Distribution des services de calcul pour la conception et la gestion optimale des bâtiments: perspectives des approches composants déployés via le " cloud computing "

Benoît Delinchant, Pierre-Yves Gibello, Franck Verdière, Frédéric Wurtz

Distribution des services de calcul pour la conception et la gestion optimale des bâtiments: perspectives des approches composants déployés via le « cloud computing »

B. Delinchant[1], P.Y. Gibello[2], F. Verdière [1-3], F. Wurtz [1]

[1] *G2ELAB, Laboratoire de génie électrique de Grenoble. ENSE3, 11 rue des Mathématiques, BP 46, 38402 Saint Martin d'Hères*
benoit.delinchant@G2Elab.grenoble-inp.fr, frederic.wurtz@G2Elab.grenoble-inp.fr

[2] *EBM Websourcing: 8 Place Robert Schuman, 38000 Grenoble*
pierre-yves.gibello@petalslink.com

[3] *Vesta-System: 22, avenue Doyen Louis Weil, 38000 Grenoble*
franck.verdiere@vesta-system.com

RÉSUMÉ. *La gestion optimale des bâtiment nécessite des modèles afin d'anticiper, sur un horizon plus ou moins lointain, le comportement du système bâtiment. Nous proposons, pour atteindre cet objectif, de mettre à disposition, sur le web, des services de calcul permettant la composition, la simulation et l'optimisation à distance des équipements constituant chaque bâtiment. Un changement de paradigme est alors nécessaire, afin d'outiller au mieux cette nouvelle voie. Le paradigme adapté à ces enjeux est celui des composants logiciels, associé aux technologies des web-services. Nous montrons que les modèles peuvent être embarqués dans des composants logiciels et mis à disposition sur le web sans effort particulier du modélisateur. Nous illustrons l'utilisation d'un web-service de calcul par la réalisation d'une optimisation à distance sans jamais disposer du modèle.*

ABSTRACT. *The optimal building management requires models to predict the behavior of building systems. We propose to achieve this goal, by making available on the web, remote computation services for the composition, simulation and optimization of equipments. A paradigm shift is necessary in order to fulfil this goal. The well suited paradigm is software components, associated with web-services technologies. We show that models can be embedded in software components and made available on the web without any special effort of the user. We illustrate the use of a web-service calculation by the realization of a remote optimization without ever holding the model.*

MOTS-CLÉS : *composant logiciel, service web, gestion optimale.*

KEY WORDS: *software component, web service, optimal control.*

1. Enjeux énergétiques du bâtiment et perspectives ouvertes par le « cloud computing »

1.1 Objectifs bâtiment

Le bâtiment, avec 43% de la consommation d'énergie primaire, et 66% de la consommation d'énergie électrique, est un enjeu majeur pour les années à venir. La problématique visée ici est d'imaginer de nouvelles approches pour aider à améliorer à la fois la conception (simulation, optimisation, ...) et la supervision (gestion optimale anticipative et réactive, …) des bâtiments en profitant de l'accessibilité de plus en plus grande des réseaux informatiques dans les bâtiments et d'un phénomène majeur : le « cloud computing » ou informatique dans les nuages.

1.2 Réseau énergétique et réseau informatique

La figure suivante illustre un lien possible entre deux réseaux omniprésents, celui de l'information et celui de l'énergie. En effet, nous pouvons imaginer que tous les composants énergétiques liés au bâtiment soient reliés entre eux via deux canaux : énergétique et informatique. C'est cette architecture que nous pouvons nommer smart building. Ce lien peut être effectif ou virtualité. En effet, nous pouvons considérer que dans un avenir proche il soit utopique de rendre « intelligents » tous nos équipements domestiques ou de bureau. Par contre, de la même manière que des mesures effectuées par des capteurs sur ces équipements peuvent être disponibles sur des serveurs de données standard (OPC[1], …), il est tout à fait concevable que des modèles soient disponibles sur des serveurs, « virtualisant » le fonctionnement des équipements. Ces modèles seraient fournis par les fabricants d'équipements, et mis à disposition sur des serveurs de calcul pour offrir des solutions de conception et/ou de gestion « intelligentes ».

