Calcul Haute Performance avec OpenTURNS

Renaud Barate – EDF R&D

Workshop du GdR MASCOT-NUM

« Quantification d'incertitude et calcul intensif »

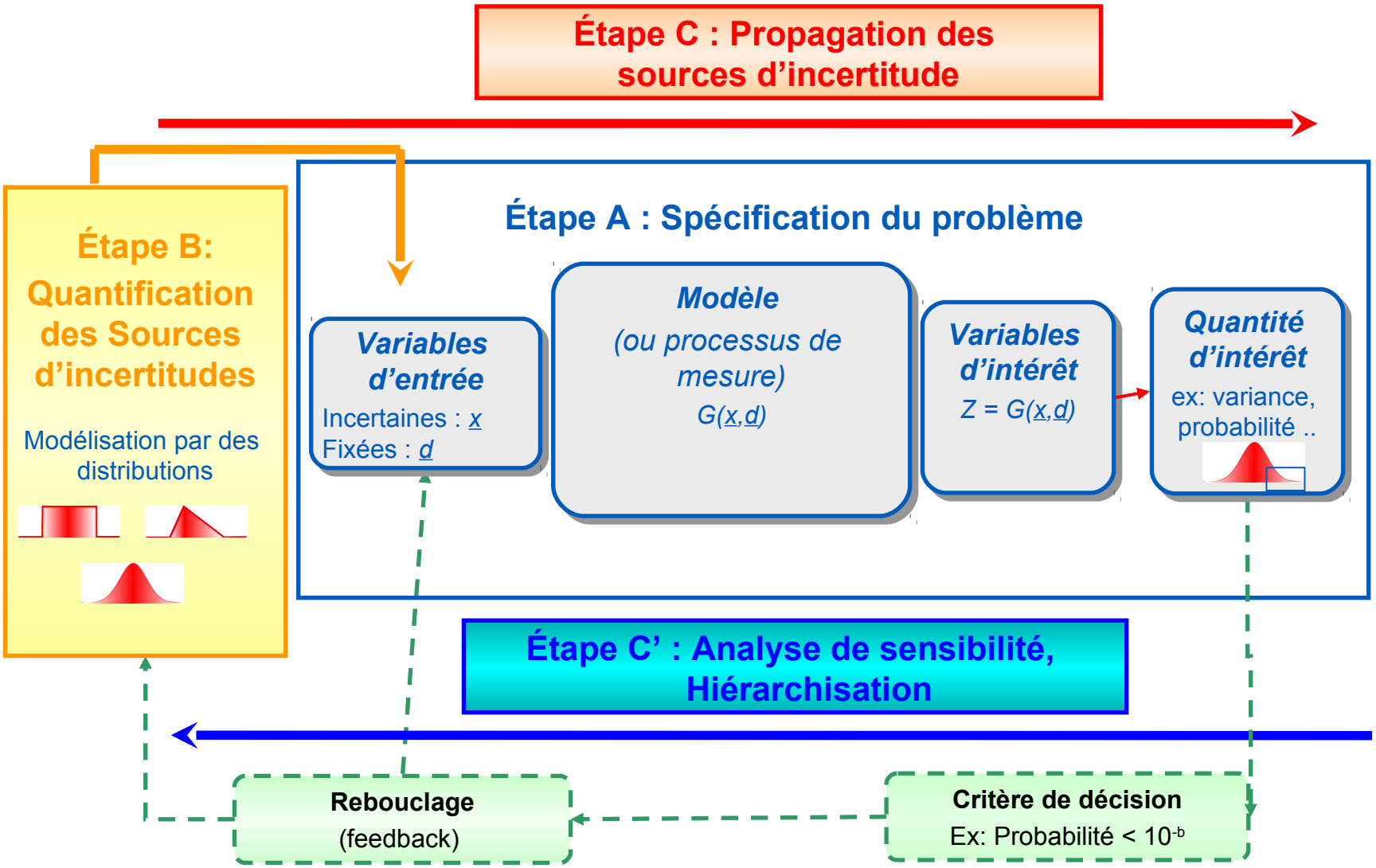
28 Mars 2013

CHANGER L'ÉNERGIE ENSEMBLE

Sommaire

- Présentation du logiciel OpenTURNS
- Problématiques du Calcul Haute Performance avec OpenTURNS
- Solutions pour le Calcul Haute Performance avec OpenTURNS
- Exemples d'application

