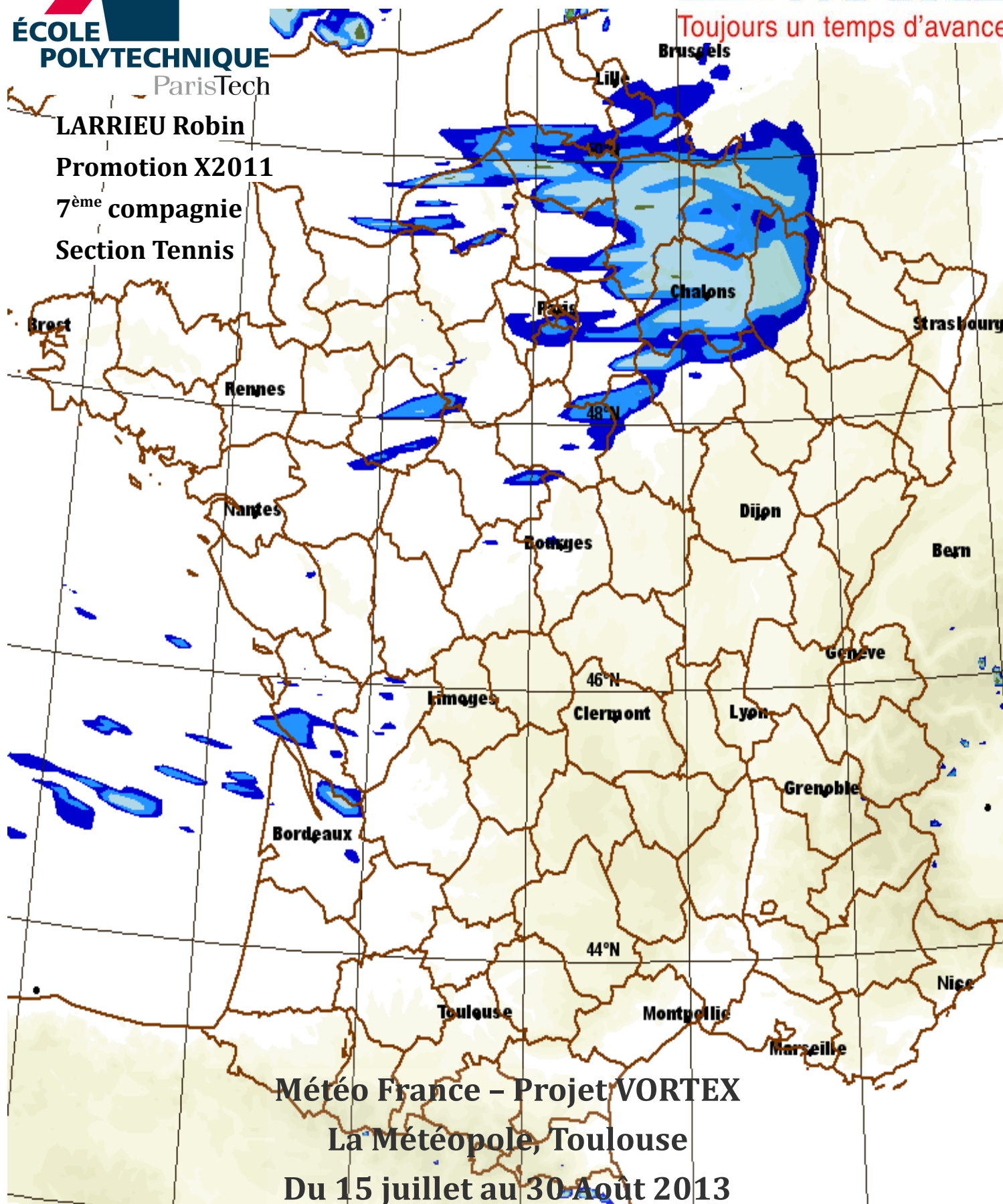


LARRIEU Robin

Promotion X2011

7^{ème} compagnie

Section Tennis



Météo France – Projet VORTEX

La Météopole, Toulouse

Du 15 juillet au 30 Août 2013

Remerciements

Je tiens à remercier particulièrement monsieur Eric Sevault du GMAP¹, chef du projet VORTEX et tuteur de mon stage, pour m'avoir fait cette proposition de stage et m'avoir apporté la formation et l'accompagnement nécessaires à son bon déroulement.