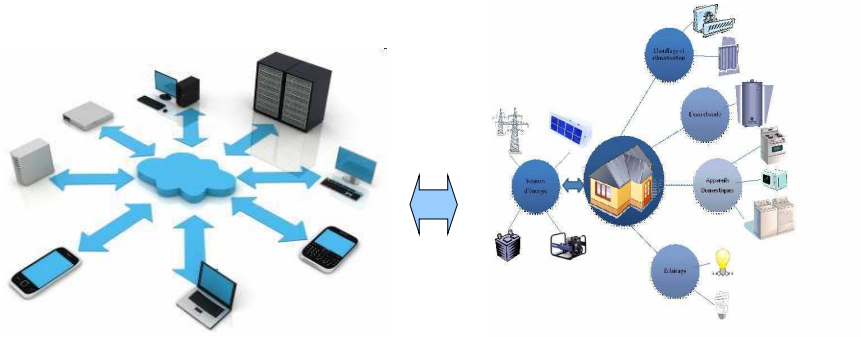


Figure 1. *La perspective du « cloud computing » au service de la simulation et de la supervision des bâtiments*

[1] OPC : standard d'interopérabilité de données entre capteurs/actionneurs et supervision d'une GTB

1.3 Cloud computing

On commencera ainsi à esquisser les perspectives et avantages qui s'offrent à nous avec des possibilités de mise à disposition de fonctionnalités de calcul, de simulation ou d'optimisation disponibles à distance dans des web-serveurs.

Au titre des avantages et perspectives, on peut commencer par rappeler ici :

- Le développement durable (*mutualisation des ressources de calcul sur des PC mutualisés et virtualisés pour une économie d'échelle énergétique*)
- Sûreté de fonctionnement (*services et stockage de données mutualisés sur infrastructures fiables*)
- Evolutivité et maintenance (*un service mutualisé peut évoluer sans redéploiement et déplacement local*)
- Disponibilité générale de services unitairement coûteux (*licences de calcul, prévision/optimisation avec calculs lourds*)
- Innovation facilitée (*assemblage « mashup » de services générant des usages nouveaux, lien avec les « smart grid », émergence d'acteurs nouveaux, usagers devenant fournisseurs de services*)
- Possibilités de disposer de capacités de stockage et de calcul (*au moins transitoirement*) non limitées.

2 Une architecture dédiée : des composants au WEB-Service

2.1. Nécessité d'un changement de paradigme

Cependant, ces perspectives ne pourront être atteintes qu'au prix d'un changement de paradigme sur la structure des applications de conception, de simulation et de supervision utilisées actuellement dans la filière. Ceci exige d'aller au delà du paradigme d'environnements, c'est-à-dire au-delà des logiciels mis en œuvre aujourd'hui dans les bureaux d'étude pour réaliser la conception d'un bâtiment, ou mis en œuvre in situ pour réaliser la gestion technique, et qui s'appuient sur des environnements informatiques non distribués et non interopérables. Par exemple, les modèles y sont généralement développés dans des outils logiciels spécifiques n'offrant parfois aucune solution d'interopérabilité dans un secteur où la complexité croissante nécessite une prise en compte globale du système.

C'est pour cela que nous proposons de nous appuyer sur un des derniers paradigmes de structure des architectes informatiques basée sur le concept de framework ouvert, lui-même basé sur le concept de composant logiciel autonome, capitalisable et conceptuellement dédié à l'interopérabilité. L'approche composant fait suite à l'approche orientée objet dans l'histoire des paradigmes de structuration des architectures logicielles en informatique [SZY 98]. C'est dans ce cadre qu'ont été développés les composants ICAR-MUSE étudiés dans les projets ANR SIMINTHEC et PLUME [GAA 11].