Cadre méthodologique (1/2)

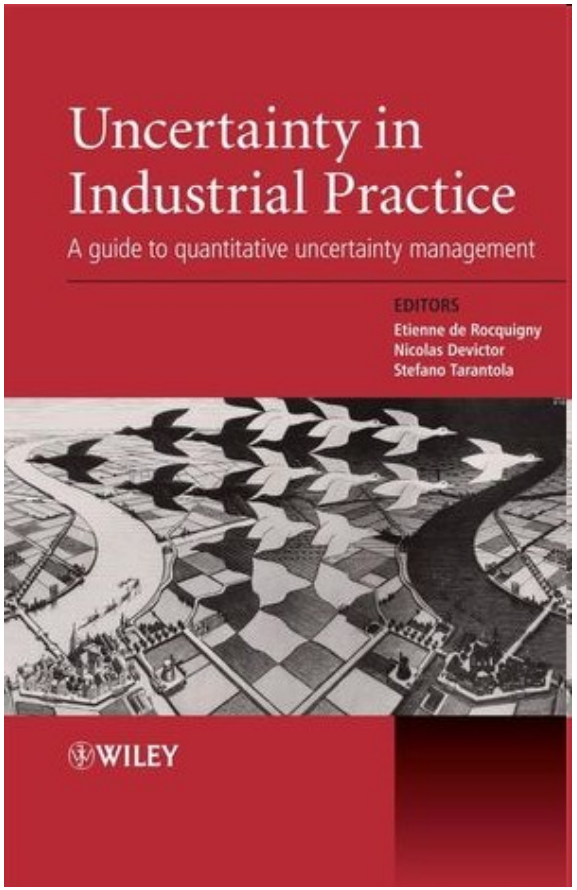


Cadre méthodologique (2/2)

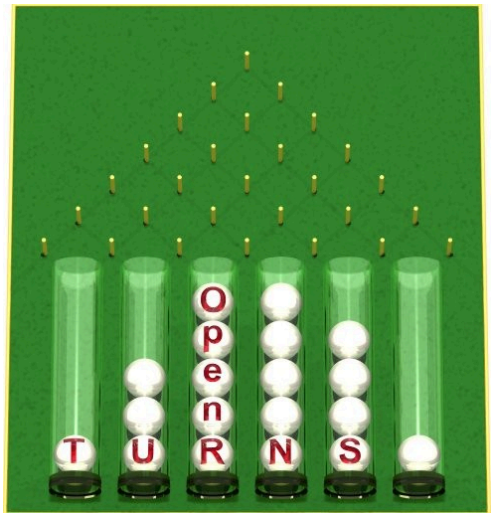
Un cadre méthodologique largement partagé, issu de réflexions au sein de groupes de travail :

EDF, EADS, Dassault-Aviation, CEA, Hispano-Suiza, JRC …

Ouvrage de référence
(GT ESReDA, 2005-2008)



Open TURNS - l'outil de mise en œuvre informatique de la méthodologie « incertitudes »

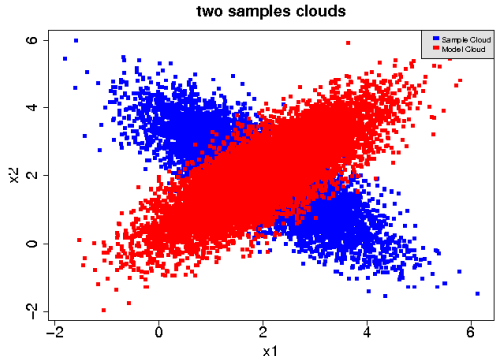


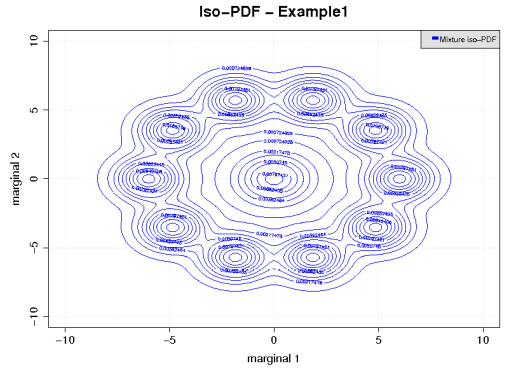
- TURNS : Treatments of uncertainties, risk'n statistics (2007)
- Open : Open source → LGPL, FDL
- Langages : Python (Interface texte), C++ (librairie)
- Partenariat EDF-EADS-Phimeca dep. 2005
- Déploiements
  - Linux (intégré aux distr. Debian)
  - Windows
  - Disponible comme composant de la plate-forme SALOME

