

Co-Dfns Compiler

Aaron W. Hsu

Indiana University





APL is fast. Who needs a compiler?





Compilers can enhance reliability and performance.





Interpreter Limitations

Scalar fusion leads to
poor cache behavior

Procedure calls can
be expensive

Common
optimizations are not
possible

No static analysis of
code for errors or
correctness





Co-Dfns is a headlong jump into APL compilation.





D-fns eliminates serious roadblocks to performance.



The interpreter can have poor cache behavior.

$((\otimes S \div X) + (r + 2 \div \sim v * 2) \times T) \div v \text{sqrt} T \leftarrow v \times T * 0.5$





Co-Dfns is for predictable, safe, controllable performance.
Tuning expertise should not be a black art.



Co-Dfns is a two-primitive
extension of the D-fns language.



Examples

```
X ← 5 5 p @    a 5x5 single-assignment array
X[0;0] ← 3     a Set a single assignment cell
3 ≡ X[0;0]     a Reference a cell
? ≡ X[0;1]     a Blocking call waiting for data
F || "         a Parallel Each
a "Parallel" reduction of depth 1 vector
VECREd ← {
  Z[] ← (ω), (-1 ↓ ι ρ ω) α α {Z[α] α α ω} || "1 ↓ ω → Z ← @ ρ ~ ρ ω
}
```



Examples

```
TreeVecRed←{  
  
}
```

```
Life←{>1 ωv.^3 4=+/,1 0 -1°.θ1 0 -1φ¨cω}
```

```
LifeP←Life||
```

```
LifeP2←{>1 ωv.^3 4=+/,1 0 -1°.(θ||)1 0 -1φ||¨cω}
```





Why shouldn't we be able to reason about performance?

ZPL and others make this a critical component.



00bp^7!s=1;7∈U-1a0a□.p≠-lw-ld0vc∇U3d∇vε;kθΦ+[d↑|[e"↓>u←Ttw*←θUT@\.?◊=↓≤≠~↓Hpo!7□≠1-
!60v∇←Δ~◊*2/η≠ξ@3?=a7~\↑L∇2I◊h2npesqn-◊/cΔI×~↑zmcϕΔ.≠14gsϕcxΦ=i∇ie→lrT@t□:y^∇≠2=
H-↖↘□i-∇-ε-HθUig6=I;v!g@T^↖□□=wαξoa9^ua_n17θu∇d@` ,#.□θ↑↑O|0-H≠w≡↓:z7-xΛθ≠÷ι`!=ff2z
ε_!3+ιlbj↖Hι1w>Δ∇+□U@Δ≡Ijlh>□i?→v+aa*ox>,o?m∇≡gtwn◊∇x↑9*s≠Λ□1s@≡∇b≡□÷t∇∇c@∇≡^y-
ε0ex□g~◊≡rΔ=□Φ=bnpo◊-n,ck]Iα/*n□≤I↑t4oga∇n?=-↑z^-◊cz↓θ◊b~~^↑|[j3a□v↑/≤O9u≡ΔΔ=<□/eb≠
∇Λ@∈lyTT\Δθ□≡↑*a≠∇ϕ9;!≡≥÷9h□n□[l/l↑↑∞÷>r∇og□∇4zIjpc=`□□09∇"◊@^0↑Δaθ0ehθ≤HΔf~t^θ@
◊8@p|∇□!H?^∇-∇ϕ↖bllw≡k1θΔg^*θ∇tΔ.t÷`ie□:a-x↖a□∇a-tH;#*^ε↑.gceuw~;O4≠≠^p□4@□Urθrm
→a~"[paΔ□8-Hl@p^7↑≥□iH-oOwp4^5H~"cι4@□≠7□≡^r-ριvp27◊;□\k3iε4≤↓]≡.O^]Δ↖7Λ^L3,1L91n@ε1
TO1Lc@ε□sd◊^ω`kι,≤>1a^∇17≠□aaεj^r÷↓_zθr*iθ~^H1÷∇←2≠a!□◊□s□a≠?~y≠∇^3q<n+8□|εaLwLn^
□←_↑1,o,∇+h?>Λ→@-∖u1[v1gx^□ϕ|H~→ι`od≥→d_@wb^≤^vθθ@↑\~k[θ∇omϕ7i?□8εka∇w?nΔ*5Δt◊≡]
a1□>jr↑6=→Λ1θ>@=_@ι^-1↖□>θ^r^3Δ!lι7n+j→Φxna□Liqw∇6+↓_ϕv|. `co>□□∇vqH-T!→@-~H↑>~◊_Te>m
I-H∇xbcc→4◊i^iϕH9a↑Φuv≠q→≠0↓amrw>1~↓`◊u^ob/↑>I≥□↓]□v≡Tuι`495anιΔ,d∇0!^?Λa□@p-71Ly
q\`k◊≠5w/↑~fU-H∇e^Φ~"↑ε←∇∇~"□a□εa1←:□/2p~"bv"4θ□-wi<∇n\>Δ;`^u7Δ?1↓6a∇5p7L□p7kkθΔp@1
6≡↓<"=□◊□ι◊7r7Δ!vxo|ι!L≠◊∇pg8□↓`6εq≠↖Φ,w→<∇Δ1≠uι7<9h7`≠0p∇]uaO4]^∇f←mϕyθ◊Ln□∇4≡L
tu>a]≥nukcI<□k∇nop-HLnθH~. ^y?@npIn6ε>u\lθ6@Hmgns;v@avi^*θ<c↓^1◊≠≥@fq÷q>?1i↖z2∇d≤\
r□□81,Δεf←◊↖80Δ^|ka]_xH,~nn_\8an◊`c□≡3^,`,→∇↑~θw@↓7kξeqε^75^v,a.θ∇◊≡@_≡sfk□j◊.→ϕ
□7.*.↑8>↓.HUn≤a-w∇1Tp15□k5diw∇∇θpa096>ι|8Lk^□a≠!^fθ◊1[∇33,_↓^∇iw[θT3xnuεϕ0h÷^9Lι
∇xpO6↓∇nιkv∇ϕ□↓≡g^r<5Uuεθd□]^a×H-HHng8!ϕvanlv-HyT□>^L^ge4L←Φ□@LΛk@L1∇0εθ∇◊a∇≠iΔfε
x-2ic^r□6÷0a∇uT^i□ni×a@caθh7u≡ε1^ϕve^□8Δ∇∇ι^*U!L;k≥h≠>0θ≡7@La,[~◊av/^*□Δ≠H≤-□O~∇↖+i
∇|÷L>pa◊java□4.8Tabwo>p∇≠r+H^∇w-Ha≤^*\j≡a-H096εppy4=a_3<n]◊ε□*Δ≠6Φ≠^2∇◊≡aa_≠∇↖ξ◊*s=
□∇θθ←vb@SΔ]^zd/3Δ@1c>qca*ax↑xeεL-H→y≡r≠nt2edL|].3f1u,~w=□6-oo↑□Φ@>s=OιL∇U∇y<→njUθ
UΛ□↖8H-pi6~Δg@=>◊-Hg737.=≤Λ→]fI□\@a+m2θ>≥j~r≡v]h0ε□<!^~Δx>iT~9◊[Φ≠-nHkθy@↑H4≡.~?4
iϕιϕ□g>xc≠j□≡w9uxL+o□□∇vi1□qr^ιoma0U□wa?=>Φ-≠71>≡↖!Tjv;axL77L◊∇Un◊av∇i23>gΔn@i!h
@31ϕεθ;2d-Hp_↑;^◊zi□ι≡]nθΛHn◊@□∇←□∇∇~"~□_==^a3n∇~θ≡∇Δ↖o□g^r◊-θ≡ι∇ϕΔ>≤a0>x^∇≤a~□rx◊_
hΛ-H×8∇≡↑∇∇←r□≠→n≠aεOg←a`".n~--~]n_1θ◊∇Lii↓□8↓_>k~◊;←^n@>←εΦ`ro1t↑@5fθ>~↖≡↖r2L@a^a6