Je tiens également à remercier :

- Monsieur Guillaume Beffrey et madame Véronique Mathiot de COMPAS/GCO², pour m'avoir aidé dans la prise en main du projet et pour m'avoir expliqué les notions météorologiques dont j'avais besoin.
- Monsieur Pascal Lamboley de COMPAS/Support Informatique, pour m'avoir apporté une autre vision du projet tout en le découvrant en même temps que moi, et pour m'avoir fourni certains outils informatiques qui m'ont été très utiles pendant le stage.
- Plus généralement, tous les membres des équipes avec lesquelles j'ai été amené à travailler pour le bon accueil qui m'a été réservé.

1. Groupe pour la Modélisation et l'Assimilation en Prévision numérique

2. COntôle et Monitoring, Prévisibilité, Adaptation, Statistique/Gestion des Cycles Opérationnels

Table des matières

Remerciements	2
Introduction	4
1 Le contexte du stage	5
1.1 Le fonctionnement d'un modèle de prévision	5
1.2 Chronologie du projet	6
2 Le projet VORTEX	7
2.1 Idée directrice	7
2.2 Intérêt du projet	8
2.3 Application au modèle AROME	9
3 Réalisations au cours du stage	10
3.1 Adaptation d'une configuration OLIVE	10
3.2 Développement de la boîte à outils VORTEX	10
3.3 Écriture de scripts quasi-opérationnels	11
4 Enseignements tirés	12
4.1 Enseignements techniques	12
4.2 Enseignements personnels	13
Conclusion	15

Introduction

Ce rapport expose le déroulement du stage effectué au sein de Météo France pendant les mois de Juillet et Août 2013, dans le cadre de la deuxième année du cycle ingénieur de l'Ecole Polytechnique. Plus précisément, le stage s'est déroulé au *GMAP*³ sur le projet VORTEX (**Versatile Objects Rounded up in a Toolbox for Environmental eXperiments**) ; il consiste entre autres en une réécriture progressive des scripts de la chaîne de traitement opérationnel pour la prévision numérique du temps. En vue de son adaptation effective en opérations, ce projet est en outre mené en étroite collaboration avec certains membres de *COMPAS*⁴, notamment au sein de l'équipe *GCO*⁵.

Afin de mener à bien la réécriture envisagée, le projet VORTEX a conduit au développement d'une boîte à outils utilisant le langage Python. Il est caractérisé par une syntaxe claire, permettant que la nouvelle version de scripts soit d'une écriture plus simple qu'actuellement. Le stage s'est déroulé selon trois composantes :

- La mise en place d'une configuration OLIVE (du nom du logiciel de génération automatique de scripts utilisé par les équipes de recherche) pour un modèle de prévision numérique particulier, en utilisant la boîte à outils VORTEX existante.
- Le développement de nouvelles fonctionnalités pour cette boîte à outils, en particulier lorsque la mise en place de la configuration du modèle étudié le nécessitait.
- Le scriptage ad-hoc (tel qu'il est réalisé en opérations) utilisant la boîte à outils VORTEX des tâches nécessaires au fonctionnement du modèle étudié. Ceci doit permettre de montrer l'adéquation du projet avec le contexte opérationnel, mais aussi le gain apporté en termes de simplicité et de clarté à l'écriture du script, ainsi que de lisibilité de la trace d'exécution.

3. Groupe pour la Modélisation et l'Assimilation en Prévision numérique

4. COntôle et Monitoring, Prévisibilité, Adaptation, Statistique

5. Gestion des Cycles Opérationnels

1 Le contexte du stage

1.1 Le fonctionnement d'un modèle de prévision

Afin de bien comprendre les enjeux d'un tel projet, il semble important de présenter rapidement le cheminement nécessaire à une prévision météorologique.