2.2. Avantages de l'approche à composants logiciels

Ces composants logiciels permettent de prendre en charge automatiquement des problématiques telles que la communication entre outils logiciels, l'hétérogénéité de langage de programmation et de système d'exploitation. Il permettent également la capitalisation et la mise à disponibilité « sur étagère » de modèles dans le cas de systèmes modélisés/simulés par composition de modèles. Ainsi, on peut noter que la norme ICAr, exploitée depuis une dizaine d'années dans le secteur du dimensionnement optimal de systèmes du génie électrique, offre aujourd'hui des solutions d'interopérabilité entre des outils ou langages tels que Modelica, Pleiade/Comfie, TRNSys, etc.

Une caractéristique importante de ces composants réside dans leur capacité d'autonomie. C'est-à-dire qu'ils peuvent être échangés entre partenaires, ou simplement mis à disposition en téléchargement sur des sites dédiés à la capitalisation/réutilisation de modèles tels que DIMOCODE. (http://www.dimocode.org)

L'étape suivante que nous proposons ici consiste à rendre les capacités de calcul d'un modèle, encapsulé dans un composant, directement exploitables à distance sur un web-serveur.

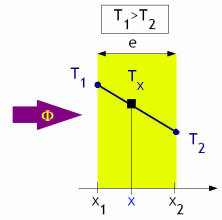
2.3. Mise à disposition sur Internet via des WEB-Services

La solution que nous avons mise en œuvre dans le cadre du projet ANR-SIMINTHEC respecte les préconisations techniques suivantes :

- Technologie basée sur un protocole usuel de l'Internet : HTTP, infrastructure REST (*Representational State Transfer*)
- Déployable sur un socle technique d'usage commun : Application Web Java (*serveur d'application Apache Tomcat*)
- Neutralité maximale vis-à-vis des protocoles, des technologies et des langages (*approche peu normative*) : encodage JSON

Des composants ainsi spécifiés sont faciles à développer, mettre en œuvre et assembler, et n'induisent aucune limitation de nature à brider l'innovation future.

Prenons l'exemple d'un modèle simple de conduction de la chaleur dans une paroi (Figure 2). La loi de Fourier appliquée à l'équation (1) nous donne l'expression de la densité de flux de chaleur (2) et donc du flux total (3). Nous retiendrons simplement l'équation finale (4) donnant la différence de température en fonction du flux et de la résistance thermique de la paroi.



$$T = T_1 + \frac{x - x_1}{e}(T_2 - T_1) \quad (1)$$

$$\varphi = -\lambda \frac{dT}{dx} = \frac{\lambda}{e}(T_1 - T_2) \quad (2)$$

$$\Phi = \frac{\lambda S}{e}(T_1 - T_2) \quad (3)$$

$$\Delta T = R \cdot \Phi \quad (4)$$

Figure 2. *Cas simple de conduction thermique d'une paroi*

Ce modèle peut être décrit simplement dans l'outil *CADES (www.cades-solutions.com)* [DEL 07] qui génère alors un composant logiciel *ICAr (Interface for Component Architecture)* [DEL 11]. Ce composant ICAr est également compatible *WAR (Web Application ARchive)* c'est-à-dire qu'il suffit de le télécharger (*via FTP : File Transfert Protocol*) dans le dossier d'un serveur d'application de type Tomcat (*tomcat.apache.org*) pour que le service de calcul soit déployé et accessible en web-service.

Nous avons également développé une interface web pour télécharger le composant sur le serveur, accessible par login, qui permet ainsi de gérer ses services :