APL is math you can use. Let's use it for performance.



Examples

$\text{Life} \leftarrow \{ \triangleright 1 \ \omega \vee \cdot \wedge 3 \ 4 = + / , - 1 \ 0 \ 1 \circ \cdot \ominus^{-1} \ 0 \ 1 \phi \ddot{\cdot} \subset \omega \}$
 $\langle (\rho \text{Life } A) \leftrightarrow (\rho A) \rangle$

$\langle Z \ F \ (Y \ F \ X) \leftrightarrow (Z \ F \ Y) \ F \ X \rangle$
 F / ω

$\langle G \ \text{DeepMap} \leftrightarrow G \rangle$

$X + Y \ G \ Z \leftrightarrow \{ x \ y \ z \leftarrow \omega \ \diamond \ x + y \ G \ z \} \ \text{DeepMap} \ X \ Y \ Z$
 $\leftrightarrow \{ \langle (, 3) \equiv \rho \omega \rangle \ x \ y \ z \leftarrow \omega \ \diamond \ x + y \ G \ z \} \ \text{DeepMap} \ X \ Y \ Z$



Examples

$\langle (\theta \equiv \rho S) \wedge (F \text{ DeepMap } \leftrightarrow F) \rangle$

$M \ F \ (\rho M) \rho S \rightarrow M \ F \ S$

$\langle (\theta \equiv \rho X) \wedge (X \ F \ Y \leftrightarrow Y \ F \ X) \rangle$

$X \ F \ Y \rightarrow Y \ F \ X$





Plans and Schemes

Usable version this year

Target multi-core, GPU, and distributed clusters

Fully integrate with Dyalog

Focus on scalable parallel performance

Leverage APL-style formalism

Create a dialog between tuning expert and compiler





Version 1 will have low-hanging fruit optimizations



Interpreter vs. Compiler

INTERPRETER

- Garbage collected
- Idiom-based special casing
- No static verification
- No user-guided optimization
- Supports much more of APL
- Numerous extensions

COMPILER

- Stack-based allocation
- Whole program optimization
- Explicit verification and proof
- Safe user-defined optimizations
- Restricted to Co-Dfns
- Limited extensions and interop





Using Co-Dfns should require less than trivial effort.



Demo time





Performance not included, some assembly required.
We're not preaching super-compilation here.





Coding Goodies

Easily integrated with other code

A complete, useful, general parser for APL

A complete, rigorous specification of the language

Supports multiple runtimes

Language integration



Acknowledgments

Dyalog has provided significant funding and support.

PL Group at IU is a great place to test and strengthen these ideas.

John Scholes, for creating D-fns.

Intel's Concurrent Collections

Mathematics of Arrays

C++ (Oh dear!)

NanoPass



Thank you

Email: awhsu@indiana.edu



Photo Credits

Hotnstock

MoonsongStock

Ladyaway

Fall_Stock

Estruda

Animalphotos

Dark_dragon_stock

Drezdany_stocks

Stupiddeeppeople

Thiselectricheart

Vampsstock

NefaroStock

NikiljuiceStock