Plusieurs modèles de calcul à différentes échelles sont lancés régulièrement (toutes les 3 à 6 heures suivant le modèle) et permettent d'obtenir une estimation de l'état de l'atmosphère à plus ou moins longue échéance. Les deux principaux modèles utilisés à Météo France sont *ARPEGE* (*Action de Recherche Petite Echelle Grande Echelle*) à l'échelle du globe et *AROME* (*Applications of Research to Operations at MEso-scale*) avec une résolution beaucoup plus fine qui s'exécute à l'échelle de la France métropolitaine ou d'un département d'Outre-mer.

Cet état de l'atmosphère (exemple en figure 1) est ensuite analysé et interprété par les prévisionnistes pour pouvoir informer le grand public du temps à venir et, le cas échéant, avertir les autorités de phénomènes potentiellement dangereux.

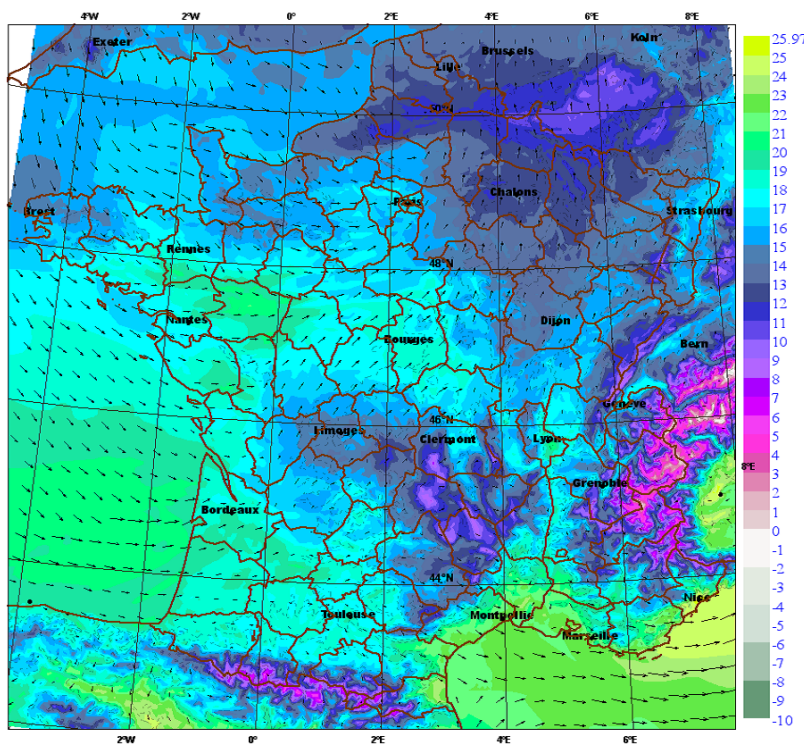


FIGURE 1 – Une sortie du modèle : vent et température.

La phase de calcul présente elle-même deux aspects :

- Le modèle numérique proprement dit, qui traduit les principes physiques régissant la dynamique de l'atmosphère sous forme d'un programme exécutable informatiquement. Ce programme gère également les optimisations (parallélisation du calcul, options de compilation ...) spécifiques à la machine sur laquelle il sera lancé.
- Un certain nombre de tâches annexes gérant l'import des ressources météorologiques telles que les données d'observation ou les informations géographiques, ainsi que le traitement des fichiers de sortie. Ces tâches annexes forment ce qu'on appelle une configuration et constituent l'essentiel des scripts enrobant le modèle numérique. Ce sont ces derniers qui font l'objet du stage.

1.2 Chronologie du projet

Les bases du projet ont été posées fin 2009 par un groupe de réflexion composé de membres des équipes CNRM⁶/GMAP/ALGO, DPREVI⁷/COMPAS/GCO et DSI/OP/IGA⁸ ; l'objectif étant de proposer des outils communs aux contextes opérationnels et de recherche, facilitant les transferts d'applications, leur suivi et leur validation. Au cours de ces réunions, un certain nombre de spécifications sont définies, ainsi qu'un cahier des charges et une liste de besoins fonctionnels dessinant les grandes orientations du projet.