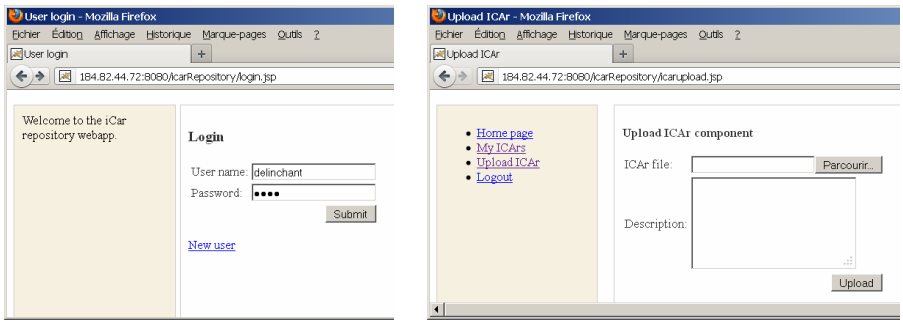


Figure 3. *Interface web pour rendre disponible un modèle en web-service*

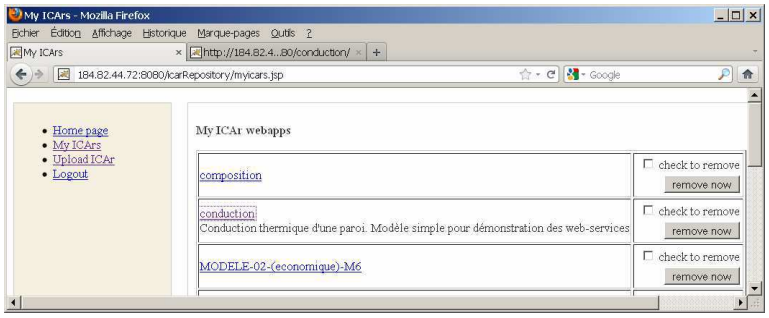


Figure 4. *Interface web pour gérer ses web-services*

Une fois le composant chargé, la liste des services disponibles et le détail des ports des différents services peuvent être affichés par une simple requête http (Figure 5).

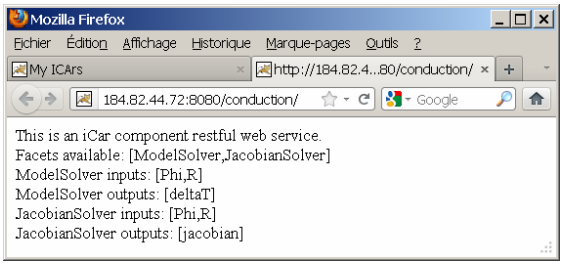


Figure 5. *Liste des services et ports du web-service « conduction »*

La requête est donc formulée en HTTP sur une URL, qui représente une ressource. La requête se compose de paramètres (*partie QUERY_STRING*), que la ressource devra alors interpréter. Ceci peut être réalisé simplement dans le navigateur web, mais surtout dans n'importe quel langage (*Python, Java, .net, …*) et de manière très simple. Par exemple, pour connaître les services disponibles sur le composant, il suffit d'invoquer la requête suivante :

```
Requête : http://184.82.44.72:8080/conduction/facets

Réponse : [ModelSolver,JacobianSolver]
```

Le composant et les facettes fournissent des méta-données de description, par exemple la liste de leurs ports en entrée et sortie. Ici, les ports d'entrée :

```
Requête : http://184.82.44.72:8080/conduction/facets/ModelSolver/inputnames

Réponse : [Phi,R]
```

Il sera possible d'invoquer ces facettes pour effectuer un calcul, par exemple (*calcul de u pour i=3 et r=2*) :

```
Requête : http://.../conduction/facets/ModelSolver/resolve?{R:3.0,Phi:2.0}
Réponse : {deltaT:6}
```

3 Cas d'application : dimensionnement et gestion optimale d'un système enveloppe + chauffage

Le cas d'application illustrant notre propos consiste à dimensionner un système de chauffage associé à une température de relance optimale (*minimisant l'énergie consommée*) afin de satisfaire un confort utilisateur (*consignes de température*). Pour cela nous disposons de deux modèles :

- Un modèle d'enveloppe nous permettant de calculer la température interne en fonction de la température externe et des sources.

- Un modèle de chauffage régulé, injectant une puissance de chauffe en fonction d'une consigne et de la température courante.