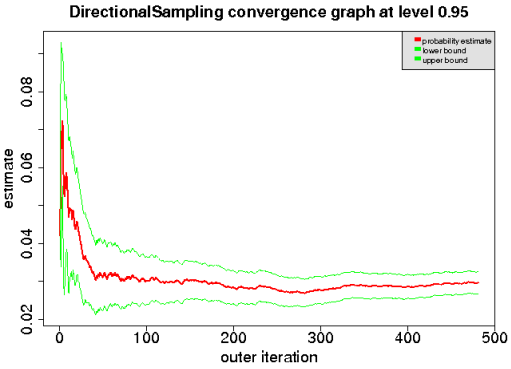
www.openturns.org

Open TURNS - particularités

- Peut propager les incertitudes à travers des codes de calcul externes (vus comme des « boîtes noires »)
  - Approche non-intrusive
- Outil « Open Source » expressément dédié au traitement des incertitudes
- Permet de décliner la méthodologie « Incertitudes » grâce à une large palette de méthodes
  - Prise en compte explicite de la dépendance entre les entrées X
- Structure modulaire (sur le principe de R, Scilab …)







OpenTURNS - interfaçage avec les codes

- Deux possibilités pour appeler un solveur externe
  - Appel d'une fonction Python
  - Utilisation d'un système de « wrapper »
    - Fichier de description XML (variables échangées, fichiers, type d'interfaçage)
    - Interface logicielle (bibliothèque dynamique permettant d'appeler du code C, C++, Fortran ou un exécutable)
- Gestion des fichiers d'échange avec le code
  - Génération de fichiers d'entrée à partir de modèles (templates)
  - Extraction de valeurs depuis des fichiers de sortie
- Gestion des erreurs

Sommaire

- Présentation du logiciel OpenTURNS
- Problématiques du Calcul Haute Performance avec OpenTURNS
- Solutions pour le Calcul Haute Performance avec OpenTURNS
- Exemples d'application

Calcul Haute Performance pour OpenTURNS

- OpenTURNS, cible « idéale » pour le Calcul Haute Performance (HPC)
  - En général, appel d'un grand nombre de calculs indépendants (algorithmes de type Monte Carlo)
  - Interfaçage de type « boîte noire »

- Temps de calcul des algorithmes d'OpenTURNS généralement négligeable par rapport aux appels au code de simulation
  - Algorithmes multi-threadés (sur une seule machine)
  - Appels au code de simulation distribués

Problématiques associées au HPC

- Utilisation des ressources distribuées très liée au contexte :
  - Utilisation d'un cluster (homogène, centralisé) / d'une grille (hétérogène, décentralisé) ?
  - Protocole pour la communication avec le cluster ?
  - Quel gestionnaire de batch / de grille ?
  - Possibilité d'installer des logiciels sur le cluster ?
  - Système de fichiers unique / par nœud ?
  - Exécution du script OpenTURNS sur le poste client / sur le cluster ?
  - Intergiciel pour la distribution sur le cluster ?
  - Taille des fichiers d'entrée et de sortie du code de calcul ?
- Pas de solution unique adaptée à tous les contextes

HPC : Administration et environnement logiciel

- Cas 1 : Possibilité d'installer des logiciels sur les clusters
  - Possibilité d'exécuter OpenTURNS dans un job sur un nœud du cluster
  - Nécessite une couche logicielle supplémentaire pour distribuer les calculs
    - MPI
    - CORBA
    - Parallel Python
    - Connexions SSH

- Cas 2 : Impossibilité d'installer des logiciels sur les clusters
  - Nécessité d'exécuter OpenTURNS sur un poste client
  - Gestion de la soumission et du suivi des jobs depuis le wrapper OpenTURNS
    - Nécessite une interface avec le gestionnaire de batch (ex : DRMAA + implémentation adaptée, libbatch)
  - Gestionnaires de batchs souvent mal adaptés à la soumission de milliers de jobs
  - Nécessité de laisser le poste client connecté durant toute l'étude