S'ensuit une longue phase de développement menée en grande partie par le chef de projet E. Sevault, mettant en place la plupart des fonctionnalités actuelles de la boîte à outils VORTEX. En parallèle, une mise à jour est appliquée au logiciel OLIVE (utilisé par le centre de recherche pour gérer ses expériences), afin que les scripts qu'il génère puissent (au choix de l'utilisateur) être réécrits dans le nouveau langage de programmation, tout en utilisant les outils mis en place.

Ceci a permis courant 2013 de faire tourner sous OLIVE les premières configurations de modèle entièrement basées sur VORTEX. À la fin du stage, des configurations pleinement fonctionnelles existent pour la prévision ARPEGE (mise en place par E. Sevault) ainsi que pour l'essentiel des tâches de couplage et de prévision AROME (dont la mise en place était un des objectifs du stage).

Du fait des changements importants qu'il implique quant à la gestion des ressources, le projet ne peut être mis en application à une échelle plus générale qu'à

6. Centre National de Recherches Météorologiques

7. Direction de la prévision

8. Direction des Systèmes d'Information/OPérations/Intégration et Gestion des Applications

l'occasion d'une transition majeure de la chaîne opérationnelle. Il a été envisagé de le faire au moment du portage vers le nouveau calculateur, livré au printemps 2013 et dont le transfert de la chaîne était en cours au moment du stage, mais l'état d'avancement du projet ne l'a pas permis.

En 2014, une version haute résolution du modèle AROME (rendue possible par les performances de calcul accrues du nouveau calculateur) doit être développée au centre de recherche et il est envisagé que cette nouvelle configuration utilise VORTEX. L'application du projet en opérations se ferait alors au moment de l'intégration du modèle AROME haute résolution dans la chaîne de production.

2 Le projet VORTEX

2.1 Idée directrice

L'innovation principale du projet VORTEX consiste en une gestion complètement revue des ressources météorologiques. Actuellement dans les scripts opérationnels, ces ressources sont caractérisées uniquement par un nom de fichier et de répertoire, définis selon une nomenclature plus ou moins standardisée et explicite.

Au lieu de cela, on préfère adopter une approche descriptive grâce à des structures de données et des méthodes de résolution ingénieuses. En effet, un langage de haut niveau comme Python permet le développement d'outils complexes pour gérer ces descriptions.

Pour simplifier, on utilise une reconnaissance d'empreintes : sachant que chaque ressource a ses propres caractéristiques (un type, une date, un modèle qui l'a produite...), on regarde quelle ressource correspond au mieux à la description fournie ; c'est à dire qu'aucun attribut obligatoire ne manque à la description, et que les différents critères correspondent à des valeurs autorisées pour la ressource (s'il y a plusieurs candidats pour une description donnée, on se base sur le nombre de critères qui correspondent, et sur des notions de priorité).

La lecture et l'écriture sur le disque sont gérées par d'autres objets disposant d'une fabrique de noms, permettant de manière transparente (par un simple appel à une méthode `get()` ou `put()`) de récupérer la ressource ou, dans le cas d'une ressource produite, de la stocker en vue d'une utilisation future.

En plus d'être plus transparente, la récupération des ressources par ce type d'objet est revue dans certaines de ses modalités. Certains types de ressources comme les données de modèle ou les données climatologiques (qui se trouvent être les plus volumineuses) n'ont pas besoin d'être modifiées lors de l'expérience ; un lien vers

le fichier est donc suffisant. En ne copiant le contenu des données que lorsque c'est nécessaire (modification d'un fichier de paramètres, décompression d'une archive ...), on obtient un gain sensible en termes de performances : les ressources, représentant plusieurs giga-octets de données, peuvent être "récupérées" en moins d'une seconde alors que leur copie en aurait nécessité une dizaine.