CADES est utilisé pour décrire le modèle composé, ceci sous la forme d'équations algébriques et d'algorithmes :

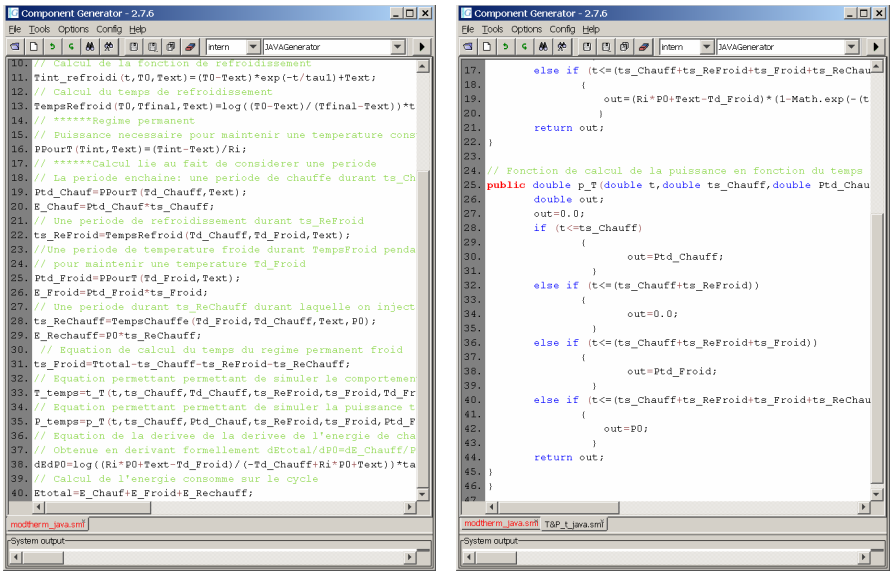


Figure 6. *Génération du composant ICAr à partir des équations (gauche) et algorithmes (droite) dans l'environnement CADES*

Le composant logiciel est alors automatiquement généré (*analyse du modèle, définition de l'incidence du graphe de calcul, génération de code Java, compilation, empaquetage*). Le fichier correspondant est déposé sur un serveur d'application en passant par un navigateur Internet. Il est automatiquement mis à disposition, prêt à être utilisé en tant que service web.

Ce web-service peut alors être connecté à un outil d'optimisation. Cette optimisation permet alors de dimensionner, pour un jeu de données météo et des caractéristiques de l'enveloppe (*paramètres R et C*), la puissance nominale du chauffage, associée à la commande optimale de relance satisfaisant les consignes de confort.

La Figure 7 illustre cette optimisation dans laquelle le modèle du système que l'on optimise est virtuellement présent et connecté à un algorithme d'optimisation.

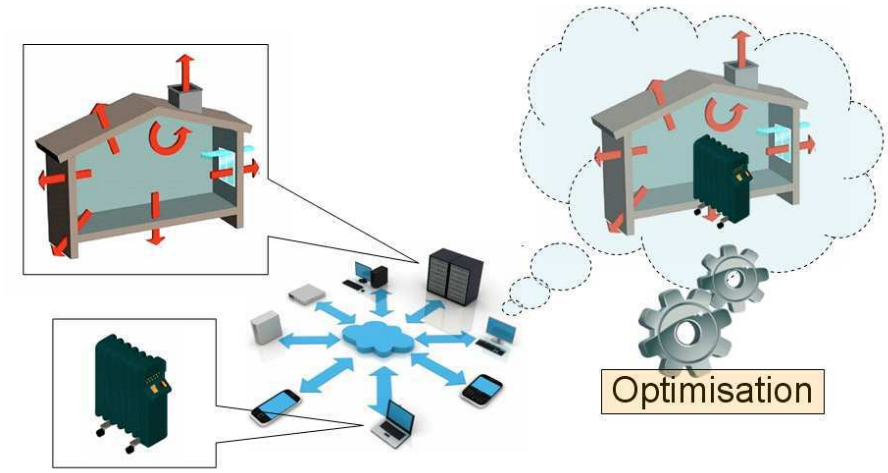


Figure 7. *Un exemple d'optimisation avec composants répartis et composés/orchestrés dans le « cloud »*

La Figure 8 illustre l'optimisation qui s'effectue en 10 itérations (*rapide grâce à la disponibilité du Jacobien du modèle*) et 3.7 secondes. Les résultats d'optimisation permettent de déterminer la puissance du chauffage (*P0=100W*), la température de relance optimale (Td_{froid}*=2°C*) permettant de minimiser l'énergie sur une journée (E_{total}*=647J*).

