

**SPECIFICATIONS DETAILLEES
DU LOGICIEL**

Delta Compression

Evolutions du document

DATE	REVISION	OBJET	Auteur
08/01/16	0	Création	S.D.

Diffusion papier contrôlée :

	CONCEPTEUR / REDACTEUR	APPROBATEUR
NOM	DUTERTRE Stéphane	
FONCTION	R.B.E.	

ATTENTION : sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.

nke ☎ (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..<http://www.nke.fr>

Modèle SPDL.dot Rev 1 du 08/06/98

	SPECIFICATIONS DETAILLEES DU LOGICIEL	SPDL_DeltaCompression.docx	
	Delta Compression	Rev : 0	Page 2/8

SOMMAIRE

1. OBJET DU DOCUMENT	3
2. DOCUMENTS APPLICABLES - DOCUMENTS DE REFERENCE	3
3. DEFINITIONS - TERMINOLOGIE	3
4. SPECIFICATIONS	3
4.1 PRINCIPE DE BASE	3
4.2 L'ARBRE DE PROBABILITES	3
4.2.1 <i>Détail de l'arbre des probabilités</i>	4
4.2.2 <i>Taille en bit de l'encodage</i>	5
4.2.3 <i>Stockage des bits</i>	6
4.2.4 <i>Les limites absolue de compression</i>	7
4.3 EXEMPLE DE COMPRESSION	7

ATTENTION : sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.

nke ☎ (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..<http://www.nke.fr>

Modèle SPDL.dot Rev 1 du 08/06/98

	SPECIFICATIONS DETAILLEES DU LOGICIEL	SPDL_DeltaCompression.docx	
	Delta Compression	Rev : 0	Page 3/8

1. OBJET DU DOCUMENT

Description du principe de fonctionnement de la compression delta. Cet algorithme permet à partir d'un buffer de données brutes de la convertir en un buffer de stockage de taille réduite. L'intérêt de cette algorithmique est surtout intéressante pour de la compression de données de mesure physique en faible, voire très faible, quantité. Elle est aussi pertinente pour des systèmes informatiques intégrant une faible puissance de calcul et de très faibles ressources de stockage.

2. DOCUMENTS APPLICABLES - DOCUMENTS DE REFERENCE

Spécifications Générales Ref :

3. DEFINITIONS - TERMINOLOGIE

Algorithme de compression sans perte d'information : Il s'agit d'un algorithme de compression/décompression qui après passage dans la procédure de compression puis après passage dans l'algorithme de décompression retrouve l'intégralité et l'ordre des données initiales.

Taux de compression : mesure de la performance d'un algorithme de compression de données informatiques. Ici, le principe de calcul utilisé correspond à la taille initiale du buffer divisé par la taille du buffer compressé.

Le **codage différentiel** : est prise en compte, la différence mathématique entre deux données contigües stockées en mémoire. Pour qu'il soit performant, celui-ci a besoin de connaître le format de la donnée à compresser (8, 12, 16, ... bits).

Codage **Huffman simplifié** : On prédéfinit un arbre de probabilité sans tenir compte du taux de récurrence réelle des données.

Données à longueur variable : La donnée est stockée sous forme d'une suite de bit de longueur variable qui n'est pas modulo 2^8 . Ce point est important à prendre en compte car il signifie qu'une donnée peut être enregistrée sur moins de 8 bits et/ou qu'elle peut être stockée de façon non aligné sur deux octets ou plus.

4. SPECIFICATIONS

4.1 Principe de base

Le principe de base est assez simple à comprendre :

- Si une suite de données est issue de mesures physiques qui évoluent peu, la différence entre deux mesures tend à être faible. Cette différence peut être stockée en mémoire avec moins d'information codante.

Néanmoins afin d'optimiser au mieux le codage, il est nécessaire d'avoir un symbole qui indique le nombre des bits (information) à suivre.

L'algorithme se base sur un codage d'Huffman simplifié en prenant comme hypothèse de départ que les données variaient peu ou pas entre deux mesures. Dit autrement, cela signifie que la probabilité que dans le buffer la donnée à suivre soit de même valeur est plus forte que la donnée à suivre varie d'un point. De même, la probabilité de trouver une donnée qui ne varie que d'un point est plus importante que la donnée varie de deux points, Etc...

Il est également prévu que si la donnée varie trop fortement alors la nouvelle donnée est stockée en brute avec un symbole codant spécifique. Dans ce cas, on arrive à un taux de compression ponctuel inférieur à 1. Cela signifie aussi que si les données stockées à suivre dans un buffer varient de façon trop importante que la compression est inefficace.