2.2 Intérêt du projet

En découplant complètement la définition de la ressource de l'interaction avec le fichier correspondant, il devient possible de dire au programme *"je cherche un état du modèle issu de la prévision ARPEGE d'hier à 12h"*, plutôt que *"je veux le fichier ICMSHARPE+0000 de tel répertoire"*. On pourra comparer les scripts présentés sur la figure 2.



The figure consists of two panels. The top panel shows a VORTEX initialization script in a dark-themed code editor. The bottom panel shows a shell script for the same task in a light-themed terminal window.

```

init = toolbox.input(
    role='Initial Conditions',
    format='fa',
    local='CPLIN[geometry::area]+[term::fmthm] ',
    date='2013082612',
    geometry='globalsp',
    kind='historic',
    model='arpege',
    term=0,
    provider=init_provider
)
  
```

```

rm -f ICMSH${CNMEXP}INIT
cp -p ${DATA_INPUT_PATH}/ICMSHARPE+${EC02}.r${RESPREV0} ICMSH${CNMEXP}INIT
  
```

FIGURE 2 – En haut : l'import d'une ressource en utilisant VORTEX ;
En bas : l'instruction équivalente dans les scripts actuels.

S'il est vrai que le code à produire pour gérer les ressources voulues s'en trouve plus long, surtout du point de vue de quelqu'un qui sait où se trouve le fichier dont il a besoin, son écriture est par contre bien plus intuitive. De plus, cet inconvénient est atténué par la possibilité de définir des valeurs par défaut concernant les caractéristiques partagées par une majorité de ressources nécessaires à l'expérience. Ceci faisant, il est rarement nécessaire de fournir plus de 4 ou 5 attributs pour achever la description d'une ressource.

Par ailleurs, et il s'agissait de la motivation initiale du projet, cela simplifie considérablement la tâche de l'équipe de contrôle des opérations dont le but est de vérifier

la conformité du code utilisé avec la configuration développée au centre de recherche.

Un autre intérêt majeur est l'introduction d'une certaine verbosité directement intégrée dans la boîte à outils, ce qui signifie que même si l'utilisateur n'a mis aucune instruction dans son script pour envoyer des messages d'information, les fonctions qu'il sera amené à utiliser renseigneront d'elles-mêmes de manière lisible sur ce qui a été fait. Ceci résulte en un rapport d'exécution à la fois plus clair et plus informatif, permettant d'identifier plus rapidement l'origine d'un problème.

2.3 Application au modèle AROME

On va maintenant illustrer les apports de ce projet avec l'exemple du modèle AROME qui a été plus spécifiquement étudié pendant le stage.

Le modèle AROME est un modèle dit à aire limitée, c'est à dire qu'il simule l'atmosphère sur une zone restreinte, typiquement la France métropolitaine et ses proches environs. Il nécessite donc d'être couplé au modèle global (ARPEGE) pour définir des conditions aux limites.

Une première phase de couplage récupère ainsi les données de prévision ARPEGE et fournit après de nombreux calculs des fichiers de couplage. Une deuxième phase d'analyse de surface évalue les échanges sol-atmosphère à partir des prévisions du modèle global et des informations géographiques. À partir des résultats de ces deux phases, la prévision peut être effectuée et produire les fichiers décrivant l'évolution de l'atmosphère.

L'utilisation de VORTEX permet ainsi d'adopter une approche descriptive (comme on le fait dans le langage parlé : comparer au paragraphe précédent) pour déclarer les entrées-sorties de ces trois phases. Par exemple, les ressources produites par la phase de couplage sont identifiées comme des conditions aux limites pour le modèle AROME, produites à partir d'ARPEGE, à des heures données. La boîte à outils les stocke alors sur le disque conformément à la fabrique de noms et est donc capable de les retrouver lorsque la phase de prévision est lancée et demande des données dont la description correspond.

3 Réalisations au cours du stage

Plusieurs nouvelles applications ont été développées pendant le stage. Comme précisé dans l'introduction, ces réalisations sont de diverses catégories.