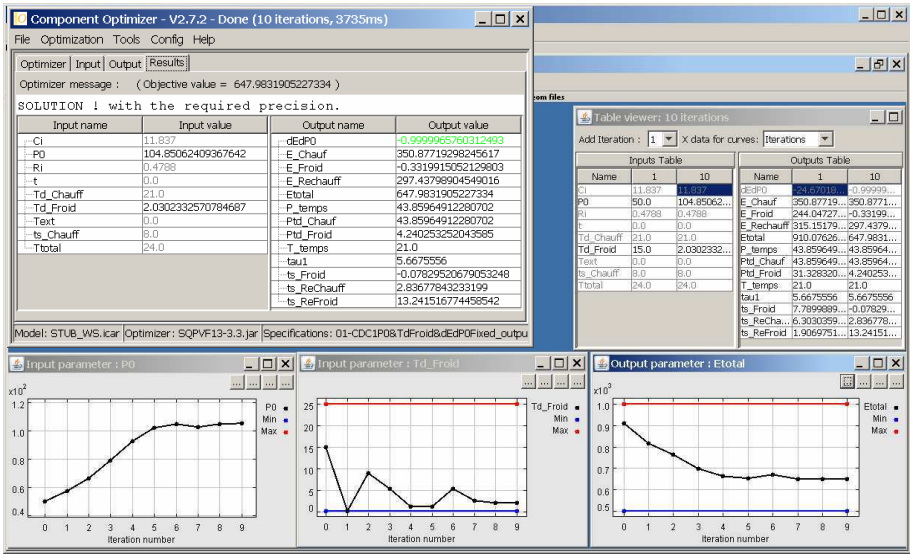


Figure 8. *Résultats d'optimisation du modèle disponible comme web-service*

4 Conclusions

En conclusion, nous proposons une nouvelle vision pour la conception et la gestion optimale des bâtiments. Dans cette vision, les modèles sont créés et mis à disposition sur des serveurs d'application au travers du web puis utilisés durant les phases de dimensionnement, mais également durant la gestion anticipative optimale embarquée dans les bâtiments.

5 Bibliographie

[DEL 04] DELINCHANT, WURTZ F., D. MAGOT, B., GERBAUD L., "A component-based framework for the composition of simulation software modeling electrical systems", *Journal of Simulation, Society for Modeling and Simulation International, Special Issue: Component-Based Modeling and Simulation*. Jul 2004; vol. 80: pp 347 - 356.

[DEL 07] DELINCHANT B., DURET D., ESTRABAUT L., GERBAUD L., NGUYEN HUU H. H., DU PELOUX B., RAKOTOARISON H.L., VERDIERE F., BERGEON S.; WURTZ F., "An Optimizer using the Software Component Paradigm for the Optimization of Engineering Systems", *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 26 No. 2, 2007, pp. 368-379

[DEL 11] DELINCHANT B., *La CAO et l'optimisation de systèmes, une approche par couplages dynamiques de composants*, Habilitation à diriger des recherches, Université Joseph Fourier, 2011.

[GAA 11] GAALOUL S., DELINCHANT B., WURTZ F., VERDIÈRE F., "Software components for dynamic building simulation", *IBPSA 2012 - 12th Conference of International Building Performance Simulation Association*, Sydney, Australie, 2011

[SZY 98] SZYPERSKY C., Component Software – Beyond Object-Oriented Programming, Addison-Wesley, 1998