HPC : Transfert de fichiers

- Données nécessaires à OpenTURNS relativement faibles (listes de réels)
- Mais nécessité de générer des fichiers d'échange avec le code de calcul (en général)
- Transfert de millions de fichiers rédhibitoire

- Nécessité de générer les fichiers directement sur le cluster
  - Si possibilité d'installer des logiciels, utilisation d'un « double wrapper »
  - Si impossibilité d'installer des logiciels, remplacement du mécanisme de substitution d'OpenTURNS par des scripts (bash, perl, awk, etc. en fonction de l'environnement)

Principe d'un wrapper pour le calcul distribué

- Possibilité d'écrire un wrapper OpenTURNS pour gérer la distribution
  - Appels au gestionnaire de batch ou de grille (DRMAA, SAGA, libBatch)
  - Génération et transfert des fichiers d'entrée et de sortie du code
  - Possibilité d'utiliser un « double wrapper » pour exécuter OpenTURNS sur un poste client mais générer les fichiers sur le cluster
  - Possibilité d'interfaçage avec MPI ou Parallel Python
  - Etc.

- Mais divergences en fonction de l'environnement logiciel et matériel
  - Différents wrappers adaptés à des environnements différents

Sommaire

- Présentation du logiciel OpenTURNS
- Problématiques du Calcul Haute Performance avec OpenTURNS
- Solutions pour le Calcul Haute Performance avec OpenTURNS
- Exemples d'application

Solution 1 : Fonction Python distribuée

- Fonction Python distribuée
  - Incluse en standard dans OpenTURNS depuis la version 1.1 (Janvier 2013)
  - Solution documentée, testée et maintenue
- Choix techniques :
  - Communication SSH entre les nœuds de calcul
  - Pas de gestion de la soumission de job (Soumission effectuée par l'utilisateur)
  - Installation d'OpenTURNS nécessaire uniquement sur le nœud maître
  - Fonctionne même avec des systèmes de fichiers non partagés entre les nœuds
- Solution simple et rapide à mettre en œuvre

Solution 2 : Intégration d'OpenTURNS avec SALOME

Principes et objectifs

- Étude complète réalisée dans une plate-forme unique
- Intégration avec OpenTURNS facilitée pour les codes de calcul déjà intégrés à SALOME
- Fonctionnalités incluses dans SALOME pour la distribution de calculs
- Lancement de calcul distant depuis l'interface graphique de Salome grâce au module JobManager
- Possibilité de réaliser une étude d'incertitudes sur des schémas de calcul (couplages de codes)
- Solution industrielle, testée, documentée et maintenue