3.1 Adaptation d'une configuration OLIVE

En s'inspirant des expériences existantes écrites en Korn Shell (le langage de programmation déjà utilisé), une configuration OLIVE pour le modèle AROME a été adaptée sous VORTEX. OLIVE est le logiciel utilisé par le centre de recherche pour gérer ses expériences et générer les scripts pour les lancer. De par son fonctionnement (il dispose également d'une boîte à outils pour une approche descriptive des ressources), il se prête particulièrement à la migration vers VORTEX.

Il s'agissait de trouver une description convenable des ressources compte tenu de ce qui était défini dans la boîte à outils pour que celles-ci soient correctement reconnues. Il a également été nécessaire de modifier certains paramètres généraux qui n'étaient pas correctement mis à jour lors de la création de l'expérience VORTEX.

Toutes les modifications effectuées devaient être répertoriées pour que les administrateurs d'OLIVE puissent identifier et corriger les erreurs à la création d'une expérience, de sorte que les configurations générées automatiquement soient fonctionnelles et que l'utilisateur n'aie plus à modifier que des critères spécifiques comme la date de la « prévision » qu'il souhaite effectuer.

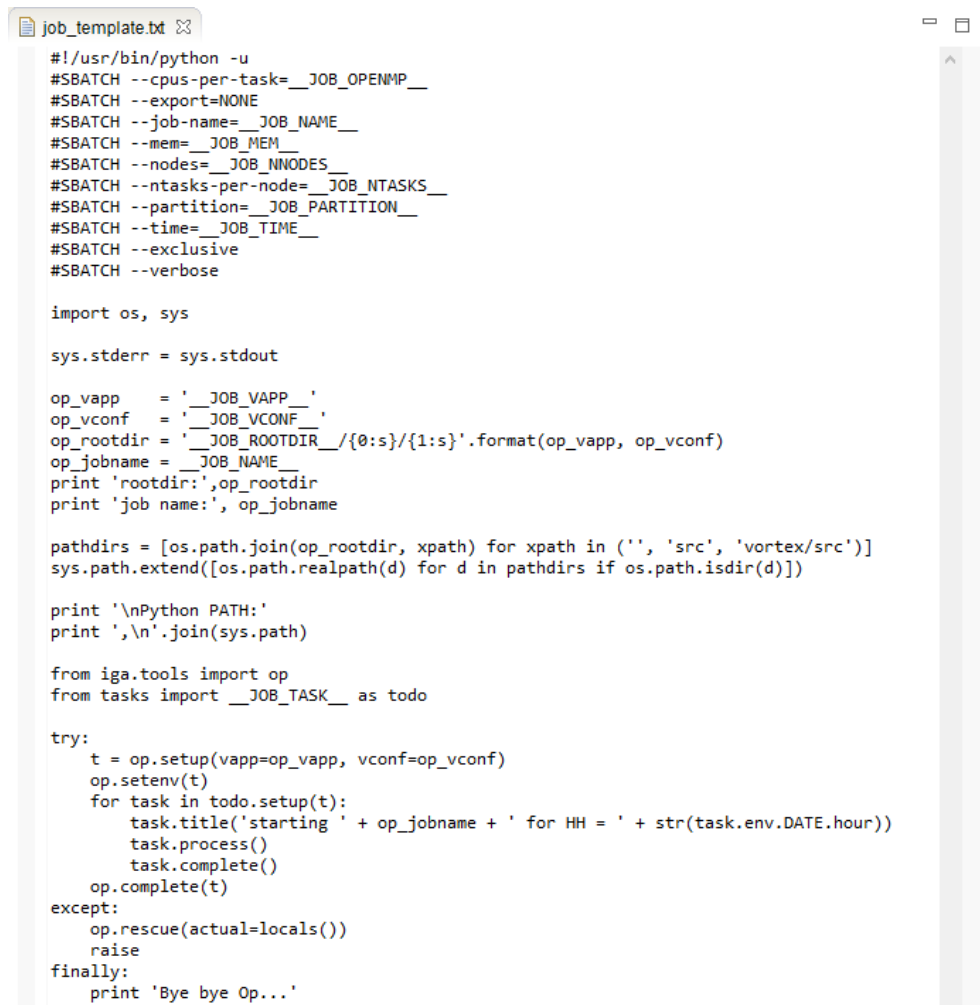
3.2 Développement de la boîte à outils VORTEX