4.2 L'arbre de probabilités

Comme indiqué plus haut, l'arbre des probabilités est préétabli afin de simplifier (et même d'optimiser) le codage et le code compressé.

A noter que ceci est vrai uniquement s'il y a peu de données à compresser. Si le nombre de données est important d'autre solution plus efficace serait à favoriser par rapport à ce type de compression.

ATTENTION : *sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.*

nke ☎ (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..<http://www.nke.fr>

Modèle SPDL.dot Rev 1 du 08/06/98

	SPECIFICATIONS DETAILLEES DU LOGICIEL	SPDL_DeltaCompression.docx	
		Delta Compression	

- « 0b000001 » → Une différence entre +/- 64 à 127
- « 0b100001 » → Une différence entre +/- 128 à 255
- « 0b0000001 » → Une différence entre +/- 256 à 511
- « 0b10000010 » → Une différence entre +/- 512 à 1023
- « 0b10000011 » → Une différence entre +/- 1024 à 2047
- « 0b000000010 » → Une différence entre +/- 2048 et 4095
- « 0b000000011 » → Une différence entre +/- 4096 et 8191
- « 0b100000010 » → Une différence entre +/- 8192 et 16383
- « 0b100000011 » → Une différence entre +/- 16384 et 32767
- « 0b100000000 » → Une différence entre +/- 32768 et 65535
- « 0b000000000 » → Une différence entre +/- 65536 et 131071
- « 0b100000001 » → Une différence entre +/- 131072 et 262143
- « 0b0000000010 » → Une différence entre +/- 262144 et 524287
- « 0b10000000100 » → Une différence entre +/- 524288 et 1048575
- « 0b10000000101 » → Une différence entre +/- 1048576 et 2097151
- « 0b00000000110 » → Une différence entre +/- 2097152 et 4194303
- « 0b00000000111 » → Une différence entre +/- 4194304 et 8388607

Les codes de probabilité représentent ainsi en mémoire les champs suivant :

0b11

0b01S

0b101SX

0b001SXX

0b1001XX...XX (Suivant le format de la donnée de base)

0b0001SXXX

0b00001SXXXX

0b10001SXXXXX

0b000001SXXXXXX

0b100001SXXXXXXX

0b0000001SXXXXXXXX

0b10000010SXXXXXXX

0b10000011SXXXXXXX

0b000000010SXXXXXXXX

0b000000011SXXXXXXXX

0b100000010SXXXXXXXX

0b100000011SXXXXXXXX

0b100000000SXXXXXXXX

0b000000000SXXXXXXXX

0b1000000011SXXXXXXXX

0b0000000010SXXXXXXXX

0b10000000100SXXXXXXXX

0b10000000101SXXXXXXXX

0b00000000110SXXXXXXXX

0b00000000111SXXXXXXXX

Avec S pour le signe et XX..XX une valeur binaire.
Si S = 1 alors le delta est négatif sinon il est positif.

4.2.2 Taille en bit de l'encodage

L'encodage en bit permet d'avoir des tailles variables, ainsi :

- 1) 0b11 as une longueur de 2 bits
- 2) 0b01... as une longueur de 3 bits
- 3) 0b101... as une longueur de 5 bits
- 4) 0b001... as une longueur de 6 bits
- 5) 0b1001 as une longueur de 4bits + format de la donnée

ATTENTION : sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.

nke (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..http://www.nke.fr

Modèle SPDL.dot Rev 1 du 08/06/98

	SPECIFICATIONS DETAILLEES DU LOGICIEL	SPDL_DeltaCompression.docx	
		Delta Compression	

- 6) 0b0001... as une longueur de 8 bits
- 7) 0b00001 ... as une longueur de 10 bits
- 8) 0b10001 ... as une longueur de 11 bits
- 9) 0b000001 ... as une longueur de 13 bits
- 10) 0b100001 ... as une longueur de 14 bits
- 11) 0b0000001 ... as une longueur de 16 bits
- 12) 0b10000010 ... as une longueur de 18 bits
- 13) 0b10000011 ... as une longueur de 19 bits
- 14) 0b000000010 ... as une longueur de 21 bits
- 15) 0b000000011 ... as une longueur de 22 bits
- 16) 0b100000010 ... as une longueur de 23 bits
- 17) 0b100000011 ... as une longueur de 24 bits
- 18) 0b100000000 ... as une longueur de 25 bits
- 19) 0b000000000 ... as une longueur de 26 bits
- 20) 0b1000000011 ... as une longueur de 28 bits
- 21) 0b0000000010 ... as une longueur de 29 bits
- 22) 0b10000000100 ... as une longueur de 31 bits
- 23) 0b10000000101 ... as une longueur de 32 bits
- 24) 0b00000000110 ... as une longueur de 33 bits
- 25) 0b00000000111 ... as une longueur de 34 bits

Par exemple, on voit ainsi que le meilleur taux de compression que l'on puisse avoir pour une donnée originale stockée sur 16 bits est d'un facteur 8 (si la donnée ne change pas).
La limite de compression efficace (donc supérieure à 1) est lorsque les données varient moins de +/- 255.