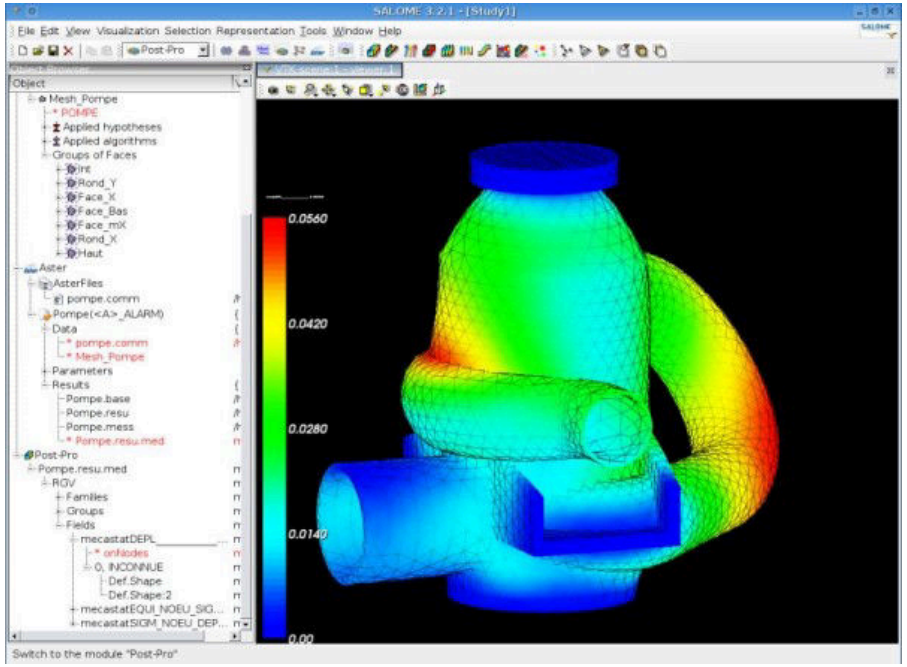
Choix techniques

- Exécution du schéma de calcul sur le cluster comme un unique job
- Nécessite l'installation sur le cluster de la plate-forme SALOME intégrant OpenTURNS
- Possibilité d'utiliser différents protocoles entre les nœuds (SSH, srun, pbsdsh, …)
- Gestion de la communication avec le cluster (SSH ou RSH)
- Gestion du transfert de fichiers
- Soumission et suivi du job
- Abstraction des gestionnaires de batch (LSF, PBS, Slurm, SGE, LoadLeveler, OAR)

Solution plus complète pour la construction de plate-formes métiers

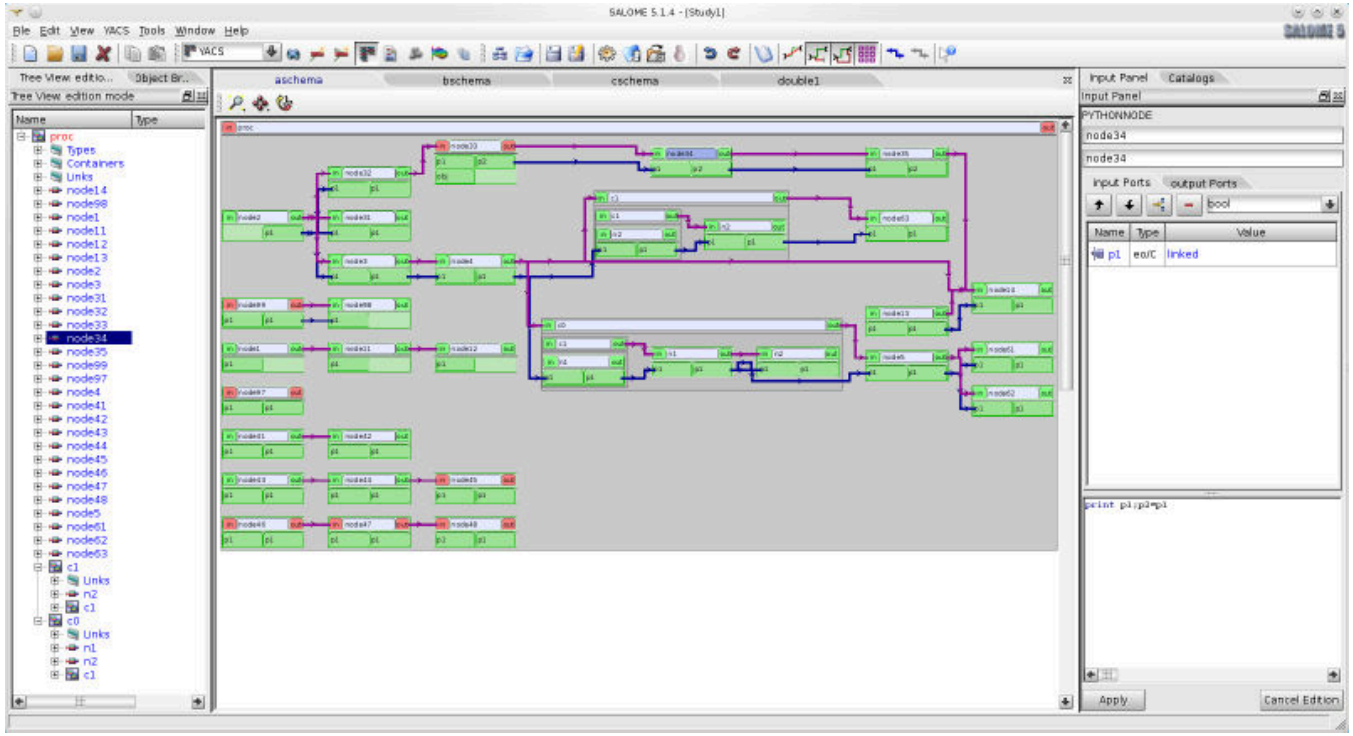
Présentation de SALOME

- Plate-forme logicielle d'intégration pour les codes de calcul
  - Fournit des modules de CAO, de maillage et de visualisation
  - Permet de réaliser l'ensemble d'une étude au sein d'une même plate-forme
  - Permet la standardisation et l'interopérabilité des codes de calcul (modèle d'échange de données)
  - Plate-forme distribuée (basée sur CORBA)
  - Open Source (LGPL)
  - Partenariat EDF, CEA, OpenCascade
  - http://www.salome-platform.org