Certaines fonctionnalités de la boîte à outils ont été revues ou ajoutées. D'une part, le déploiement de la nouvelle configuration a fait apparaître des bogues sur des cas qui ne s'étaient pas présentés auparavant et des modifications ont été apportées pour corriger ces comportements.

D'autre part, de nouvelles fonctions ont été définies, notamment pour améliorer la trace d'exécution. Par exemple, lorsque aucune ressource définie dans VORTEX ne correspond à la description, la boîte à outils envoie un message expliquant pour quelle raison telle ou telle ressource n'a pas pu être choisie ; des outils ont alors été créés pour regrouper les messages selon la raison pour laquelle la ressource ne correspondait pas.

3.3 Écriture de scripts quasi-opérationnels

Des scripts se voulant dans une optique pseudo-opérationnelle ont été écrits pour les trois étapes du modèle AROME (voir figure 3). Il s'agissait de reprendre ceux fournis par la configuration OLIVE et de les adapter dans leur forme et leurs fonctions (autant que le temps réduit le permettait) aux besoins des services de production.



```
job_template.txt
#!/usr/bin/python -u
#SBATCH --cpus-per-task=__JOB_OPENMP__
#SBATCH --export=NONE
#SBATCH --job-name=__JOB_NAME__
#SBATCH --mem=__JOB_MEM__
#SBATCH --nodes=__JOB_NNODES__
#SBATCH --ntasks-per-node=__JOB_NTASKS__
#SBATCH --partition=__JOB_PARTITION__
#SBATCH --time=__JOB_TIME__
#SBATCH --exclusive
#SBATCH --verbose

import os, sys

sys.stderr = sys.stdout

op_vapp = '__JOB_VAPP__'
op_vconf = '__JOB_VCONF__'
op_rootdir = '__JOB_ROOTDIR__/{0:s}/{1:s}'.format(op_vapp, op_vconf)
op_jobname = __JOB_NAME__
print 'rootdir:', op_rootdir
print 'job name:', op_jobname

pathdirs = [os.path.join(op_rootdir, xpath) for xpath in ('', 'src', 'vortex/src')]
sys.path.extend([os.path.realpath(d) for d in pathdirs if os.path.isdir(d)])

print '\nPython PATH:'
print ',\n'.join(sys.path)

from iga.tools import op
from tasks import __JOB_TASK__ as todo

try:
    t = op.setup(vapp=op_vapp, vconf=op_vconf)
    op.setenv(t)
    for task in todo.setup(t):
        task.title('starting ' + op_jobname + ' for HH = ' + str(task.env.DATE.hour))
        task.process()
        task.complete()
    op.complete(t)
except:
    op.rescue(actual=locals())
    raise
finally:
    print 'Bye bye Op...'
```

FIGURE 3 – Un gabarit de script opérationnel.

À l'issue du stage, ces scripts sont parfaitement fonctionnels et forment un exemple pour montrer l'applicabilité du projet. Cependant, du fait du peu de temps dispo-

nible, il manque encore certaines fonctionnalités pour les rendre utilisables dans un contexte opérationnel.

Par exemple, aucun mode secours n'a été développé pour gérer les procédures automatiques dans le cas d'une ressource manquante. De même, la récupération des ressources au fur et à mesure qu'elles sont produites n'est pas encore possible. Or, avec des contraintes de temps réel, on préférerait lancer les deux modèles simultanément et ne récupérer les ressources que lorsqu'on en a besoin (quitte à attendre en cours d'exécution) plutôt qu'attendre que le premier modèle ait terminé avant de lancer le second.

4 Enseignements tirés

4.1 Enseignements techniques

Le stage s'est avéré très riche sur le plan technique car il a fallu mettre en oeuvre des compétences à la fois approfondies et variées.

