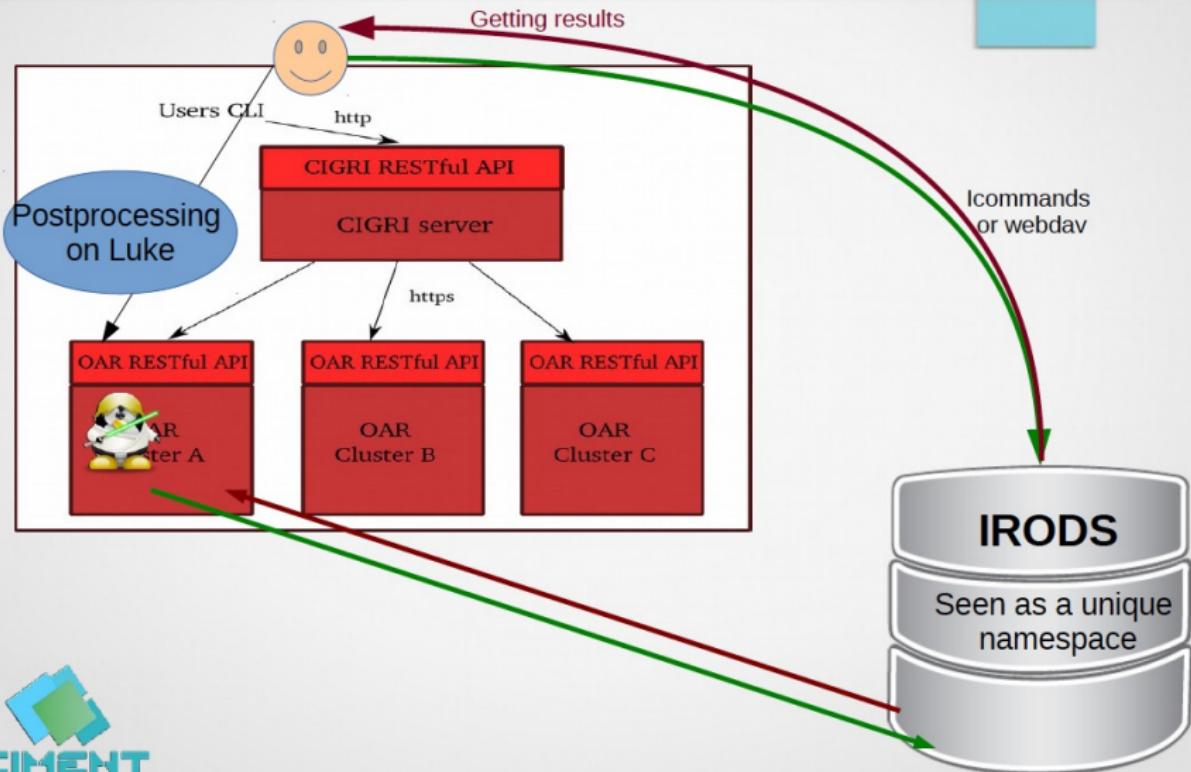
CIGRI and IRODS



CIGRI and IRODS



CIGRI and IRODS

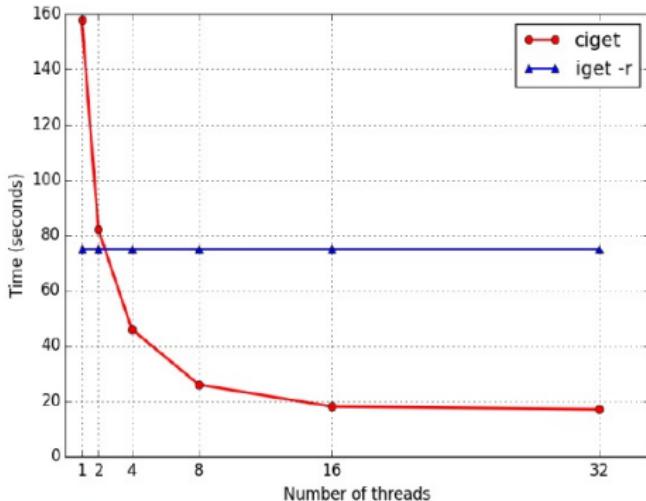


How does IRODS work on CIMENT?

Cirods : a small library based on pyrods to parallelize operations on large sets of files < 100kB

- “ciget.py” vs “iget -r”

ciget.py compared to 'iget -r' with small files
Test with a collection of 10000 files of 60kB



- Parallelization on files:
1 thread → 1 file
- Meta-data creation test:
 - ~2600 / s
 - From an input file (ease of use)

$S_{i_1 i_2 \dots i_n} = \frac{V_{i_1 i_2 \dots i_n}}{\text{Var}(Y)}$

$F_Y(t) = P(Y \leq t) = \int_{v_i(x) \leq t} dF(x)$

$E(Y^k) = \int (v_i(x))^k dF(x)$

$P(Y > M) = \int_{v_i(x) > M} dF(x)$

$\sum_{s=1}^n \sum_{i_1 < \dots < i_s} S_{i_1 \dots i_s} = 1$

$Y_{\text{obs}} = F_{\text{obs}}^T \beta + \epsilon$

$\text{Var}(Y) = \sum_{i=1}^n V_i + \sum_{1 \leq i \leq j \leq n} V_{i,j} + \dots + V_{1,n}$

$V_i = \text{Var}[E(Y|X_i)]$

$V_{i,j} = \text{Var}[E(Y|X_i, X_j)] - V_i - V_j$

$V_{i,j,k} = \text{Var}[E(Y|X_i, X_j, X_k)] - V_i - V_j - V_k$

MERCI

Questions?