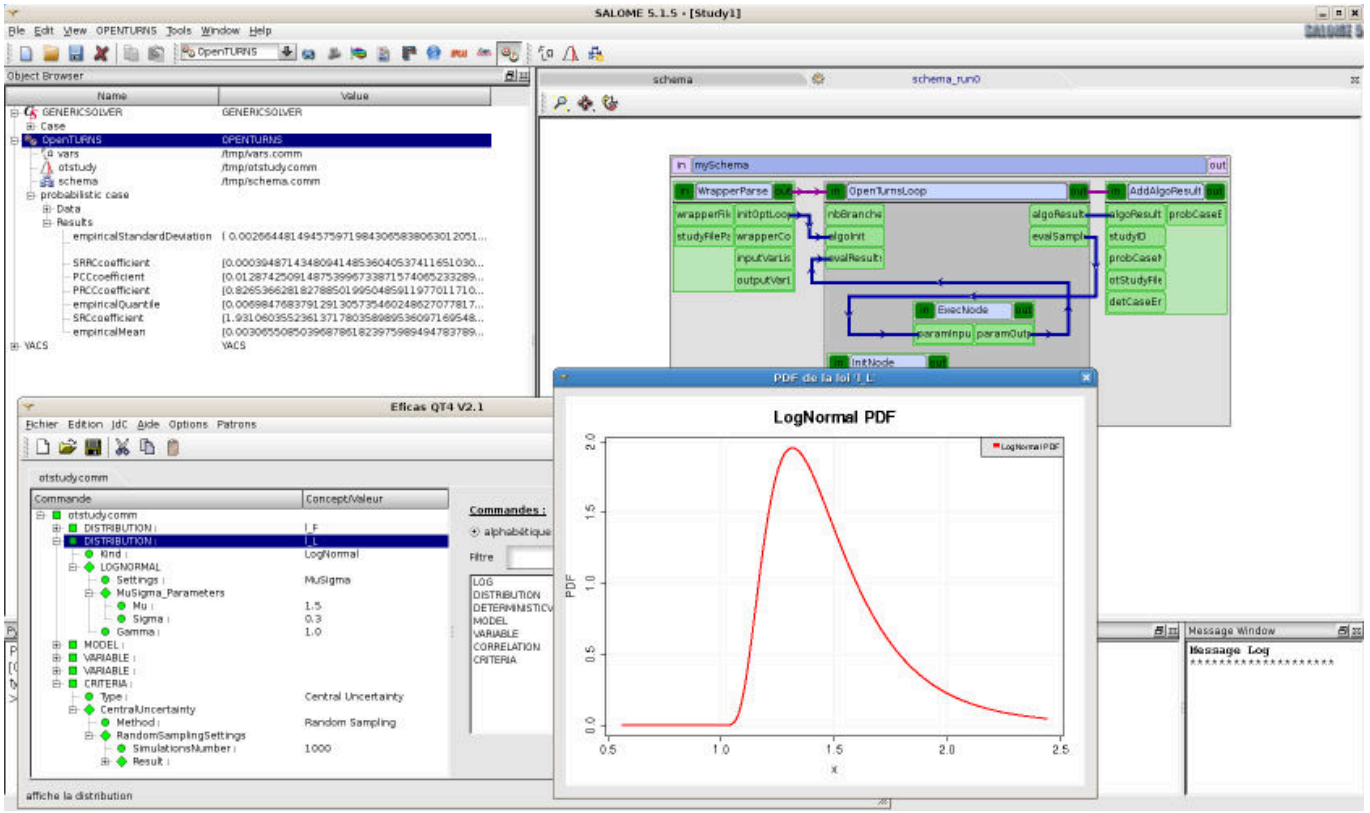
Présentation de YACS

- Module de supervision des calculs de SALOME
  - Interface graphique pour définir des enchaînements ou des couplages de composants de calcul
  - Superviseur pour l'exécution des schémas de calcul
  - Possibilité d'exécuter les composants sur des machines différentes



