Introduction to kdb+

Rory Winston
Ajay Padala

Outline

- What is kdb+?
- History and Philosophy of kdb+
- Features
- Real-world usage examples
- Getting started

What is kdb+?

- A language
- A database (in-memory store and persistent)
- An interpreter
- An IPC / client-server architecture
- All in a ~500kb binary!

Who uses kdb+?

- Financial Institutions
- Exchanges
- Regulators
- Pharma
- Military
- And more…

Why use kdb+?

- Performance, performance, performance
- Simplicity
- Scalability

Comparing, say, Kx systems Q/KDB (80s technology which still sells for upwards of $100k a CPU, and is worth every penny) to Hive or Redis is an exercise in high comedy. Q does what Hive does. It does what Redis does. It does both, several other impressive things modern “big data” types haven’t thought of yet, and it does them better, using only a few *pages* of tight C code, and a few more pages of tight K code.

scottlocklin.wordpress.com

A Brief History of kdb+

The Originators

Photo Credit: Rob Hodgkinson

Ken Iverson

- Ken Iverson's "A Programming Language" is 52 years old this year
- Iverson developed a mathematical notation for manipulating arrays while teaching students at Harvard
- Developed APL while working at IBM in 1960

APL: A Programming Language

- In the paper “Notation as a Tool of Thought”, Iverson expanded on his philosophy for the design of APL

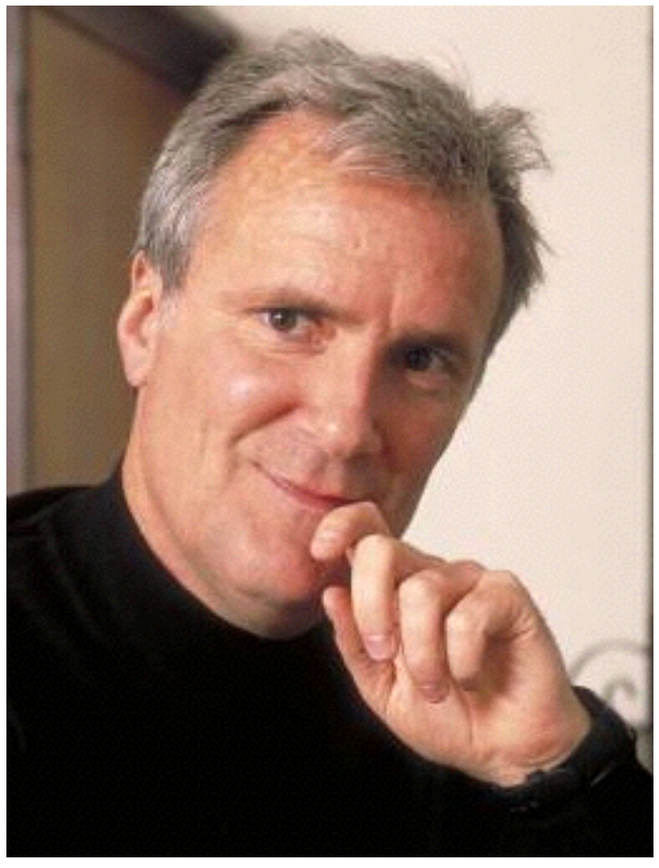
- Designed an efficient and consistent programming notation inspired by mathematics

APL: A Programming Language

```
(~R∈R∘.×R)/R←1↓⍳R
```

```
life←{⍵∨.∧3 4=+/,¯1 0 1∘.⊖¯1 0 1∘.⌽⊂⍵}
```

Arthur Whitney



- Developed A+ while working at Morgan Stanley in 1988
- A+ was a high-performance APL implementation designed to run on SunOS
- Had already collaborated with Iverson on APL for many years

The J Language

- When Iverson retired, he wanted to create a version of APL that was cheap and easily accessible (e.g. no special keyboards required, just using standard character sets)

- Enlisted the help of Arthur Whitney to produce the first prototype

> Work began in the summer of 1989 when I first discussed my desires with Arthur Whitney. He proposed the use of C for implementation, and produced (on one page and in one afternoon) a working fragment …
>
> *Ken Iverson, "A Personal View Of APL", 1991*