A noter que comme il est nécessaire pour ré enregistrer une nouvelle valeur en brute qu'il faut 4 + 16 = 20bits. On utilise les codes de probabilité jusqu'à « 0b10000011 » car il ne stocke en mémoire que 19 bits donc plus rentable.

Il en va de même pour les autres formats de compression, on utilise :

- 0b10001 (soit 11 bits) pour un format de données en 8 bits
- 0b10000011 (soit 19 bits) pour un format de données en 16 bits
- 0b00000000111 (soit 34 bits) pour un format de données en 32 bits.

4.2.3 Stockage des bits

Les bits sont stockés en mémoire au format big Endian, soit bit de poids fort en premier.

A la décompression, les bits des données sont stockés au format little endian donc les poids faible au début. Si l'on considère la suite suivante : 202, 197, 198, 197, 196, 204 soit en format little endian 16 bits le tableau suivant :

0xCA, 0x00 ; 0xC5, 0x00 ; 0xC6 ; 0x00, 0xC5, 0x00 ; 0xC4, 0x00 ; 0xCC, 0x00.

Passée dans l'algorithme de compression on a : 0x09, 0x30, 0xC5, 0xCA, 0x46, 0x00.

Soit en décortiquant encore un peu la suite ci-dessus :

0b00001001, 0b00110000, 0x11000101, 0b11001010, 0x01000110, 0x00000000

En violet le code : 0b1001 soit une valeur absolue.

En vert la donnée: 0b00000000,11001010

En orange le code suivant : 0b001

En bleu la donnée associée 0b101, ici le delta est négatif de 1 + 4 donc -5

En Rouge le code suivant 0b01

En gris la donnée associée 0b0, soit +1

En pourpre le code suivant, soit 0b01

En orange la donnée suivante associée 0b1 soit -1

En bleu foncé le code suivant : 0b01

En vert foncé la donnée suivante associée : 0b1 soit -1

En bleu le code suivant soit : 0b0001

ATTENTION : sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.

nke ☎ (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..http://www.nke.fr

Modèle SPDL.dot Rev 1 du 08/06/98

En rouge la donnée associée soit : 0b0000 soit +8

Donc, on reconstitue la suite d'origine :

Soit 202 (=0x00CA), 197 (=202-5) ; 198 (=197+1) ; 197 (=198-1) ; 196 (=197-1) ; 204 (196+8)

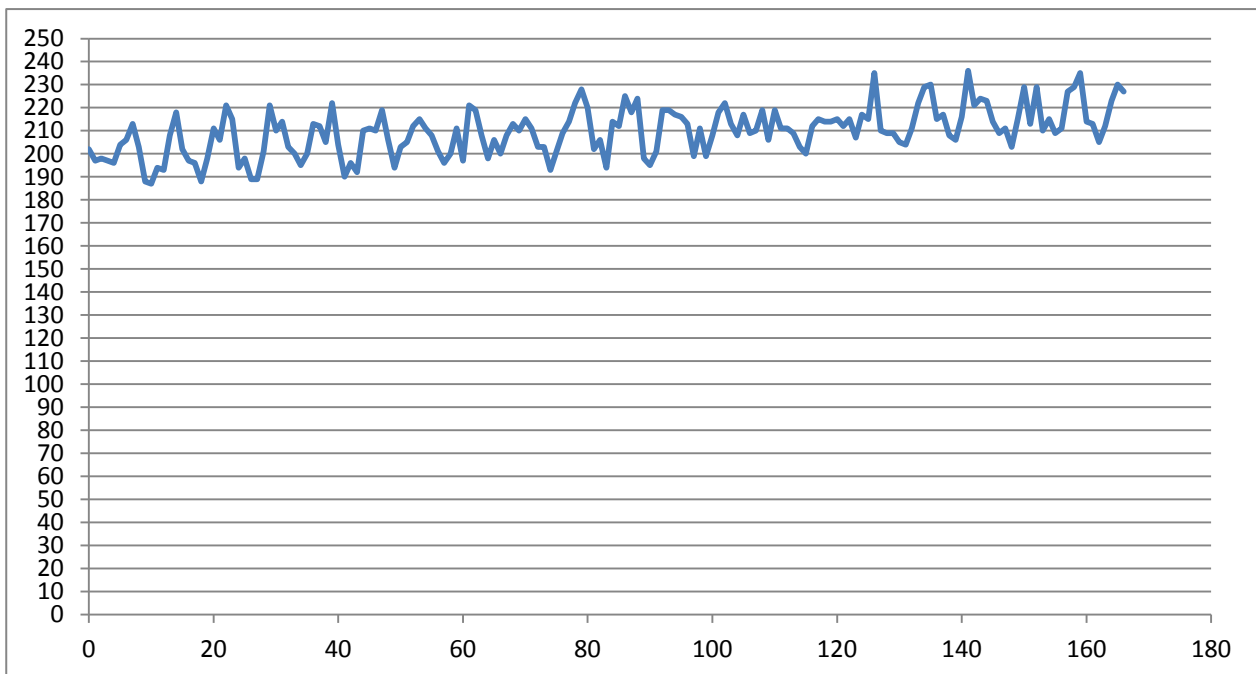
On retrouve bien nos données de départ.