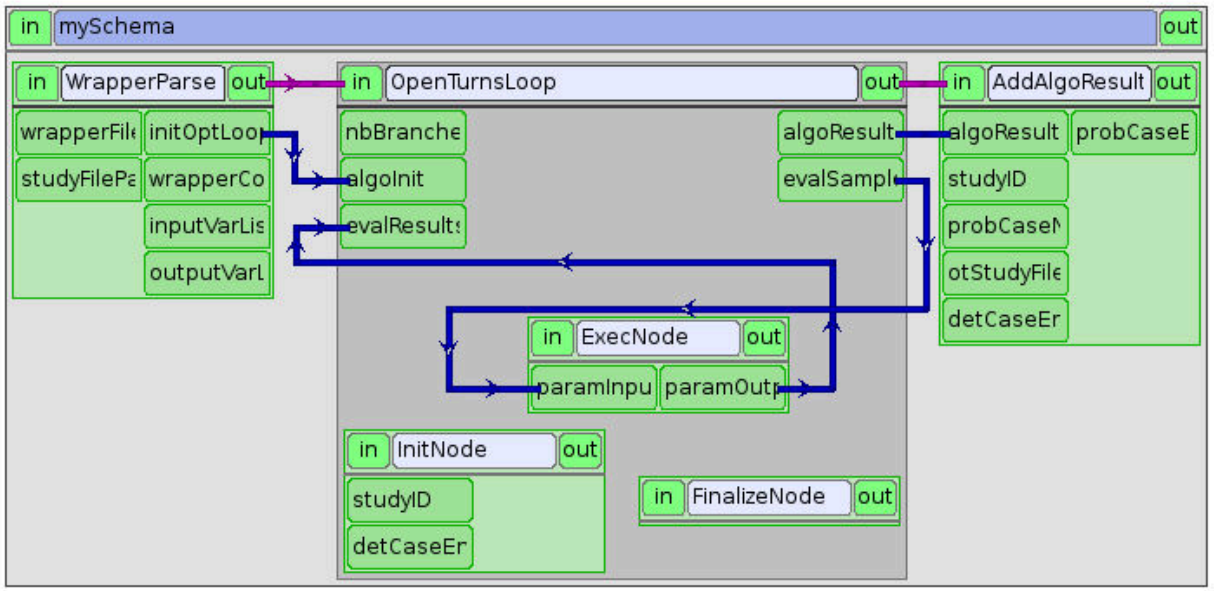
Utilisation du module OpenTURNS de SALOME

- Interface graphique pour la mise en données de l'étude probabiliste (outil Eficas)
- Outil d'aide à la création de schémas YACS pour OpenTURNS



Intégration des codes de calcul

- Création d'un composant SALOME au lieu d'un wrapper OpenTURNS
- Principale obligation : Pouvoir effectuer des traitements paramétriques
- SALOME fournit un outil (YACSGEN) pour simplifier la génération de composants
- Possibilité de définir un solveur sous la forme d'un script Python



Distribution et évolutions envisagées

- Distribution de la plate-forme SALOME + OpenTURNS
  - OpenTURNS est intégré avec YACS depuis SALOME 5.1.5 (Décembre 2010)
  - Licence LGPL pour l'ensemble (SALOME + OpenTURNS + module OPENTURNS)
  - Plate-forme SALOME intégrant OpenTURNS non distribuée sur le site http://www.salome-platform.org
  - Mais distribution via Salome-Meca (http://www.code-aster.org/astersalome)
- Améliorations envisagées
  - Meilleur suivi en cours d'exécution
  - Possibilité de reprise après erreur, tolérance aux pannes
  - Meilleure intégration avec les gestionnaires de batchs
    - Dimensionnement des jobs en fonction de la charge du cluster
    - Adaptation dynamique en fonction de la charge du cluster

Sommaire

- Présentation du logiciel OpenTURNS
- Problématiques du Calcul Haute Performance avec OpenTURNS
- Solutions pour le Calcul Haute Performance avec OpenTURNS
- Exemples d'application