Premièrement, ce stage était l'occasion de travailler sur un ordinateur hautes performances (photo en figure 4) et de se familiariser avec les problématiques spécifiques à ce genre de machine. On citera comme exemple tout ce qui relève de la soumission des tâches et de l'allocation des cibles d'exécution : l'efficacité d'un supercalculateur tient à sa capacité d'exécuter un grand nombre de programmes simultanément. Pour éviter que ceux-ci n'interfèrent, il faut donc demander pour chaque programme un certain nombre de processeurs, une certaine quantité de mémoire etc ; et le programme lui-même doit être conçu pour fonctionner avec ces paramètres.

De plus, le développement en Python qui était au cœur du stage a permis l'appropriation et le perfectionnement de nouvelles méthodes de programmation. Il a par exemple fallu adapter sa vision d'un problème à la volonté de généricité inhérente au projet. Ainsi, pour la fonction servant à regrouper les messages d'erreur par catégorie (évoquée en 3.2), il a été reproché à la première version d'être trop étroitement liée au contexte dans lequel elle devait être utilisée, et cette approche a dû être revue pour l'adapter à n'importe quel type de message d'erreur et n'importe quels critères de tri.

Enfin, le stage a mis en avant l'intérêt de la formation pluridisciplinaire dispensée à l'École Polytechnique. En effet, outre la forte dimension informatique du stage, il restait en arrière plan la notion de modèle numérique qu'il était important d'appréhender lorsqu'on le manipulait. En cela, les connaissances vues en mathématiques appliquées concernant les méthodes de résolution numérique se sont avérées particulièrement utiles.



FIGURE 4 – Le supercalculateur BULL de Météo France

4.2 Enseignements personnels

De par la nature du projet, projet de développement impliquant un petit nombre de personnes, sur une durée assez courte, qui plus est pendant la période estivale, le stage ne se prêtait pas autant que l'École l'aurait souhaité à l'observation des relations hiérarchiques et managériales. Il s'est cependant révélé très enrichissant en ce qui concerne les méthodes de travail en équipe. Le nombre réduit de personnes impliquées dans le développement du projet permettait un ajustement permanent et réactif. Ainsi, des discussions ont parfois eu lieu pour déterminer la meilleure façon d'aborder un problème, ou le comportement souhaité pour un programme dans certains cas particuliers.

De plus, le stage s'est déroulé pendant la phase dite de vérification de service rendu

du nouveau calculateur, ce qui montre un autre aspect du travail collaboratif. Cette période de 30 jours permettait à la fois la prise en main de la nouvelle machine par les différentes équipes, les premiers tests d'exécution. Cette phase permet également la formulation de réclamations diverses pour des logiciels nécessitant d'être installés, mais aussi (tel est le but premier de cette période) de vérifier que la disponibilité et la fiabilité globales de la machine sont conformes aux termes du contrat signé entre Météo France et le constructeur. Comme on s'en doute, une telle période d'activité a donné lieu à de nombreuses communications inter-équipes concernant par exemple la manière dont chacun s'adaptait aux spécificités de la nouvelle machine.

Conclusion

Ce stage était donc l'occasion de participer, certes pour un temps limité, à un projet important, qui s'inscrit sur le long terme. Sa forte dimension technique a permis à la fois la mise en oeuvre de connaissances personnelles et l'acquisition de nouvelles compétences, tandis que le contexte général à Météo France mettait en évidence différents aspects du travail en équipe. Outre une participation directe au développement du projet, le travail effectué comprend la réalisation d'exemples d'utilisation qui, une fois complétés, pourraient jouer un rôle dans l'adoption éventuelle de VORTEX par la chaîne opérationnelle.

Le stage s'est ainsi avéré une expérience professionnelle très intéressante et me conforte dans mes choix d'orientation mêlant développement informatique et considérations scientifiques théoriques. J'attends donc beaucoup du stage de recherche de 3^e année pour explorer davantage le second aspect et confirmer cette vision.