The First J Interpreter

```
typedef char C;typedef long I;
typedef struct a{I t,r,d[3],p[2];}*A;
#define P printf
#define R return
#define V1(f) A f(w)A w;
#define V2(f) A f(a,w)A a,w;
#define DO(n,x) {I i=0,_n=(n);for(;i<_n;++i){x;}}
I *ma(n){R(I*)malloc(n*4);}mv(d,s,n)I *d,*s;{DO(n,d[i]=s[i]);}
tr(r,d)I *d;{I z=1;DO(r,z=z*d[i]);R z;}
A ga(t,r,d)I *d;{A z=(A)ma(5+tr(r,d));z->t=t,z->r=r,mv(z->d,d,r);
 R z;}
V1(iota){I n=*w->p;A z=ga(0,1,&n);DO(n,z->p[i]=i);R z;}
V2(plus){I r=w->r,*d=w->d,n=tr(r,d);A z=ga(0,r,d);
 DO(n,z->p[i]=a->p[i]+w->p[i]);R z;}
V2(from){I r=w->r-1,*d=w->d+1,n=tr(r,d);
 A z=ga(w->t,r,d);mv(z->p,w->p+(n**a->p),n);R z;}
V1(box){A z=ga(1,0,0);*z->p=(I)w;R z;}
V2(cat){I an=tr(a->r,a->d),wn=tr(w->r,w->d),n=an+wn;
 A z=ga(w->t,1,&n);mv(z->p,a->p,an);mv(z->p+an,w->p,wn);R z;}
V2(find){}
V2(rsh){I r=a->r?*a->d:1,n=tr(r,a->p),wn=tr(w->r,w->d);
 A z=ga(w->t,r,a->p);mv(z->p,w->p,wn=n>wn?wn:n);
 if(n-=wn)mv(z->p+wn,z->p,n);R z;}
V1(sha){A z=ga(0,1,&w->r);mv(z->p,w->d,w->r);R z;}
V1(id){R w;}V1(size){A z=ga(0,0,0);*z->p=w->r?*w->d:1;R z;}
pi(i){P("%d ",i);}nl(){P("\n");}
pr(w)A w;{I r=w->r,*d=w->d,n=tr(r,d);DO(r,pi(d[i]));nl();
 if(w->t)DO(n,P("< ");pr(w->p[i]))else DO(n,pi(w->p[i]));nl();}
C vt[]="+{~<#,";
A(*vd[])()={0,plus,from,find,0,rsh,cat},
 (*vm[])()={0,id,size,iota,box,sha,0};
I st[26]; qp(a){R  a>='a'&&a<='z';}qv(a){R a<'a';}
A ex(e)I *e;{I a=*e;
 if(qp(a)){if(e[1]=='=')R st[a-'a']=ex(e+2);a= st[ a-'a'];}
 R qv(a)?(*vm[a])(ex(e+1)):e[1]?(*vd[e[1]])(a,ex(e+2)):(A)a;}
noun(c){A z;if(c<'0'||c>'9')R 0;z=ga(0,0,0);*z->p=c-'0';R z;}
verb(c){I i=0;for(;vt[i];)if(vt[i++]==c)R i;R 0;}
I *wd(s)C *s;{I a,n=strlen(s),*e=ma(n+1);C c;
 DO(n,e[i]=(a=noun(c=s[i]))?a:(a=verb(c))?a:c);e[n]=0;R e;}

main(){C s[99];while(gets(s))pr(ex(wd(s)));}
```

The J Language

```
avg=: +/ % #

avg 1 2 3 4
2.5
```

$$avg(x) = \frac{\sum x}{|x|}$$

J is freely available for download at:

`http://www.jsoftware.com/`

The K Language

- In 1993, Whitney left MS and started to develop K
- Formed Kx systems and signed a commercial contract with UBS
- In 1998, released the first version of kdb
- kdb contained the q language, a higher-level language built in k

```
s:{(,x)(,/{y@[x]'10^x*|/p[;y]=p,:,3/-3!p:!9 9}')/&~x}
```

Arthur's one-line Sudoku Solver in k

The q Language

- Q is the modern interface most programmers will use when interfacing with kdb
- Built entirely a single K file
- Contains a SQL-like interface and a rich set of datatypes (tables, functions, dictionaries, lists)

```
select count i by exchange,sym from quotes
    where date=2014.09.09, qty>100000
```

Photo Credit: Rob Hodgkinson