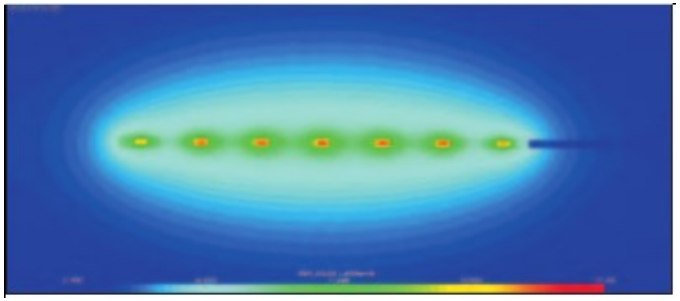
Évacuation thermique de puissance résiduelle

- Calcul d'incertitudes et de hiérarchisation (avec 4 paramètres incertains) sur l'évacuation thermique de puissance résiduelle par le puits de cuve dans un réacteur GenIV RNR-Na

- Solveur : SYRTHES (code de thermique conduction/rayonnement)
- Calcul élémentaire : Maillage de 10500 mailles. Convergence en environ 20 secondes sur une station de travail (sur un seul processeur, le cas élémentaire étant très petit)
- Étude OpenTURNS : 5000 à 10000 calculs élémentaires ➔ quelques dizaines d'heures CPU
- Utilisation de SALOME pour la distribution des calculs

Stockage de déchets HA/VL

- Propagation d'incertitudes pour les calculs de dimensionnement thermique du stockage des déchets de haute-activité et à vie longue



Champ de température après plusieurs années en stockage géologique

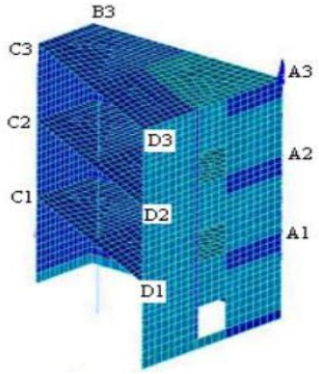
- Solveur : Code Syrthes (code de thermique)

- Calcul élémentaire : 10 minutes sur 8 processeurs sur une station de travail (parallélisation en mémoire partagée)
- Calcul OpenTURNS : 6000 calculs élémentaires ➔ environ 8000 heures CPU (jobs de 512 calculs élémentaires sur 32 nœuds)
- Utilisation de SALOME pour la distribution des calculs

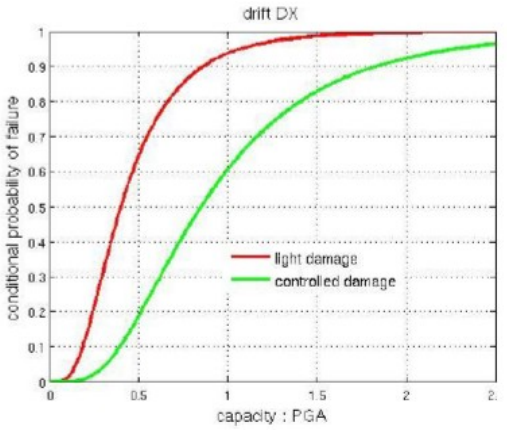
Résistance aux séismes avec Salome-Meca : Benchmark SMART

- Courbes de fragilité dans le cadre du benchmark SMART 2008 (CEA-EDF)
- Solveur EF : Code_Aster (mécanique des structures)
- Variable d'intérêt : déplacement différentiel entre deux étages
  - Deux niveaux de dommage considérés (3mm et 6mm)
- Modélisation des incertitudes
  - 50 accélérogrammes
  - 3 paramètres de modèle incertains

Maquette béton du benchmark SMART



Maillage utilisé par Code_Aster



Courbes de fragilité

Résistance aux séismes avec Salome-Meca : Application réelle

- 2012 : Réutilisation de la méthodologie définie dans le cadre du benchmark SMART pour une structure industrielle réelle
- Temps du calcul de dynamique élémentaire : 50 à 200h sur une station de travail
- Étude OpenTURNS : 186 calculs Aster avec différents paramètres et accélérogrammes
- Calculs répartis sur 35 processeurs pour un total de 425 heures (environ 18 jours) ➔ environ 15000h CPU
- Utilisation de SALOME pour la distribution des calculs

Conclusion

- OpenTURNS est une cible idéale pour le HPC (grand nombre de calculs indépendants)

- Deux solutions maintenues pour distribuer des calculs avec OpenTURNS
  - Fonction Python distribuée incluse en standard dans OpenTURNS
  - Intégration avec SALOME pour disposer d'une plate-forme de simulation avec davantage de fonctionnalités