4.2.4 Les limites absolue de compression

Pour des données :

- au format 8 bits, le taux de compression maximal est < à 4.
- au format 16 bits, le taux de compression maximal est < à 8.
- au format 32 bits, le taux de compression maximal est < à 16.

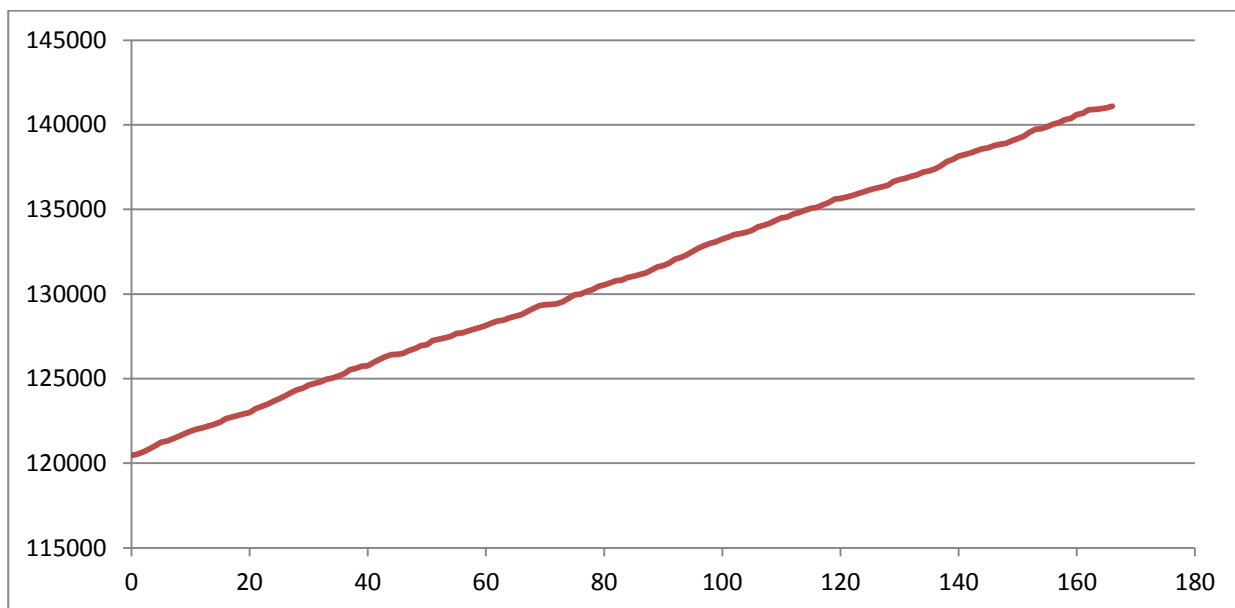
4.3 Exemple de compression



Une pseudo-courbe de température ci-dessous comprenant 167 mesures au format 16 bits.

Soit un buffer de $167 * 2 = 334$ octets donne comme résultat de compressions 142 octets soit environ un facteur de 2,3.

Si l'on prend maintenant un exemple sur un compteur de type S0 sur 32 bits, avec une courbe de ce type



Il y a également 167 mesures mais sur 32 bits soit 668 octets. La compression donne un résultat de 278 octets soit un gain de taille d'un facteur d'environ 2,4.

ATTENTION : sauf indication contraire (mentionnée ci-dessus), ceci est une copie non gérée d'un document informatique. Consulter le logiciel Gestionnaire de Documents nke pour plus d'informations.

nke ☎ (33).02.97.36.10.12 Fax (33).02.97.36.55.17.. ..<http://www.nke.fr>

Modèle SPDL.dot Rev 1 du 08/06/98